UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

RESMED CORP, PETITIONER, V. CLEVELAND MEDICAL DEVICES, INC., PATENT OWNER. _____	)CASE IPR 2025-00159)U.S. PATENT NO. 11,375,921))CASE IPR 2025-00160)U.S. PATENT NO. 11,786,680))MOBILE VIDEOCONFERENCE)DEPOSITION OF)SANDEEP CHATTERJEE, Ph.D.))

MOBILE VIDEOCONFERENCE DEPOSITION OF SANDEEP CHATTERJEE, Ph.D., taken remotely before Cheryl A. Rooney, RPR, CRR, Online General Notary Public within and for the State of Nebraska, beginning at 9:58 a.m. PST, on January 27th, 2026.

A P P E A R A N C E S

(Appearing Remotely)
THE WITNESS:
SANDEEP CHATTERJEE, Ph.D.

(Appearing Remotely)
FOR THE PETITIONER:

PAUL HASTINGS LLP
BY: KAMILAH ALEXANDER
4655 Executive Drive
Suite 350
San Diego, CA 92121
858.458.3000
kamilahalexander@paulhastings.com

(Appearing Remotely)
FOR THE PATENT OWNER:

HERBER SMITH FREEHILLS KRAMER
BY: JENNA FULLER
JEFFREY PRICE
1177 Avenue of the Americas
New York, NY 10036
212.715.7502
jenna.fuller@hsfkramer.com
jeffrey.price@hsfkramer.com

I N D E X

CASE CAPTION..... PAGE 1
APPEARANCES..... PAGE 2
INDEX..... PAGE 3
TESTIMONY..... PAGE 4
REPORTER CERTIFICATE..... PAGE 78

E X A M I N A T I O N

DIRECT EXAMINATION:

BY MS. FULLER..... PAGE 4

E X H I B I T S

NO EXHIBITS MARKED

SANDEEP CHATTERJEE, Ph.D.,
having been first duly sworn,
was examined and testified as follows:

DIRECT EXAMINATION

BY MS. FULLER:

Q. Good morning, Dr. Chatterjee. How are you today?

A. Good morning. I'm doing well.

Q. Thank you for being here. Do you understand why you are here today?

A. To have my deposition taken.

Q. And your deposition taken in connection with your reply declaration submitted in IPR 2024-00159 and 00160; is that correct?

A. I do not recall the IPR numbers, but it's the IPR numbers related to the '921 and '680 patents.

Q. Is that okay today when referencing your declarations that we call them the '921 declaration or the '680 reply declaration?

A. Yes, that will be great.

Q. And just a little housekeeping before we get started. Do you have your declarations available to you, or would you like them uploaded? Let me know.

A. I have printed out a hard copy of the declarations in front of me.

Q. And aside from the hard copies of the declarations, what other materials do you have in front of you?

A. I have a coffee and a water. And a -- it looks like a binder that has the patents, like all of the materials for the IPRs.

Q. Okay. Good to know. Then if I reference something outside of your reply declaration you'll have a binder available to you for hard copies of exhibits, for example; correct?

A. I believe so. I'm not sure if it has everything, but it's a pretty thick binder. I haven't looked through it. But I know for a fact it has the two reply decs.

Q. Okay. Well, we'll cross that bridge if we get there. If you need something that's not available to you in your binder, I can upload it to the chat. Is that all right?

A. That sounds great.

Q. So in preparing for your deposition today, did you review any materials, exhibits in that preparation?

A. I did.
Q. And what were those materials that you reviewed?
A. I reviewed my two reply declarations. I reviewed Dr. Goodrich's declarations. I obviously reviewed the patents. I reviewed my original declarations. So those are some of the items that come to mind, but there were quite a few documents that I reviewed in preparation for today.
Q. Just giving me a ballpark, how much time did you spend in your preparation for today's deposition?
A. So do you mean only for preparing for the depo today, not like everything, like from my original declarations, just for preparation for today?
Q. Yes, just for preparation for today's deposition.
A. I don't know, but probably in the 20ish hours, I would think.
Q. So Dr. Chatterjee, can you open your '921 reply declaration, and you just spoke that you have that available to you.
A. Okay. I have my '921 declaration in

front of me.

Q. So I'm going to direct you to Paragraph 3 where you state that you've reviewed the patent owner response and the declaration of Dr. Goodrich, and provide opinions to certain arguments made in those documents. Do you see that?

A. I do.

Q. Why did you only provide opinions to certain arguments? Did you agree with the other arguments that Dr. Goodrich provided?

A. No. Dr. Goodrich's analysis and opinions are just wrong, and fundamentally wrong from a technical perspective.

Q. So then what made you only submit a reply declaration that addressed certain arguments as opposed to his whole declaration?

A. Well, I think some of the arguments that he makes I had previously addressed. For example, even at my deposition -- like for example, his assertions about NAT technology is just plain wrong. And this was addressed earlier, I believe, in my earlier declaration so there was no need to again respond to things that already have been identified as being

fundamentally flawed and incorrect.

Q. Did you address NAT technology in your opening brief -- or opening declaration?

A. I don't believe I did, but I believe I addressed it -- I believe you asked me questions during my deposition in relation to my first declaration.

Q. So during your testimony in your -- during your first deposition and testimony, that's where you addressed the NAT, network address translation issues; is that correct? Is that what you're saying?

A. I believe so, because like I'm saying, what he's saying in his declaration is just fundamentally wrong. Like you can read any basic computer science textbook and that would show that what he's saying in his declaration is wrong. So it was addressed -- you asked me -- you or your colleague asked me questions, and I addressed some of those at the deposition.

Q. What do you believe Dr. Goodrich opined to in his opinions on network address translation that is fundamentally wrong?

MS. ALEXANDER: Objection. Outside the scope.

THE WITNESS: Well, for example, I don't have his declaration here in front of me, but he seems to say that if a server is behind a NAT, that it's unreachable, and that is just fundamentally wrong. I've built web servers, I've had web servers that are behind the NAT, and they're clearly reachable. And so the fact that he makes these kind of statements which are fundamentally flawed demonstrates that his -- as I've pointed to in some of my other statements and opinions in my reply declaration, that his opinions and his technical analyses are wrong and flawed.

BY MS. FULLER:

Q. So you believe that his opinions for NAT is fundamentally wrong, but you did not provide the reasoning in your reply declaration; is that correct?

MS. ALEXANDER: Objection; mischaracterizes his testimony.

THE WITNESS: Well, what I stated was that I -- you or your colleague asked me questions about NAT, and then on top of that, even the point that he makes about NAT doesn't impact the analysis because he seems to say that

NAT is used prevalently. But that doesn't mean that it's used all the time. And so there was really no need for me to respond, even though his opinions are fundamentally wrong and flawed and just picking up a basic computer science textbook would demonstrate that it's wrong. But his opinion doesn't really impact the opinions that I stated because he himself is conceding that NAT is not used a hundred percent of the time. So clearly one of ordinary skill would understand that web servers don't have to be behind the NAT.

But what I'm saying is that even his arguments that are not relevant to the analysis here, it just shows that his technical analyses and the technical facts that he believes are correct are fundamentally flawed and incorrect.

BY MS. FULLER:

Q. Can we turn to paragraph -- let's start at Paragraph 5 of your '921 declaration.

A. Okay. I'm there.

Q. So in this section is it a correct understanding that you're opining that a POSITA would have implemented Toge's physician-side computer as both a client device and a web

server?

A. Say that again? I'm not sure I understood your question.

Q. So starting off of Paragraph 5, which is under the section Toge and BEO Kumar renders obvious a remote Internet site, is it your opinion that a POSITA would have implemented Toge's physician-side computer as both server and client?

A. I'm not sure what you mean by implemented as both a server and a client. What I'm stating is that one of ordinary skill would have known that implementing the physician-side computer as, for example, a web server was well known and common, even at that time period. And then in addition, with regards to the Kumar reference, it expressly talks about the use of web technologies and web pages.

And so what I'm stating here is that the physician-side computer could have a server running on it, as well as a client running on that same machine on the physician-side computer in order to, for example, allow that physician to access the data from that server.

Q. So when you speak to improving Toge's

system based on Kumar's web server teaching, is this in reference to the implementing of server on Toge's physician-side computer? And speak to it in Paragraph 20, for example. Or pardon me, Paragraph 15, a POSITA implementing the improved Toge system.

MS. ALEXANDER: Counsel, what's the pending question?

BY MS. FULLER:

Q. So where you speak to a POSITA implementing the approved Toge system, is that in reference to a physician-side computer that is implemented with a web server?

A. I'm not sure I understand your question still.

Q. I'll break it down. The improved Toge system, what is improved in the Toge system?

MS. ALEXANDER: Objection. Vague.

THE WITNESS: So are you -- you first asked me about Paragraph 5. And I think you read from the subtitle that's above Paragraph 5, which states that Toge, in view of Kumar, renders obvious a "remote intranet site where Toge's physician-side computer is

implemented as the web server."

So in view of what the title is saying, I'm not sure I understand your question.

BY MS. FULLER:

Q. I'm just trying to understand what is improved in Toge's system.

MS. ALEXANDER: Objection. Vague.

BY MS. FULLER:

Q. So is it a correct understanding that what is improved in Toge's system is its physician-side computer because that is implemented now as a web server in addition to a client?

A. I don't understand --

MS. ALEXANDER: Same objection. Sorry.

THE WITNESS: I didn't understand the last part of your question where you're saying it's implemented as to web server in addition to a client. What does that mean?

BY MS. FULLER:

Q. Is a physician-side computer in Toge's unmodified system, is that just a personal computer?

A. It's a computer. I'm not sure it's limited to a personal computer but it could be a personal computer.

Q. And now that it's in the modified system, it would be implemented with a web server based on Kumar's teaching; is that correct?

A. Well, just to be clear, when you're saying a web server, it's software that would be running on Toge's physician-side computer.

Q. So in running software on Toge's physician-side computer, which is the web server software, is that the improvement that you speak to in Toge's improved system?

A. That is one aspect of it, and I think I talk about it in my original declaration, the numerous benefits of that. And I believe it's readily obvious, for example, in Toge. Toge doesn't provide implementation details, and so a POSITA would naturally look to, for example, Kumar, and as I explained earlier, even outside of Kumar, one of ordinary skill would have readily known about web servers and web client technology or web-based technologies, and the fact that it would allow access to the data from

multiple devices and from different locations as well.

Q. Can we turn to Paragraph 8 of your '921 reply declaration.

A. Okay.

Q. So you say in the case of client and server both residing on the same computer, that communications from client server are located on the same machine. Are these communications going through a virtual interface on the local machine to allow, for example, the client browser and the local server to communicate with one another?

MS. ALEXANDER: Objection. Outside the scope.

Jenna, your words are getting clipped a little bit. If you could just enunciate a little better.

MS. FULLER: Oh, are they?

MS. ALEXANDER: Yes. Thank you.

MS. FULLER: Thank you for bringing that to my attention.

THE WITNESS: So I'm not sure I understand your question. What do you mean by virtual interface?

BY MS. FULLER:

Q. I want to understand how the communications from a client to server occur on the same machine. Could you walk me through that?

A. How they occur on the same machine? Well, like I mentioned, for example, server is a software, and if we're looking at specifically web server, it's a specific type of software. And a POSITA would know that web servers typically accept connections to port 80 or port 443.

And so with regards to what I'm talking about in Paragraph 8, a client, for example a web browser, which is an example of a client software, if the client is connecting to the local host, the operating system would deliver that message to port 80 or port 443 of the web server.

So what I'm talking about here is that the specific design of having a client and a server on a single machine is so common that there is a shorthand that is used called local host, and anyone who has ever developed software, especially network communication

software or web-based software, knows the use of local host.

And so I was very surprised in reading in Dr. Goodrich's statements that people don't have clients and servers running on the same machine, and in fact, even his own documents that he cites to disproves his own statements. He cites to one portion of a document, and if you look up like 10, 15 lines above to the document that he's citing to, it even says that clients and servers can be on the same machine.

And so what I'm simply stating here is again, like I explained to you at the beginning, that Dr. Goodrich's technical analyses and opinions are fundamentally incorrect, fundamentally flawed, and this is just another example, another datapoint of why Dr. Goodrich's statements and opinions are wrong and flawed.

Not only is it possible to have client and server software on the same machine, but it's common to do so, and it's so common that there is a shorthand called local host to allow this kind of communications between clients and servers running on the same machine to happen.

Q. So in this local host example, the

client browser that's on the machine would send a request for data to the local host server that's on the same client machine, and the local host server would send back the data; is that correct, what's going on in the communication?

A. Yes. So for example, if the server is a web server and the client is a web client, for example, a browser, you would type into the part of the browser where you put in the URL, you would put in HTTP://localhost, and then you would put a colon for either port 80 -- even if you don't put the colon, the browser will automatically put in port 80. And it will send the message to the server software listening on that port and the server software will respond to that request, that HTTP request with the data. Which would be, for example, HTML page or HTML formatted web page.

Q. And that's the local host server that's on the same machine; correct, that's responding?

A. No, it's not a local host server. It's a web server. Local host is simply a shorthand to refer to servers on that machine.

Q. So it's a web server on that same

machine that's responding to the HTTP request from the client browser; that's a correct understanding?

A. On that same machine, yes.

Q. Dr. Chatterjee, do you have Exhibit 2020 available to you? Let me know.

A. I'm being told that I do have it.

MS. ALEXANDER: At the other end, the back are exhibits, these are some of them.

THE WITNESS: Oh, okay, I see it.

BY MS. FULLER:

Q. You have a good team on your side prepping you.

A. Yeah. Okay, I have Exhibit 2020 open.

Q. Can I direct you to the first page of that exhibit, specifically where it says servers play very important roles in information systems. It's right under the server heading.

A. The servers heading, like --

Q. Yes, page one. Let me know when you get to that sentence and I'll let you read it.

A. So it's the second sentence in that section?

Q. Yes.

A. Okay. I've quickly scanned through

it.

Q. So in that sentence it has the term distributed computing environments. Do you see that term?

A. I do.

Q. Do you have an understanding what distributed computing environments means? And if so, could you elaborate on that?

MS. ALEXANDER: Objection. Outside the scope.

THE WITNESS: So distributed computing typically refers to computing resources and functionality that's distributed and connected by a network.

BY MS. FULLER:

Q. And what are the advantages that flow from distributed computing environments?

A. Well, it can -- like we've talked about, it can allow access to data from different devices and different locations.

Q. So for an example, a client device could access data from a database server that's at a different location? Is that an example of an advantage of a distributed computing environment?

A. It could, yes.
Q. And do you have an understanding of why a computer could not take advantage of this distributed mobile -- distributed computing environment? For example, why a client device would not be able to access a database server?
MS. ALEXANDER: Objection; vague.
THE WITNESS: I'm not sure I understand the question. What do you mean, why a client device would not be able to access a database server?
BY MS. FULLER:
Q. Are there reasons why a computer could not take advantage of the distributed computing environment?
MS. ALEXANDER: Objection; vague.
THE WITNESS: I'm still not understanding the question. If a client device is meant to be used within that distributed computing environment, I'm not sure your question makes sense.
BY MS. FULLER:
Q. Well, let's go to -- give me one second. Okay. Let's go to Page 4.
A. Of Exhibit 2020?

Q. Page 4 of Exhibit 2020, under the heading Servers For Multiple Tiers of Client Server.

A. Okay. I'm there.

Q. So this states or discusses a single-tier client server application. Do you see that discussion?

A. Are you looking above Section 2 or below?

Q. I'm looking in the paragraph below Section 2.

A. You mean the sentence where it confirms that a client process and server process can be on the same computer?

Q. The sentence after that, A single-tier client server application.

A. Okay. I'm there.

Q. So it talks about a single-tier client server application. And then in the next sentence it says, This kind of application is rarely used today because it will not take advantage of the distributed computing environment.

Do you see that?

A. I do.

Q. Do you have an understanding why a single-tier client server application cannot take advantage of the distributed computing environment?

A. Well, it's saying if the application is limited to having the user interface, the business rules, and the data manipulation processees all on one computer. But that's not what we're talking about. We're talking about you have the server and you have the client, and in that typical web server with a web client it's not precluded from having other devices access that server on that machine as well.

And so I don't really understand the relevance of what you're pointing me to, to what we're talking about in the combination based on Toge that I opined on.

Q. What is your understanding of a single-tier client server application? Is that where a client and server are on the same machine?

A. Well, it's talking about where you have the user interface, the business rules, and the data manipulation processees all on one computer. But what we're talking about in the

Toge system, in combination with the knowledge of a POSITA or with Kumar, we're talking about a web server and a web client, and clearly there can be other clients that are able to access that web server.

And so this is talking about something completely different than what the actual combination that we're talking about in my declarations.

Q. A user interface, is that part of a client browser?

A. It could be.

Q. And then the business rules, what -- is that part of a server?

MS. ALEXANDER: Objection. Relevance.

THE WITNESS: The business rules could be part of the business logic that the server performs.

BY MS. FULLER:

Q. And the data manipulation processees, could that also be part of the server?

A. Could be part of the functionality that the server performs. But again, like I explained, this is not what we're talking about.

We're talking about a web client and the web server, and this is talking about something else where it's not a web client, it's talking about a client which cannot be used on any other device.

As we know, a web client by definition, for example, a web browser, can be used on multiple devices. And so you can have a web client on that same machine that is hosting the server, but you can also have web clients on a multitude of other machines as well, and access the server on the physician-side computer in the combination that we're talking about.

Q. And the combination we're talking about in which Toge's physician-side computer is operating as a web client, as well as a web server; is that correct?

A. Yes. But like I explained, that's not what this is talking about. This is talking about the situation where you don't have or cannot have other clients that are accessing the server.

With the web client, a web browser can be run on multiple devices, and by putting in the URL for the physician-side computer, which

is running the server, any of those devices can access the data from the physician-side computer. So in the combination, the benefits of distributed computing directly come into play. And I talked about that throughout my original declaration, and I talked about it additionally in my reply declarations.

And I just show why what Dr. Goodrich is saying is wrong because he seems to believe that people don't do this, people don't run a web client and a web server on the same machine. In fact, a lot of the management software for managing web servers are in fact run as web clients as well. And even back in the relevant time period, in the 2005 time period, that was the case as well.

Q. And that goes back then to your Paragraph 8 that you speak to of communications from clients and servers known as a local host; is that correct?

A. Like I explained, local host is a shorthand for a specific IP address which sends the message data to that same machine.

Q. So in this Paragraph 8 then, this is how clients and server are then communicating

with one another; is that correct?

A. It's not that they're designed to communicate like that. This is industry standard technology. You just type in local host into the browser. If the browser is on the same machine as the server, you just type in local host and you're able to access that web server on that same machine.

If you have a browser on a different machine you type in the IP address of the physician-side computer that's running the web server and you can equally access the data that is hosted by that web server running on the physician-side computer. This is all industry standard, common -- well-known industry standard technology that has existed from the 1990s.

Q. And you speak to -- in your reply declaration you speak to this as shorthand communication known as local host; that's correct?

A. I state that local host is a shorthand for IP address 127.0.0.1, and that sends the message to that same machine.

Q. Can we go to Paragraph 15 of your '921 reply declaration.

A. Okay. I'm there.

Q. So in this paragraph you say that if clients and server are implemented in the proper way, the principal concern of computer architecture such as security can be addressed. Do you see that?

A. Yes, I do cite to some language that talks about implementing in the proper way, yes.

Q. How would a POSITA implement a clients and server that are on the same machine in a proper way such that security vulnerabilities for example, web server vulnerabilities, would be mitigated?

A. I'm not sure I understand the question. If you have the web browser and the web server on the same machine, you have very few, if any, vulnerabilities. And like I've explained, security is handled by, for example, the use of HTTPS. Again, these are all industry standard technologies. The use of web servers and web pages and web browsers was not new or uncommon. There's a reason why there were web pages and websites for millions of companies, banks, because this was all secure, well-built technology if you did it the correct way.

For example, by using the secure form of HTTP for communications, if you're using standard web browser technologies, standard web server technologies. So these are all well understood industry standard technologies that were used throughout the world for different types of businesses, including banks and governments. And I, in fact, was making my mobile banking technology at that time, and we also used web technologies, and clearly web technologies were secure and safe, even to perform banking transactions.

Q. In your banking transaction example were you working with a database server that was located at a different location than the client devices?

A. There were a large number of client devices that were located remotely, but we also -- like I explained, having a client running on the same server, that's very common. And that's common for doing things like management of the web server, as well as all you have to do was open a web browser on that server where you are running the web server to access the data.

And so, yes, exactly what I'm talking about in my combinations for the patents that we're talking about here, that's precisely what was done in the software industry at that time and even before that time, and I myself, in developing one of the first mobile banking systems in the world, if not the first mobile banking system in the world, used the same secure web technologies to do this.

Q. When you reference management, what do you mean by management? Is that like for testing purposes?

A. It could be for testing. Like you were saying, oftentimes you would simply open up a browser and see that you're able to access the data on the web server, see that it's being displayed correctly. But what I really mean by management is more generically configuring the web server, how you want the web server to behave. Potentially having whitelists or blacklists. So it's more generically managing the web server software and changes to the configuration or data files that the web server might be using.

Q. So the configurations, is this like

all making a web server ready for deployment?

MS. ALEXANDER: Objection. Outside the scope.

THE WITNESS: It's not really about making it ready for deployment only. There's ongoing configurations that you -- that people do when running or managing a web server. So like I said, uploading different entries to the whitelist or the blacklist, that's one example. So there's ongoing management that happens all the time for a web server, and oftentimes these web servers, the management of the web server is done by simply opening a browser, which is a client, and typing in a local host and then you're able to access the management dashboard, and then you are able to make the configuration changes, the management changes that I'm talking about.

BY MS. FULLER:

Q. So going back to the data improving security, are you familiar with SQL injections on databases?

A. I am, but I'm not sure I understood the first part of your question where you said database improving security.

Q. So I want to go back to the line of questioning with improving the security. And particularly how the clients and server are implemented in the proper way to improve the security. And I just want to kind of get an understanding on that there existed SQL injection attacks; correct? Or there exists SQL kind of malicious attacks.

MS. ALEXANDER: Objection. Vague.

THE WITNESS: Can you give me an example of what you're talking about? There can be attacks, but I'm not sure I understand what you're talking about, or why the physician in the combined system that we're talking about would do any such attacks. So I'm not really understanding your question.

BY MS. FULLER:

Q. So SQL injections are web server vulnerabilities; correct?

A. I'm not sure what you mean by it's a web server vulnerability.

Q. Are web servers vulnerable to SQL injection attacks?

A. I don't think web servers in general

are vulnerable to any type of attack. A specific web server that may have poor implementation can have vulnerabilities, but the point that I'm trying to make in Paragraph 15 is that if you implement web server technology and web client technology in the proper way -- and again, these are industry standard technologies. There were implementations available to build -- or to take advantage of existing web server technologies and web browser technologies. So I'm not sure I understand your question about web servers being, in general, vulnerable to any type of attack.

Q. Well, a web server can be injected with a -- or have an SQL injection; correct, where it targets the database?

MS. ALEXANDER: Objection; vague.

BY MS. FULLER:

Q. Does it happen in the art that an SQL injection targets a web server's database?

A. Which database are you running? Which immersion are you running? What is the software? I don't understand this question. And I can't answer it without you giving me details about the architecture, the software,

what the system is doing, what this alleged attacker is trying to do.

Q. What is an SQ injection attack? Can you just provide that to me?

A. Well, this is a high level concept that is like an umbrella term that can -- that says that, you know, you can put data into a database and you can corrupt the database by providing bogus data and potentially overriding existing data.

But the point that I'm trying to make is that if you develop these systems the correct way -- and it's not hard, even back then in 2005 or even before that, these are industry standard technologies. And so -- and then on top of that, the actual combination that we're talking about, we're talking about a web server running on the physician-side computer. So it's a secured computer, as well.

So I don't see how the relevance of what you're talking about comes into the actual combination, the actual system that we're talking about in my declarations.

Q. Well, if an SQL injection corrupts data, then it's getting on the computer that's

running the client and the web server; correct?
MS. ALEXANDER: Objection; vague.
THE WITNESS: Say that again, it's getting on the client --
MS. FULLER: Let me strike that.
BY MS. FULLER:
Q. You just testified to an SQL injection, in broad strokes, being an attack on the data; correct?
A. I explained that it's an umbrella term that could encompass something like that. But I also testified that I don't see the relevance of that to the combination that we're talking about because we're talking about industry standard technologies, which were well known how to implement them properly.
And then on top of that in the specific combination that we're talking about, the web server would be running on the physician-side computer, which would be secured as well. So I don't see how these attacks, not only this kind of attack that you're talking about, but any other attack, would be able to happen in the specific combination that we're talking about.

So that's where the confusion is coming from. You're not really telling me how this attack is happening, why it's happening. But the fundamental point that I'm making in my declaration is that a POSITA would know how to implement industry standard technologies because it's already well known how to implement these technologies using best practices because these are industry standard technologies and industry standard software.

Q. I want to get to two points on that line of questioning. One is you say that the physician-side computer would be secure. Why would it be secure?

A. Well, it's the physician-side computer. So it's meant for the physician to be using. I would think that a POSITA would understand from the disclosure of the Toge reference that a physician-side computer would not be open and accessible to the public, for example, in the middle of a parking lot or in the middle of a mall or something.

I think a POSITA reading the full disclosure of Toge would understand that the physician-side computer is a computer that is

accessible to one or more physicians, and so it would not be accessible to the general public.

Q. The physician-side computer has network access, though; correct?

A. It does have network access, yes.

Q. And with access to the general public network or other devices, doesn't that introduce security issues?

A. It does not. That's my point. If the use of web technologies, even in 2005 or even before that, introduced security issues, then nobody would be -- would have been using web technologies at that time.

And so the whole point I'm trying to make is that there are correct ways of using industry standard technologies. And even in the 2005 and earlier time frame a POSITA would have known those best practices, and regardless, whatever issues there are -- and I think I touch on this in other portions of my reply declaration -- that there would not be these fundamental security vulnerabilities.

And one example to just understand this is that web pages and websites were used throughout the world even at that time, and like

I mentioned, even for performing secure transactions, governments were using it, banks were using it, even in the healthcare space people were using it. And like I said, I was developing technology myself at this time period, and we had the largest banks all around the world license our technology, and they were using it to serve millions of end users with huge numbers of devices accessing the web servers that were running our web server technology, as well as other application server technologies.

So it's just wrong to say that as soon as you have a network connection or as soon as you have a server and a client, that there's vulnerabilities.

Q. But you do admit that vulnerabilities exist at that time?

MS. ALEXANDER: Objection; vague.

THE WITNESS: If you --

BY MS. FULLER:

Q. I'm just trying to understand whether you're saying there's no vulnerabilities whatsoever, or that there are vulnerabilities, but a POSITA would know how to implement the

architecture in the proper way to mitigate.

A. Well, I'm not even sure that what you said in that second part is correct. I don't think that there are vulnerabilities and that a POSITA would know how to mitigate. What I'm trying to say is that there were best practices that were known. And in fact, even my book, which was published in 2004, talks about the best practices in how computers communicate.

So if you use the best practices, if you use encryption technology that existed at that time, and you developed these systems using the industry standard best practices, which were well known and publicized in many books, including my own book from 2004, then you would be able to develop a network-based system that is secure and safe to use. And I provided the example, the fact that there were a multitude, millions of websites that were available and used even at that time period to demonstrate that web technologies, if they're developed the right way, are secure and safe to use.

Q. So then let's drill down, then, on the specific implementation of the physician-side computer implemented as a client and a web

server. Can you give me concrete examples of the proper ways that such implementation would be done to mitigate the security concerns?

A. I think I disagree with your premise of the security concerns. Because you're like -- you're talking about the solution that I'm talking about where you have -- on the physician-side computer you're running the web server, as well as the client. And then you also have the ability to have other devices access that web server.

And so what -- I think I've explained this to you already that if you use standard encryption technologies, for example, HTTPS was well known way before 2005. In fact, my book talks about HTTPS and the encryption that it allows if you use things like proper authentication, for example, having a password and a user name to allow authorized people to get access to the data that the web server has, these were all well known concepts long before 2005, and a POSITA would have known about these concepts. And again, like I mentioned, these are best practices, they were published heavily, including in my own book, and a POSITA would

have implemented these industry standard technologies using the best practices and using even third party software that was available.

And I'll give you another example. Long before 2005 I worked at a company called Bluestone Software, which was acquired by Hewlett Packard. Bluestone developed Java 2 Enterprise Edition application server technology, which incorporates a web server in it.

The reason people use software from Bluestone as well as from other competing companies like BEA and others is that you didn't have to worry about security issues. You would simply license the Bluestone application server or the BEA application server, or even IBM had its own technology called WebLogic, you could just license that software, and that server software, you could run it on your own machine or you could rent a machine and you could run BEA or IBM or Bluestone Software on it, and it handled all of the security authentication, authorization, everything that I'm talking about, all this plumbing was provided for you by these third parties, and all you had to do is

develop your website on top of it. You have to specify, well, I want to have a blue-colored page that has this log-in that asks for your user name and password, what designs you want. That's all you have to do.

Even by 2005 this technology was available, it was available to people in the industry. They didn't have to build it from scratch. They could simply license it and develop a website on it, and the security issues, the authentication issues, authorization issues would all be handled for them using these software that already had these issues worked out.

Q. So what you just spoke to is where a -- you are using the software that is run on a server; correct, and that software then will provide the security?

A. Yes. It provided the plumbing, so the basic concept of having these application servers is -- they were known as middleware because it was understood long before 2005 that whenever you're building anything for the web you have to build these same technologies for security, for authentication, for authorization,

and even other things. Everyone had to build it. So rather than everyone having to build it from scratch, you could simply license technology from BEA, IBM, or Bluestone, and you could just use that software and not worry about the plumbing, like the middleware functionality, because it was common to anyone using web-based technologies.

Q. So a person that is implementing a web page or implementing a web server would then just use this IBM software; correct?

A. Yes, they could license the IBM software. You would have to pay for it. Whereas the Bluestone Software was -- I believe there was a free license. And for support you could pay for more. But yes, it's that the basic middleware was provided by third parties. And again, web technologies are industry standard technologies. These are not some custom technology that somebody is making in their garage. These were industry standard technologies, they were robust technologies that incorporated security, authentication, authorization, all of these other functionalities that we've talked about, and all

you had to do is build whatever functionality you had for your web page, for your website, without having to worry about these underlying security issues.

MS. FULLER: We've been going for about an hour on the record. So let's go off the record, Cheryl. Let's take a five-minute break, Dr. Chatterjee. Does that work?

THE WITNESS: Sounds great.

MS. ALEXANDER: Okay. So 11:05.

(11:00 a.m. - Recess taken.)

(At 11:07 a.m., with parties present as before, the following proceedings were had, to-wit:)

BY MS. FULLER:

Q. Dr. Chatterjee, in the 2005 time frame was it well known to run a web server on a physician's computer?

A. In 2005 and even before that, it was well known to run a web server on any type of computer.

Q. Do you have any examples that you are aware of where a web server was run on a physician computer?

A. So when you say "physician computer," you're talking about the disclosure from Toge of a physician-side computer; correct? That's how I understand your question.

Q. Correct.

A. Yes. So like I mentioned, even at that time period I was developing systems, and we were not only working in banking but also in healthcare as well, because the systems that we were doing in mobile banking were directed to villages in remote areas throughout the world, in countries throughout the world. And so there

was a need for healthcare and telemedicine type of services as well.

And so yes, there were examples of people running web servers on, for example, PCs in a doctor's office, and then people could access that PC based on other devices.

So like I mentioned, long before 2005 it was well known and understood that using web server technology -- because again, like I talked about, there is all of this infrastructure was available, and it was easy to develop custom web pages and websites, and that builds upon this underlying middleware that was available that a lot of people developed custom applications based on these industry standard web technologies, not only in the banking space, but in the healthcare space, even in logistics management.

We were also working on transportation and logistics, and there were PCs that could run web server software in an office, like a trucking type of office. And then the truck drivers could connect into that web server and update it or provide information. And for example, that PC running the web server, it

would capture different information from those different truck drivers and have it available in that central location, which was that PC running that web server software.

Q. Are these computers that you are providing examples on, were they also operating as a personal computer to an end user, or just a computer in the office that was the server?

A. I think there were both. For example, like I talked about for each of those examples, for the medical example, the healthcare example, we're talking about villages, for example, in India or Mexico that they don't have a hundred machines lying around. So this would be a single PC that would be running the web server and then the doctor or like I was mentioning, logistics management, the office, logistics management company owner or somebody, they would be accessing that same computer.

So that same computer that is running the web server was also running other kinds of software, and there was client software, including a web browser that could be run and connect into those as well. This was all very standard technology. And that's the point that

I make, that it's really very surprising that Dr. Goodrich has testified through his declaration that this was not done, or that there were problems with having a client and server on the same machine, because it was used oftentimes -- and I've pointed to documentation that confirms that, and in fact, like I mentioned at the beginning of the deposition, that even Dr. Goodrich's -- the documentation that he himself points to contradicts him where it says that you can run client and server software on the same machine.

Q. Let's go to Paragraph 7 of your declaration where you cite to Exhibit 2020, Page 4. Do you see that?

A. I do.

Q. So in the first parenthetical you cite, In a typical client server based application, the client process and server process can be on the same computer.

Do you see that?

A. I do.

Q. Then it says, The 2-tiered client server structure is the simplest client server structure that is still in use for many

applications today.

Do you see that?

A. I do.

Q. Is a 2-tier client server structure, is that where the client and server are on different machines?

A. They could be on different machines, but they can also be on the same machine.

Q. What is a 2-tier client server structure?

A. Well, if you look at the first quote, it's simply saying that you can run the client process and the server process on the same machine. A 2-tier client server would be that you have a client, which is the user interface, and then you have a server that has both the business logic as well as the data manipulation, and then of course you can have a 3-tier where you have the data manipulation broken out as well.

And so it's simply saying that the 2-tier is the simplest client server structure where you have the client providing the user interface and the server providing the business logic and the data manipulation.

Q. So in the 2-tier, the user interface is on the client and then the business logic is on the server; is that correct?

A. And the data manipulation is also on the server.

Q. And this is the simplest client server structure that's still in use today? I guess we can go to Exhibit 2020 to discuss this.

A. Okay. I'm at Exhibit 2020.

Q. It says, The 2-tier client server structure is the simplest client server structure that is still in use for many applications today. That is on Page 4 of Exhibit 2020 under the 2-tier client server architecture.

A. Okay.

Q. So is this 2-tier client server architecture used for web-based applications where web clients are accessing web servers?

A. I think it can be. I don't think this is -- this paper seems to be talking about server classification. So web servers and web clients are one examples of servers and clients.

So I think to the extent that your question is whether web clients and web servers

can follow what this document is talking about, they can.

Q. So following this document, the 2-tier architecture would be what would be implemented in a web server application such as the Toge system where the client and the server are on different computers?

MS. ALEXANDER: Objection. Vague.

THE WITNESS: I think that's where there's a fundamental misunderstanding or confusion. Having a client on the physician-side computer does not preclude having clients on other devices.

For example, I have a client on my laptop, which is a browser, and I also have a browser on my cell phone. Just because I have a browser on my laptop does not preclude the fact that I have or I can have a browser on my cell phone.

And so I can run a web server on my laptop and have a web client on my laptop and still have a web client on my cell phone, and my wife can have a web client on her laptop and her cell phone and they all, all of those web

clients, can access the web server running on my laptop.

BY MS. FULLER:

Q. And this would be in a web-based architecture that this would be allowed?

A. I'm not sure what you mean by "this would be allowed," but yes, you can use web technologies -- web technologies simply refers to industry standard protocols like HTTP, HTML. So it's just talking about these protocols, and HTML -- yeah, I think I mentioned HTTP, HTML.

So there could be other such client server architectures that you can develop as well that follow that same logic of what I explained of having a server and a client running on my laptop, and having clients run on my cell phone, my wife's laptop, my wife's cell phone, and they can all access the server on my laptop.

So one example of that scenario is using web technologies, but even nonweb technologies can use that. But for purposes of my declarations and the combinations that I'm talking about with regards to Toge, together with the knowledge of a POSITA and Toge with

Kumar, we are talking about web-based technologies, and yes, this was all commonly used, well-known concepts and technologies.

Q. And going back to your Paragraph 8 of your '921 reply declaration, that's where you use the example of this is shorthand known as a local host; is that correct?

A. I did talk about local host in my reply declaration, yes.

Q. So for a client to -- client browser to access a web server on the same machine, it would just do HTTP local host; is that correct?

A. They would use HTTPS://localhost and then they can even put a forward slash and put in other path entries as well, or they can just do local host and that will hit the web server running on the default port. So yes, that's all they have to do.

Q. Okay. So that's all they had to do for a client device to access a web server running on a computer; is that correct?

A. On that same computer, and at the same time other computers could simply type in HTTPS:// and use the IP address of that first computer, and they could also access the web

server as well.

Q. So they would have to type in that IP address of the computer to be able to access the server on that machine?

A. Well, that's how Internet communications works, it uses TCP/IP. You could use URL, which gets resolved to an IP address. You don't have to type in an IP address per se. You can type in a URL, like CNN.com or Google.com. Those get resolved. But URL is simply an easy way to remember a URL. And then those get resolved to an IP address.

Q. So if a physician computer is operating behind -- in a private network, a device that's outside the private network in the public network, they would have to know the IP address of the computer to be able to access the web server operating on that computer?

A. No, they would not. In fact, they will not know because that's on a private network. And so this is what we talked about when we talked about the NAT earlier, that a NAT or some other type of router would simply convert the incoming IP address or URL, and then it would simply route it to the web server,

whatever the IP address was. And that's what these home routers or NATs do. And that's precisely what I explained to you in the morning. Like even my home, I can literally run a web server on my laptop, connect it to my network within my home, which is on a private network, and you can access my web server by simply typing in the URL or the IP address, and the router, including the NAT on my home router, would forward that request to the web server.

Q. Can we go to Paragraph 20 of your declaration.

A. Okay. I am at Paragraph 20.

Q. So in this paragraph you state, When the physician's web client is active, it could continuously query the server to see if new data has been received. When you say active, are you meaning that there is an active session?

A. I'm just saying when the physician is using it, as meaning when it's active.

Q. So when the physician is logged into the website and the application is open; is that correct?

A. What do you mean that the application is open?

Q. That there's a connection between the web server and the client browser.

A. Well, this is talking about it could continuously query the server. So it's not talking about leaving that connection open, but I think other parts of my reply declaration talk about you -- implementing push-type data, and that was also well known long before the 2005 time period, that you could keep that connection open and then the server would provide the data as soon as new data came in.

Q. In the HTTP/1 protocol at the time of the 2005 invention, did that terminate after a certain amount of time?

A. That's simply based on configuration. This is what we talked about earlier about managing the web server. You could connect using, for example, a client running on that machine that's running the web server, to manage the web server. And you could configure how long to keep that connection open, and then like I explained, that even with HTTP/1 the concept of long polling was well known and that would simply -- the client would make a request to the server, and then the server would wait until new

data comes in, and then it would respond to that client request based on the new data. And then as soon as that new data was delivered to the client, the client would make a new request. And so that technology was well known even at HTTP/1.0.

So like I mentioned in my reply declaration, there were different ways that a POSITA would have known, even at 2005 and even earlier, how to get this kind of data to the client.

Q. Did web servers in the 2005 time frame have a push application available to them to push the data to a client device on another machine?

A. I'm not sure what you mean by "push application available to them." But the technology, like I just explained to you, where you're -- the client is making a request to the server, then that connection remains open, and then as soon as new data comes into the server, the server immediately provides that new data to the client. And that certainly existed at HTTP/1.0.

Q. But going to what you just spoke to is

that there had to be an active -- the web client is active to be able to receive that new data; is that correct?

A. Well, I'm not sure like what you're saying by active. Like I was mentioning what I mean by active is like, for example, the physician sitting down in front of the computer. But it doesn't have to be like that. Like I just mentioned with regards to the long polling technology, which was known at 2005 and even before that, what would happen is that the client software would simply make a request to the server, the server would not respond immediately, the server would respond whenever new data comes in, and then it would send that new data to the client, and then in response to that, the client would again make an -- immediately make another request. So that was a way of doing effectively push technology using the pull of HTTP.

And then I also talk about other ways that a POSITA would have known, for example, continuously polling as well.

Q. So the effective way to push using HTTP would be a long pull, is that what you're

saying?

A. One example of that, yes.

Q. Where there's a request for the data, and then there is a wait until new data comes in? I'm just trying to get an understanding of what a long pull is.

A. Yes. Long polling, it's where the client makes a request of the server, the server waits to respond until new data comes in. And then when that new data comes in, it immediately sends it to the client, based on that connection, and then as soon as a client receives it, it makes another request to the server.

And so clearly you can understand that in that concept, again, which was known in 2005 and even before that, the client would be receiving the data as soon as it comes in.

Q. And to receive the data, there's an active session between the client and the web server; correct?

A. In long polling there would be the active connection. And if you look at how push technologies work even now, that's the same way it works. Again, I think we've talked about

this technology is not magic. You have to have a connection between the client and the server in order to even get a push. So it works the same way even now where you have that connection that's available, and then when there's new data that comes in, it's provided to the client.

Q. Upon a request for that data, just delivered at a later time; correct?

MS. ALEXANDER: Objection. Mischaracterizes his testimony.

THE WITNESS: Are you talking about how --

BY MS. FULLER:

Q. The long pull?

A. -- the push --

Q. Your long pull testimony. Long polling.

A. Yeah, for long polling, it's not delivered at a later time, it's delivered immediately upon the new data coming in.

Q. So there's a request for the data and then it's delivered when the new data comes in?

A. Yes. And then the client immediately turns around and makes a new request, and then the same thing happens again. And this was one

way, even back in 2005 and before that, to have data delivered immediately, new data delivered immediately, as soon as it came in.

Q. What were the reasons why a client device would implement a long polling request?

A. Well, just for that same exact reason, to get data when it comes in. For example, if you have a website that is aggregating data or aggregating information when new data comes in, you would want the client to get that.

And so even back then there were news sites that -- like, for example, portals that would aggregate news and other types of website data. And so long polling was used to allow those kind of updates to be sent out immediately as soon as the new data became available.

Q. Dr. Chatterjee, can we turn to Paragraph 24 of your '921 declaration.

A. Okay. I'm there.

Q. You know what, I'm going to have you back up for a second. I'm going to have you go to Paragraph 22.

A. 22?

Q. Yes.

A. Okay.

Q. In writing your opinion, did you apply or come to an understanding of what the plain and ordinary definition of retransmit is?

MS. ALEXANDER: Objection. Calls for a legal conclusion.

THE WITNESS: So say that again. Because this section I'm talking about, what Dr. Goodrich is stating, I'm explaining that it's wrong, but then even if you adopt that wrong understanding, it would still be obvious, based on references and the knowledgeable of the POSITA that I talk about.

BY MS. FULLER:

Q. In writing your declaration, did you apply a definition to the term retransmit?

A. I think we -- that it would be understood that you're simply transmitting whatever you're retransmitting. You're simply getting it and you're transmitting it.

Q. And so are you saying that transmitting and retransmitting are one and the same?

A. No. I'm saying that within the -- what we're looking at with regards to the claims, it would be that you're receiving it and

then you're retransmitting it. So even the retransmit is a transmit. But it's simply saying that you're getting it and then you're retransmitting it, or transmitting it.

Q. So when a device gets it and then retransmits it, is that acting as an intermediary device to forward the data on to an end device?

A. So what do you mean by acting as an intermediary? It could. It's getting the data and then it could do something with that data and then it could send it, again. So I'm not really sure what you mean by an intermediary.

Q. It's an intermediary device that receives the data -- first receives the data, and then just pushes the data onward to an end device. That's what I mean by an intermediary device.

A. So I'm not really sure -- like when you're talking about pushes the data, I don't think retransmit is limited to pushing data. It simply means to a POSITA that you're getting that data, and that you could do something with that data, and then you're sending that data.

Q. And when you're sending that data, are

you waiting for a device to request that data or are you just automatically pushing it on?

A. So I think that the term retransmit is not so limiting that it would be one of those. I think retransmit could include different ways of sending that data forward, like I talked about. And that's precisely what I've talked about in this section of my reply dec. where I say that Dr. Goodrich's understanding of the term seems to be overly restrictive and wrong, but that even if you assume that overly restrictive and wrong understanding, it's still obvious based on the combinations that I talk about in my declarations.

Q. Well, you testified earlier that to -- in your first deposition that retransmit is -- pardon me, give me one second -- is receiving the data and then sending it to another device; correct?

A. I believe that's consistent with what I just explained to you, yes.

Q. And it's different from transmit, where in transmitting, a device first receives the data -- pardon me, where it's transmitting a device, transmits the data by itself.

A. Are you reading --

Q. Let me rephrase that. Retransmitting, you testified, is receiving data and then sending to another device, and you stand by that testimony today; correct?

A. And I believe what I explained to you right now is consistent with that, yes.

Q. And then transmitting, a device transmits the data by itself. You testified to that in your first deposition.

MS. ALEXANDER: Objection. Outside the scope. Jenna, do you have something you'd like to point Dr. Chatterjee to in terms of this --

MS. FULLER: We can open Exhibit 2018 if that's available to you. If not, I can upload it.

THE WITNESS: Okay. What page on 2018?

MS. FULLER: Page 38, Line 24 through 25.

THE WITNESS: So it seems like you are asking me outside of the '921 with regards to this.

BY MS. FULLER:

Q. I'm just trying to see if you stand by your testimony, your first deposition testimony today, that in transmitting -- that transmitting is receiving -- pardon me. Transmitting, a device transmits the data by itself. That was your testimony in your first deposition; is that correct?

MS. ALEXANDER: Objection. Mischaracterizes his testimony.

THE WITNESS: Yeah, I mean, if you read Page 38, starting at Line 11, I'm saying, well, again, I'm understanding your question to be limited to this sentence in a vacuum, and as I've explained to you a couple of times, I've not memorized everything in the '921, and if there is other disclosure about that question.

And so I think I answered for you at my first deposition that it was only based on that one sentence, I guess, that you were pointing me to, and in a vacuum, not remembering everything that's in the '921.

But again, to the extent your question right now is whether I stand by my testimony in the first deposition? I a thousand percent

stand by it.

Q. So when a device is retransmitting, it first receives data from another device, and then retransmits it to the end device; is that correct?

A. That could be one example, and again, in a vacuum, based on what we've been talking about with regards to what I testified to in my first deposition.

Q. What are other examples, beyond the one we just spoke of?

A. Well, I think you talked about receiving it from another device, and we've been spending a lot of time this morning talking about how you can have different software running on to the same device, and how it was well known.

So you can even receive data from a different piece of software running on that same device and then potentially retransmit it as well. So that would be another example.

Q. And in retransmitting it, the device is simply forwarding it to the end device; correct?

A. I think that's one example. But I

think, like we've talked about, and if you look at the '921, there's clear disclosure even in the '921 that there could be processing of the data. And so even -- if you're coming back to the '921 and what's -- what a POSITA would understand in the context of the '921, it's not limited to simply getting it and forwarding it. Because like I just explained, even in the '921 there is disclosure of optional processing of the data.

Q. So where a device gets it in -- let's try this example -- processes the data, after it processes the data, it then would forward it to the end device?

A. It could. That's one example. There are a lot of different examples. But yes, it could forward it. But even when you say forward it, it's kind of unclear because how technology works is that you're getting these electrical signals that are coming in, and then you're not forwarding those electrical signals. The computer will process those electrical signals that are coming in, it will take them from the analogue signals that are coming in, digitize them, and then, you know open the packet up,

look what's inside the packet, and then it would generate new electrical signals to send it out.

So it's not like -- it's not how computers work, electronic computers work is how I explained it, that you have these signals that are coming in and then signals that might be going out, and there could be some delay or lag in between the signals coming in or going out, and -- but it's not -- like I just want to be clear when you're using the term forward it, it's not just taking those same electrical signals that are coming in. Because those electrical signals are terminated when they get to that device. Or if they're not electrical signals, if it's on a Wi-Fi, there are RF signals that are coming in and they're terminated and new signals are generated.

Q. So these new signals that are generated, that's what is being then forwarded on to the end device, if I understand what you're saying?

A. Yes. For example, the packets that might go over a physical Ethernet network would be generated, and they would go over the network to their destination.

Q. So in retransmitting, a device receives data from another device, can process it or not process the data, and then forward on processed or unprocessed data to another device; is that what you're testifying to?

MS. ALEXANDER: Objection. Mischaracterizes his testimony.

THE WITNESS: So I think I just explained to you -- I want to be clear about -- you keep on using the term "forward it," and I want to explain to you, and I think I did explain to you, that how things work in reality with technology, that you're getting the signals, you're terminating the signals, you're performing processing potentially, and then you're generating new signals, either electronic signals or RF signals or optical signals or something, and that's one example.

But you can also have that data sit there for some time and then send it on. There's nothing in the term retransmit, especially in the context of the '921. But as I explained, even if you take Dr. Goodrich's incorrect and overly limiting understanding of the term, it's still disclosed by -- it's still

obvious in view of the combinations that I talked about in my declarations.

BY MS. FULLER:

Q. So your opinion that it's still obvious, even under the construction, is based upon processing of the data; is that correct? I'm looking at Paragraph 25 of your '921.

A. On 25 I'm simply stating that with respect to processing data sent or stored is the same data received. And so I'm explaining similar to what I explained for you that when the data comes in, the packet is inspected and then different things can happen to the actual payload, the actual data within the packet. It can be stored and then later retransmitted. It can be -- like I explained to you with regards to long polling, the data can come in and then it's immediately provided in response to long polling. So there are a lot of different ways and things that can happen to that data. I'm simply explaining how -- what actually happens with technology in real life.

Q. Okay. And then Paragraph 26 you alluded to that it is your automatically push data that has been received, where the data

would come in and then be pushed to another device?

MS. ALEXANDER: Objection; vague.

THE WITNESS: Well, what I'm stating in Paragraph 26 is that the Toge/Kumar system is not limited to a pull system and could be implemented to automatically "push data that has been received."

And I think I've explained to you in quite a bit of detail today why even in 2005 a POSITA would have understood that and known that, for example, with long polling.

Q. Was push protocols for web servers available in 2005?

A. What do you mean by push protocols? Like are you distinguishing between what I just explained?

Q. Like a push protocol in an HTTP/2. HTTP/2 has a push protocol where data is sent to a server and can be pushed to an end device.

A. So that protocol is different, but in terms of pushing data, to the extent you're talking about pushing data meaning that as soon as that data is available it's provided to one or more clients, I explained to you in quite a

bit of detail that the long polling achieves that same thing. Alternatively, I also explained to you that continuous polling achieves that same thing.

So they're different protocols, but they can achieve the same end result of providing data to clients immediately upon the server receiving.

Q. And then long polling, there is a client request for the data; correct?

A. And even in all push protocols, there is a connection that has to be open. Maybe you're having a misunderstanding about how push protocols work, even in 2026. You have to have a connection between the client and the server in order for the data to be, quote unquote, pushed.

Q. I was not speaking to a connection, I was speaking to the request in a long polling, there's a request for the data by the client; correct?

A. I'm not sure what you mean by a request for the data, even in push -- even in modern 2026 push protocols you have to identify, I want this data. It's not that the server

could magically figure out that you want some data and somebody else wants some other data. These are simply different protocols, but they achieve the same end result of getting data to clients immediately upon being received by the server.

Q. And in your querying, the web server example that you also spoke to, that's where the web browser client is active on the computer; is that correct, for the querying?

MS. ALEXANDER: Objection. Mischaracterizes his testimony.

THE WITNESS: So like I explained to you, when I use the term active, I meant, as an example, the physician sitting in front of the computer and doing some work, but it doesn't even have to be like that. Like I've explained to you, the long polling, it works by the data is received and then immediately the client in long polling will send another request. And so the physician doesn't even have to be sitting there in front of the computer. The data would show up using long polling, or even continuous polling.

BY MS. FULLER:

Q. So in the long polling, a client would send a GET request to the server, and then the server would send the data when it shows up?

A. That is one example, yes, of using long polling.

Q. Are there other examples?

A. Well, like I explained to you, there is continuous polling, where the client can issue a GET request and then it will get that data and then it will again issue another GET request. So it's a continuous polling example.

Q. Are there HTTP time-outs for long polling that would terminate the request?

A. So again, like I explained already, I think I answered this question before, where I explained that these are configurable. And so for -- and a POSITA knew that these were configurable, and if the time-out is set to a certain amount, when the time-out happens, the client simply sends a new request.

So with long polling, like I said, the end result is that the client gets the data as soon as a server has it, and it's simply using one protocol, one mechanism by which to do it, continuous polling is another way, and then, you

know, there are different types of protocols that are used, but they all achieve the same request -- the same end result of getting the data to the client as soon as the server has it.

MS. FULLER: Dr. Chatterjee, let's go off the record.

(Discussion was had off the record.)

MS. FULLER: Let's do a quick five-minute break.

(11:57 a.m. - Recess taken.)

(At 12:02 p.m., with parties present as before, the following proceedings were had, to-wit:)

MS. FULLER: Dr. Chatterjee, thank you for your time today. I have no further questions.

THE WITNESS: Thank you.

MS. ALEXANDER: I have no further questions.

COURT REPORTER: (Requests orders for the record.)

MS. ALEXANDER: We'll take a rough. Standard turnaround.

MS. FULLER: Thank you all.

(2:02 p.m. - Adjournment.)

** ** ** **

C E R T I F I C A T E

STATE OF NEBRASKA)
) Ss.
COUNTY OF DOUGLAS)

I, Cheryl A. Rooney, Registered Professional Reporter, General Notary Public within and for the State of Nebraska, do hereby certify that the foregoing testimony of SANDEEP CHATTERJEE, Ph.D. was taken by me in shorthand and thereafter reduced to typewriting by use of Computer-Aided Transcription, and the seventy-eight (78) pages contain a full, true and correct transcription of all the testimony of said witness, to the best of my ability;

That I am not a kin or in any way associated with any of the parties to said cause of action, or their counsel, and that I am not interested in the event thereof.

IN WITNESS WHEREOF, I hereunto affix my signature and seal this 29th of January, 2026.

Cheryl A Rooney

State of Nebraska – General Notary
CHERYL A. ROONEY
My Commission Expires
November 24, 2026

CHERYL A. ROONEY, RPR
GENERAL NOTARY PUBLIC
Registered Professional Reporter

My Commission Expires: November 24, 2026

WORD INDEX

< 0 >
00160 4:*14*

< 1 >
1 3:*2*
10 17:*9*
10036 2:*14*
11 66:*11*
11,375,921 1:*3*
11,786,680 1:*5*
11:00 44:*11*
11:05 44:*10*
11:07 45:*1*
11:57 76:*10*
1177 2:*13*
12:02 77:*1*
127.0.0.1 27:*22*
15 12:*5* 17:*9* 27:*24* 33:*4*
1990s 27:*16*

< 2 >
2 3:*2* 22:*8*, *11* 41:*7*
2:02 77:*15*
20 12:*4* 55:*11*, *13*
2004 39:*8*, *15*
2005 26:*15* 34:*13* 37:*10*, *17* 40:*15*, *22* 41:*5* 42:*6*, *22* 45:*5*, *8* 46:*7* 56:*8*, *13* 57:*9*, *12* 58:*10* 59:*16* 61:*1* 72:*10*, *14*
2018 65:*16*, *19*
2020 19:*6*, *14* 21:*25* 22:*1* 48:*14* 50:*8*, *9*, *14*
2024-00159 4:*14*
2025-00159 1:*3*
2025-00160 1:*4*
2026 1:*17* 73:*14*, *24* 78:*20*, *24*
20ish 6:*21*
212.715.7502 2:*14*
22 61:*22*, *23*
24 61:*18* 65:*20* 78:*24*
25 65:*21* 71:*7*, *8*
26 71:*23* 72:*5*
27th 1:*17*
29th 78:*19*
2-tier 49:*4*, *9*, *14*, *22* 50:*1*, *10*, *14*, *17* 51:*3*
2-tiered 48:*23*

< 3 >
3 3:*3* 7:*3*
350 2:*7*
38 65:*20* 66:*11*
3-tier 49:*18*

< 4 >
4 3:*3*, *4* 21:*24* 22:*1* 48:*15* 50:*13*
443 16:*12*, *18*
4655 2:*7*

< 5 >
5 10:*20* 11:*4* 12:*21*, *23*

< 6 >
680 4:*17*, *20*

< 7 >
7 48:*13*
78 3:*4* 78:*11*

< 8 >
8 15:*3* 16:*14* 26:*18*, *24* 53:*4*
80 16:*11*, *18* 18:*11*, *13*
858.458.3000 2:*8*

< 9 >
9:58 1:*16*
921 4:*16*, *19* 6:*23*, *25* 10:*20* 15:*4* 27:*24* 53:*5* 61:*18* 65:*23* 66:*16*, *22* 68:*2*, *3*, *5*, *6*, *8* 70:*22* 71:*7*
92121 2:*8*

< A >
a.m 1:*16* 44:*11* 45:*1* 76:*10*
ability 40:*10* 78:*13*
able 21:*6*, *10* 24:*4* 27:*7* 30:*15* 31:*15*, *16* 35:*23* 39:*16* 54:*3*, *17* 58:*2*
about 7:*21* 9:*23*, *24* 11:*17* 12:*21* 14:*16*, *23* 16:*14*, *20* 20:*19* 22:*18* 23:*9*, *16*, *22*, *25* 24:*2*, *6*, *8*, *25* 25:*1*, *2*, *3*, *13*, *15*, *19*, *20* 26:*5*, *6* 28:*8* 30:*2*, *3* 31:*5*, *18* 32:*12*, *14*, *15* 33:*11*, *25* 34:*17*, *21*, *23* 35:*13*, *14*, *18*, *23*, *25* 39:*8* 40:*6*, *7*, *16*, *22* 41:*14*, *24* 43:*5*, *25* 44:*3*, *6* 45:*15* 46:*10* 47:*10*, *12* 50:*21* 51:*1* 52:*10*, *24* 53:*1*, *8* 54:*21*, *22* 56:*3*, *5*, *7*, *16* 58:*21* 59:*25* 60:*12* 62:*7*, *12* 63:*20* 64:*7*, *8*, *14* 66:*16* 67:*8*, *12*, *15* 68:*1* 70:*9* 71:*2* 72:*23* 73:*13*
above 12:*22* 17:*9* 22:*8*
accept 16:*11*
access 11:*24* 14:*25* 20:*19*, *22* 21:*6*, *10* 23:*13* 24:*4* 25:*12* 26:*2* 27:*7*, *12* 29:*24* 30:*15* 31:*15* 37:*4*, *5*, *6* 40:*11*, *20* 46:*6* 52:*1*, *18* 53:*11*, *20*, *25* 54:*3*, *17* 55:*7*
accessible 36:*20* 37:*1*, *2*
accessing 25:*21* 38:*9* 47:*19* 50:*19*
achieve 73:*6* 74:*4* 76:*2*
achieves 73:*1*, *4*
acquired 41:*6*
acting 63:*6*, *9*
action 78:*16*
active 55:*15*, *17*, *18*, *20* 58:*1*, *2*, *5*, *6* 59:*20*, *23* 74:*9*, *14*
actual 24:*7* 34:*16*, *21*, *22* 71:*13*, *14*
addition 11:*16* 13:*13*, *21*
additionally 26:*7*
address 8:*2*, *11*, *22* 26:*22* 27:*10*, *22* 53:*24* 54:*3*, *7*, *8*, *12*, *17*, *24* 55:*1*, *8*
addressed 7:*16*, *19*, *22* 8:*5*, *10*, *18*, *20* 28:*5*
Adjournment 77:*15*
admit 38:*17*
adopt 62:*9*
advantage 20:*24* 21:*3*, *14* 22:*22*

23:*3* 33:*9*
advantages 20:*16*
affix 78:*18*
after 22:*15* 56:*13* 68:*12*
again 7:*24* 11:*2* 17:*13* 24:*24* 28:*19* 33:*7* 35:*3* 40:*23* 43:*18* 46:*9* 58:*17* 59:*16*, *25* 60:*25* 62:*6* 63:*12* 66:*12*, *23* 67:*6* 75:*10*, *14*
aggregate 61:*13*
aggregating 61:*8*, *9*
agree 7:*10*
ALEXANDER 2:*6* 8:*24* 9:*19* 12:*7*, *18* 13:*7*, *16* 15:*14*, *20* 19:*8* 20:*9* 21:*7*, *16* 24:*15* 31:*2* 32:*9* 33:*17* 35:*2* 38:*19* 44:*10* 51:*8* 60:*9* 62:*4* 65:*11* 66:*8* 70:*6* 72:*3* 74:*11* 77:*8*, *12*
alleged 34:*1*
allow 11:*23* 14:*25* 15:*11* 17:*22* 20:*19* 40:*19* 61:*14*
allowed 52:*5*, *7*
allows 40:*17*
alluded 71:*24*
already 7:*25* 36:*7* 40:*13* 42:*13* 75:*14*
Alternatively 73:*2*
Americas 2:*13*
amount 56:*14* 75:*19*
analogue 68:*24*
analyses 9:*13* 10:*15* 17:*14*
analysis 7:*12* 9:*25* 10:*14*
another 15:*13* 17:*16*, *17* 27:*1* 41:*4* 57:*14* 58:*18* 59:*13* 64:*18* 65:*4* 67:*3*, *13*, *21* 70:*2*, *4* 72:*1* 74:*20* 75:*10*, *25*
answer 33:*24*
answered 66:*18* 75:*15*
any 5:*24* 8:*15* 25:*4* 26:*1* 28:*17* 32:*16* 33:*1*, *12* 35:*23* 45:*9*, *11* 78:*14*, *15*
anyone 16:*24* 43:*7*
anything 42:*23*
APPEAL 1:*2*
APPEARANCES 3:*2*
Appearing 2:*2*, *3*, *9*
application 22:*6*, *16*, *19*, *20* 23:*2*, *5*, *19* 38:*11* 41:*8*, *15*, *16* 42:*20* 48:*19* 51:*5* 55:*22*, *24* 57:*13*, *17*
applications 46:*15* 49:*1* 50:*13*, *18*
apply 62:*1*, *15*
approved 12:*11*
architecture 28:*5* 33:*25* 39:*1* 50:*15*, *18* 51:*4* 52:*5*
architectures 52:*13*
are 4:*6*, *10* 6:*7* 7:*13* 9:*6*, *9*, *13* 10:*4*, *14*, *16*, *17* 12:*20* 15:*8*, *9*, *16*, *19* 17:*15*, *18* 19:*9* 20:*16* 21:*13* 22:*8* 23:*20* 24:*4* 25:*21* 26:*13*, *25* 28:*3*, *10*, *19* 29:*4*, *24* 31:*16*, *21* 32:*3*, *19*, *23* 33:*1*, *7*, *21*, *22* 34:*14* 36:*9* 37:*15*, *19* 38:*24* 39:*4*, *22* 40:*24* 42:*16* 43:*18*, *19* 45:*11* 47:*5* 49:*5* 50:*19*, *23* 51:*6* 53:*1* 55:*17* 60:*11* 62:*20*, *21* 63:*25* 64:*2* 65:*1*, *23* 67:*10* 68:*16*, *20*, *23*, *24* 69:*6*, *12*, *13*, *15*, *16*, *17*, *18* 71:*19* 72:*16* 74:*3* 75:*6*, *12*, *16* 76:*1*, *2*
areas 45:*24*
arguments 7:*6*, *10*, *11*, *17*, *18* 10:*14*
art 33:*19*
aside 5:*3*
asked 8:*5*, *18*, *19* 9:*22* 12:*21*
asking 65:*23*
asks 42:*3*
aspect 14:*15*
assertions 7:*21*
associated 78:*15*
assume 64:*11*
attack 33:*1*, *13* 34:*3* 35:*8*, *22*, *23* 36:*3*
attacker 34:*2*
attacks 32:*7*, *8*, *13*, *16*, *24* 35:*21*
attention 15:*22*
authentication 40:*18* 41:*22* 42:*11*, *25* 43:*23*
authorization 41:*23* 42:*11*, *25* 43:*24*
authorized 40:*19*
automatically 18:*13* 64:*2* 71:*24* 72:*7*
available 4:*24* 5:*11*, *20* 6:*24* 19:*6* 33:*8* 39:*19* 41:*3* 42:*7* 46:*11*, *14* 47:*2* 57:*13*, *17* 60:*5* 61:*16* 65:*16* 72:*14*, *24*
Avenue 2:*13*
aware 45:*12*

< B >
back 18:*4* 19:*9* 26:*14*, *17* 31:*20* 32:*1* 34:*13* 53:*4* 61:*1*, *11*, *21* 68:*4*
ballpark 6:*11*
banking 29:*9*, *12*, *13* 30:*6*, *8* 45:*21*, *23* 46:*16*
banks 28:*24* 29:*7* 38:*2*, *6*
based 12:*1* 14:*6* 23:*16* 46:*6*, *15* 48:*18* 56:*15* 57:*2* 59:*11* 62:*11* 64:*13* 66:*19* 67:*7* 71:*5*
basic 8:*16* 10:*5* 42:*20* 43:*17*
BEA 41:*13*, *16*, *21* 43:*4*
beginning 1:*16* 17:*13* 48:*8*
behave 30:*20*
believe 5:*14* 7:*23* 8:*4*, *5*, *13*, *21* 9:*15* 14:*17* 26:*9* 43:*14* 64:*20* 65:*6*
believes 10:*16*

benefits 14:*17* 26:*3*
BEO 11:*5*
best 36:*8* 37:*18* 39:*6*, *9*, *10*, *13* 40:*24* 41:*2* 78:*13*
better 15:*18*
beyond 67:*10*
binder 5:*7*, *11*, *15*, *20*
bit 15:*17* 72:*10* 73:*1*
blacklist 31:*9*
blacklists 30:*21*
blue-colored 42:*2*
Bluestone 41:*6*, *7*, *12*, *15*, *21* 43:*4*, *14*
BOARD 1:*2*
bogus 34:*9*
book 39:*7*, *15* 40:*15*, *25*
books 39:*14*
break 12:*16* 44:*8* 76:*9*
bridge 5:*18*
brief 8:*3*
bringing 15:*22*
broad 35:*8*
broken 49:*19*
browser 15:*12* 16:*15* 18:*1*, *8*, *9*, *12* 19:*2* 24:*11* 25:*7*, *23* 27:*5*, *9* 28:*15* 29:*3*, *23* 30:*15* 31:*14* 33:*10* 47:*23* 51:*16*, *17*, *18*, *19* 53:*10* 56:*2* 74:*9*
browsers 28:*21*
build 33:*8* 42:*8*, *24* 43:*1*, *2* 44:*1*
building 42:*23*
builds 46:*13*
built 9:*5*
business 23:*7*, *23* 24:*13*, *17*, *18* 49:*17*, *24* 50:*2*
businesses 29:*7*

< C >
CA 2:*8*
call 4:*19*
called 16:*23* 17:*22* 41:*5*, *17*
Calls 62:*4*
can 5:*20* 6:*22* 8:*15* 10:*19* 15:*3* 17:*11* 19:*15* 20:*18*, *19* 22:*14* 24:*4* 25:*7*, *8*, *10*, *23* 26:*1* 27:*12*, *24* 28:*5* 32:*11*, *12* 33:*3*, *14* 34:*3*, *6*, *7*, *8* 40:*1* 48:*11*, *20* 49:*8*, *12*, *18* 50:*8*, *20* 51:*1*, *2*, *19*, *21*, *24* 52:*1*, *7*, *13*, *18*, *22* 53:*14*, *15* 54:*9* 55:*4*, *7*, *11* 59:*15* 61:*17* 65:*15*, *17* 67:*15*, *18* 70:*2*, *19* 71:*13*, *15*, *16*, *17*, *20* 72:*20* 73:*6* 75:*8*
cannot 23:*2* 25:*4*, *21*
can't 33:*24*
CAPTION 3:*2*
capture 47:*1*
CASE 1:*3*, *4* 3:*2* 15:*6* 26:*16*
cause 78:*15*
cell 51:*17*, *19*, *23*, *25* 52:*17*
central 47:*3*
certain 7:*5*, *10*, *16* 56:*14* 75:*19*
certainly 57:*23*
CERTIFICATE 3:*4*
certify 78:*7*
changes 30:*22* 31:*17*, *18*
chat 5:*21*
CHATTERJEE 1:*7*, *14* 2:*3* 4:*1*, *6* 6:*22* 19:*5* 44:*8* 45:*5* 61:*17* 65:*13* 76:*5* 77:*4* 78:*8*
Cheryl 1:*14* 44:*7* 78:*4*, *21*
cite 28:*7* 48:*14*, *18*
cites 17:*7*, *8*
citing 17:*10*
claims 62:*25*
classification 50:*22*
clear 14:*8* 68:*2* 69:*10* 70:*9*
clearly 9:*7* 10:*10* 24:*3* 29:*10* 59:*15*
CLEVELAND 1:*6*
client 10:*25* 11:*9*, *11*, *21* 13:*14*, *21* 14:*23* 15:*6*, *8*, *11* 16:*3*, *14*, *16*, *21* 17:*19* 18:*1*, *3*, *7* 19:*2* 20:*21* 21:*5*, *10*, *18* 22:*2*, *6*, *13*, *16*, *18* 23:*2*, *10*, *11*, *19*, *20* 24:*3*, *11* 25:*1*, *3*, *4*, *6*, *9*, *16*, *23* 26:*11* 29:*15*, *17*, *19* 31:*14* 33:*6* 35:*1*, *4* 38:*15* 39:*25* 40:*9* 47:*22* 48:*4*, *11*, *18*, *19*, *23*, *24* 49:*4*, *5*, *9*, *12*, *14*, *15*, *22*, *23* 50:*2*, *6*, *10*, *11*, *14*, *17* 51:*6*, *12*, *15*, *22*, *23*, *24* 52:*12*, *15* 53:*10*, *20* 55:*15* 56:*2*, *18*, *24* 57:*2*, *4*, *11*, *14*, *19*, *23* 58:*1*, *12*, *16*, *17* 59:*8*, *11*, *12*, *17*, *20* 60:*2*, *6*, *23* 61:*4*, *10* 73:*10*, *15*, *20* 74:*9*, *19* 75:*1*, *8*, *20*, *22* 76:*4*
clients 17:*5*, *11*, *23* 24:*4* 25:*10*, *21* 26:*14*, *19*, *25* 28:*3*, *9* 32:*3* 50:*19*, *23*, *25* 51:*14* 52:*1*, *16* 72:*25* 73:*7* 74:*5*
clipped 15:*16*
CNN.com 54:*9*
coffee 5:*6*
colleague 8:*19* 9:*22*
colon 18:*11*, *12*
combination 23:*16* 24:*1*, *8* 25:*13*, *14* 26:*3* 34:*16*, *22* 35:*13*, *18*, *24*
combinations 30:*2* 52:*23* 64:*13* 71:*1*
combined 32:*15*
come 6:*8* 26:*4* 62:*2* 71:*17* 72:*1*
comes 34:*21* 57:*1*, *21* 58:*15* 59:*4*, *9*, *10*, *18* 60:*6*, *22* 61:*7*, *9* 71:*12*
coming 36:*2* 60:*20* 68:*4*, *20*, *23*, *24* 69:*6*, *8*, *12*, *16*
Commission 78:*24*
common 11:*15* 16:*22* 17:*21* 27:*15* 29:*20*, *21* 43:*7*
commonly 53:*2*
communicate 15:*12* 27:*3* 39:*9*

communicating 26:*25*
communication 16:*25* 18:*5* 27:*19*
communications 15:*8, 9* 16:*3* 17:*23* 26:*18* 29:*2* 54:*6*
companies 28:*23* 41:*13*
company 41:*5* 47:*18*
competing 41:*12*
completely 24:*7*
computer 8:*16* 10:*5, 25* 11:*8, 14, 20, 22* 12:*3, 12, 25* 13:*12, 23, 25* 14:*1, 2, 3, 10, 12* 15:*7* 21:*3, 13* 22:*14* 23:*8, 25* 25:*12, 15, 25* 26:*3* 27:*11, 14* 28:*4* 34:*18, 19, 25* 35:*20* 36:*13, 16, 19, 25* 37:*3* 39:*25* 40:*8* 45:*7, 10, 13, 14, 16* 47:*7, 8, 19, 20* 48:*20* 51:*13* 53:*21, 22, 25* 54:*3, 13, 17, 18* 58:*7* 68:*22* 74:*9, 16, 22*
Computer-Aided 78:*10*
computers 39:*9* 47:*5* 51:*7* 53:*23* 69:*4*
computing 20:*3, 7, 12, 17, 24* 21:*4, 14, 20* 22:*22* 23:*3* 26:*4*
conceding 10:*8*
concept 34:*5* 42:*20* 56:*22* 59:*16*
concepts 40:*21, 23* 53:*3*
concern 28:*4*
concerns 40:*3, 5*
conclusion 62:*5*
concrete 40:*1*
configurable 75:*16, 18*
configuration 30:*23* 31:*17* 56:*15*
configurations 30:*25* 31:*6*
configure 56:*20*
configuring 30:*18*
confirms 22:*13* 48:*7*
confusion 36:*1* 51:*12*
connect 46:*23* 47:*24* 55:*5* 56:*17*
connected 20:*14*
connecting 16:*16*
connection 4:*13* 38:*14* 56:*1, 5, 9, 21* 57:*20* 59:*12, 23* 60:*2, 4* 73:*12, 15, 18*
connections 16:*11*
consistent 64:*20* 65:*7*
construction 71:*5*
contain 78:*11*
context 68:*6* 70:*22*
continuous 73:*3* 74:*23* 75:*8, 11, 25*
continuously 55:*16* 56:*4* 58:*23*
contradicts 48:*10*
convert 54:*24*
copies 5:*3, 12*
copy 5:*1*
CORP 1:*3*
correct 4:*14* 5:*13* 8:*11* 9:*18* 10:*17, 22* 13:*10* 14:*7* 18:*5, 20* 19:*2* 25:*17* 26:*20* 27:*1, 20* 28:*25* 32:*7, 20* 33:*15* 34:*12* 35:*1, 9* 37:*4, 15* 39:*3* 42:*17* 43:*11* 45:*16, 18* 50:*3* 53:*7, 12, 21* 55:*23* 58:*3* 59:*21* 60:*8* 64:*19* 65:*5* 66:*7* 67:*5, 24* 71:*6* 73:*10, 21* 74:*10* 78:*12*
correctly 30:*17*
corrupt 34:*8*
corrupts 34:*24*
could 11:*20* 14:*2* 15:*17* 16:*4* 20:*8, 22* 21:*1, 3, 13* 24:*12, 18, 22, 23* 30:*13* 35:*11* 41:*17, 19, 20* 42:*9* 43:*3, 5, 12, 16* 46:*5, 20, 23* 47:*23* 49:*7* 52:*12* 53:*23, 25* 54:*6* 55:*15* 56:*3, 9, 17, 20* 63:*10, 11, 12, 23* 64:*5* 67:*6* 68:*3, 15, 17* 69:*7* 72:*6* 74:*1*
Counsel 12:*7* 78:*16*
countries 45:*25*
COUNTY 78:*3*
couple 66:*14*
course 49:*18*
COURT 77:*10*
cross 5:*18*
CRR 1:*15*
custom 43:*20* 46:*12, 14*

< D >
dashboard 31:*16*
data 11:*24* 14:*25* 18:*2, 4, 17* 20:*19, 22* 23:*7, 24* 24:*21* 26:*2, 23* 27:*12* 29:*25* 30:*16, 23* 31:*20* 34:*7, 9, 10, 25* 35:*9* 40:*20* 49:*17, 19, 25* 50:*4* 55:*16* 56:*7, 10, 11* 57:*1, 2, 3, 10, 14, 21, 22* 58:*2, 15, 16* 59:*3, 4, 9, 10, 18, 19* 60:*5, 7, 20, 21, 22* 61:*2, 7, 8, 9, 14, 16* 63:*7, 10, 11, 15, 16, 20, 21, 23, 24, 25* 64:*1, 6, 18, 24, 25* 65:*3, 9* 66:*5* 67:*3, 18* 68:*4, 10, 12, 13* 70:*2, 3, 4, 19* 71:*6, 9, 10, 12, 14, 17, 20, 25* 72:*7, 19, 22, 23, 24* 73:*7, 10, 16, 20, 23, 25* 74:*2, 4, 18, 22* 75:*3, 10, 22* 76:*4*
database 20:*22* 21:*6, 11* 29:*14* 31:*25* 33:*16, 20, 21* 34:*8*
databases 31:*22*
datapoint 17:*17*
dec 64:*8*
declaration 4:*13, 20* 5:*11* 6:*23, 25* 7:*4, 16, 17, 23* 8:*3, 7, 14, 17* 9:*2, 12, 17* 10:*20* 14:*16* 15:*4* 26:*6* 27:*18, 25* 36:*5* 37:*21* 48:*3, 14* 53:*5, 9* 55:*12* 56:*6* 57:*8* 61:*18* 62:*14*
declarations 4:*19, 23* 5:*2, 4* 6:*4, 5, 7, 16* 24:*9* 26:*7*

34:*23* 52:*23* 64:*14* 71:*2*
decs 5:*17*
default 53:*17*
definition 25:*7* 62:*3*, *15*
delay 69:*7*
deliver 16:*18*
delivered 57:*3* 60:*8*, *19*, *22* 61:*2*
demonstrate 10:*6* 39:*20*
demonstrates 9:*9*
deployment 31:*1*, *5*
depo 6:*15*
DEPOSITION 1:*6*, *13* 4:*11*, *12* 5:*23* 6:*13*, *19* 7:*20* 8:*6*, *9*, *20* 48:*8* 64:*16* 65:*10* 66:*2*, *6*, *19*, *25* 67:*9*
design 16:*21*
designed 27:*2*
designs 42:*4*
destination 69:*25*
detail 72:*10* 73:*1*
details 14:*19* 33:*25*
develop 34:*12* 39:*16* 42:*1*, *10* 46:*12* 52:*13*
developed 16:*24* 39:*12*, *21* 41:*7* 46:*14*
developing 30:*6* 38:*5* 45:*20*
device 10:*25* 20:*21* 21:*5*, *10*, *18* 25:*5* 53:*20* 54:*15* 57:*14* 61:*5* 63:*5*, *7*, *8*, *14*, *17*, *18* 64:*1*, *18*, *23*, *25* 65:*4*, *8* 66:*5* 67:*2*, *3*, *4*, *13*, *16*, *20*, *22*, *23* 68:*11*, *14* 69:*14*, *20* 70:*1*, *2*, *4* 72:*2*, *20*
DEVICES 1:*6* 15:*1* 20:*20* 23:*12* 25:*8*, *24* 26:*1* 29:*16*, *18* 37:*7* 38:*9* 40:*10* 46:*6* 51:*14*
did 5:*24* 6:*1*, *12* 7:*9*, *10* 8:*2*, *4* 9:*16* 28:*25* 53:*8* 56:*13* 57:*12* 62:*1*, *14* 70:*11*
didn't 13:*18* 41:*13* 42:*8*
Diego 2:*8*
different 15:*1* 20:*20*, *23* 24:*7* 27:*9* 29:*6*, *15* 31:*8* 47:*1*, *2* 49:*6*, *7* 51:*7* 57:*8* 64:*5*, *22* 67:*15*, *19* 68:*16* 71:*13*, *19* 72:*21* 73:*5* 74:*3* 76:*1*
digitize 68:*24*
DIRECT 3:*4* 4:*4* 7:*2* 19:*15*
directed 45:*23*
directly 26:*4*
disagree 40:*4*
disclosed 70:*25*
disclosure 36:*18*, *24* 45:*15* 66:*16* 68:*2*, *9*
discuss 50:*8*
discusses 22:*5*
discussion 22:*7* 76:*7*
displayed 30:*17*
disproves 17:*7*
distinguishing 72:*16*
distributed 20:*3*, *7*, *11*, *13*, *17*, *24* 21:*4*, *14*, *19* 22:*22* 23:*3* 26:*4*
Do 4:*9*, *15*, *23* 5:*4* 6:*14* 7:*6*, *8* 8:*21* 15:*24* 17:*21* 19:*5*, *7* 20:*3*, *5*, *6* 21:*2*, *9* 22:*6*, *24*, *25* 23:*1* 26:*10* 28:*6*, *7* 29:*23* 30:*9*, *10* 31:*7* 32:*16* 34:*2* 38:*17* 41:*25* 42:*5* 44:*1* 45:*11* 48:*15*, *16*, *21*, *22* 49:*2*, *3* 53:*12*, *16*, *18*, *19* 55:*2*, *24* 63:*9*, *11*, *23* 65:*12* 72:*15* 75:*24* 76:*8* 78:*6*
doctor 47:*16*
doctor's 46:*5*
document 17:*8*, *10* 51:*1*, *3*
documentation 48:*6*, *9*
documents 6:*9* 7:*6* 17:*6*
does 13:*21* 33:*19* 37:*5*, *9* 44:*8* 51:*13*, *18*
doesn't 9:*24* 10:*1*, *7* 14:*19* 37:*7* 58:*8* 74:*16*, *21*
doing 4:*8* 29:*21* 34:*1* 45:*23* 58:*19* 74:*16*
done 30:*4* 31:*13* 40:*3* 48:*3*
don't 6:*20* 8:*4* 9:*2* 10:*11* 13:*15* 17:*4* 18:*12* 23:*14* 25:*20* 26:*10* 32:*25* 33:*23* 34:*20* 35:*12*, *21* 39:*3* 47:*13* 50:*20* 54:*8* 63:*20*
DOUGLAS 78:*3*
down 12:*16* 39:*23* 58:*7*
Dr 4:*6* 6:*5*, *22* 7:*5*, *11*, *12* 8:*21* 17:*4*, *14*, *17* 19:*5* 26:*8* 44:*8* 45:*5* 48:*2*, *9* 61:*17* 62:*8* 64:*9* 65:*13* 70:*23* 76:*5* 77:*4*
drill 39:*23*
Drive 2:*7*
drivers 46:*23* 47:*2*
duly 4:*2*
during 8:*6*, *8*, *9*

< E >
earlier 7:*23* 14:*21* 37:*17* 54:*22* 56:*16* 57:*10* 64:*15*
easy 46:*11* 54:*11*
Edition 41:*8*
effective 58:*24*
effectively 58:*19*
either 18:*11* 70:*16*
elaborate 20:*8*
electrical 68:*19*, *21*, *22* 69:*2*, *11*, *13*, *14*
electronic 69:*4* 70:*16*
encompass 35:*11*
encryption 39:*11* 40:*14*, *16*
Enterprise 41:*8*
entries 31:*8* 53:*15*
enunciate 15:*17*
environment 20:*25* 21:*5*, *15*, *20* 22:*23* 23:*4*
environments 20:*3*, *7*, *17*
equally 27:*12*

especially 16:*25* 70:*22*
Ethernet 69:*23*
event 78:*17*
ever 16:*24*
Everyone 43:*1*, *2*
everything 5:*15* 6:*15* 41:*23* 66:*15*, *22*
exact 61:*6*
exactly 30:*1*
EXAMINATION 3:*4* 4:*4*
examined 4:*3*
example 5:*12* 7:*20*, *21* 9:*1* 11:*14*, *23* 12:*4* 14:*18*, *20* 15:*11* 16:*7*, *15* 17:*17*, *25* 18:*6*, *8*, *17* 20:*21*, *23* 21:*5* 25:*7* 28:*12*, *18* 29:*1*, *13* 31:*10* 32:*12* 36:*21* 37:*23* 39:*18* 40:*14*, *18* 41:*4* 46:*4*, *25* 47:*9*, *11*, *12* 51:*15* 52:*20* 53:*6* 56:*18* 58:*6*, *22* 59:*2* 61:*7*, *12* 67:*6*, *21*, *25* 68:*12*, *15* 69:*22* 70:*18* 72:*12* 74:*8*, *15* 75:*4*, *11*
examples 40:*1* 45:*11* 46:*3* 47:*6*, *10* 50:*23* 67:*10* 68:*16* 75:*6*
Executive 2:*7*
Exhibit 19:*6*, *14*, *16* 21:*25* 22:*1* 48:*14* 50:*8*, *9*, *14* 65:*16*
EXHIBITS 3:*10* 5:*12*, *24* 19:*9*
exist 38:*18*
existed 27:*16* 32:*6* 39:*11* 57:*23*
existing 33:*9* 34:*10*
exists 32:*7*
Expires 78:*24*
explain 70:*11*, *12*
explained 14:*21* 17:*13* 24:*25* 25:*18* 26:*21* 28:*18* 29:*19* 35:*10* 40:*12* 52:*15* 55:*3* 56:*22* 57:*18* 64:*21* 65:*6* 66:*14* 68:*8* 69:*5* 70:*9*, *23* 71:*11*, *16* 72:*9*, *17*, *25* 73:*3* 74:*13*, *17* 75:*7*, *14*, *16*
explaining 62:*8* 71:*10*, *21*
expressly 11:*17*
extent 50:*24* 66:*23* 72:*22*

< F >
fact 5:*17* 9:*8* 14:*25* 17:*6* 26:*12*, *13* 29:*8* 39:*7*, *18* 40:*15* 48:*7* 51:*18* 54:*19*
facts 10:*16*
familiar 31:*21*
figure 74:*1*
files 30:*23*
first 4:*2* 8:*6*, *9* 12:*21* 19:*15* 30:*6*, *7* 31:*24* 48:*17* 49:*11* 53:*24* 63:*15* 64:*16*, *23* 65:*10* 66:*2*, *6*, *19*, *25* 67:*3*, *9*
five-minute 44:*7* 76:*9*
flawed 8:*1* 9:*9*, *13* 10:*4*, *17* 17:*16*, *18*
flow 20:*16*
follow 51:*1* 52:*14*
following 45:*2* 51:*3* 77:*2*
follows 4:*3*
foregoing 78:*7*
form 29:*1*
formatted 18:*18*
forward 53:*14* 55:*10* 63:*7* 64:*6* 68:*13*, *17* 69:*10* 70:*3*, *10*
forwarded 69:*19*
forwarding 67:*23* 68:*7*, *21*
frame 37:*17* 45:*5* 57:*12*
free 43:*15*
FREEHILLS 2:*12*
front 5:*2*, *5* 7:*1* 9:*2* 58:*7* 74:*15*, *22*
full 36:*23* 78:*11*
FULLER 2:*12* 3:*4* 4:*5* 9:*14* 10:*18* 12:*9* 13:*4*, *9*, *22* 15:*19*, *21* 16:*1* 19:*11* 20:*15* 21:*12*, *22* 24:*20* 31:*19* 32:*18* 33:*18* 35:*5*, *6* 38:*21* 44:*5* 45:*4* 52:*3* 60:*13* 62:*13* 65:*15*, *20*, *25* 71:*3* 74:*25* 76:*5*, *8* 77:*4*, *14*
functionalities 43:*25*
functionality 20:*13* 24:*23* 43:*6* 44:*1*
fundamental 36:*4* 37:*22* 51:*11*
fundamentally 7:*13* 8:*1*, *15*, *23* 9:*5*, *9*, *16* 10:*4*, *17* 17:*15*, *16*
further 77:*6*, *8*

< G >
garage 43:*21*
General 1:*15* 32:*25* 33:*12* 37:*2*, *6* 78:*5*, *22*
generate 69:*2*
generated 69:*17*, *19*, *24*
generating 70:*16*
generically 30:*18*, *21*
getting 15:*16* 34:*25* 35:*4* 62:*19* 63:*3*, *10*, *22* 68:*7*, *19* 70:*13* 74:*4* 76:*3*
give 21:*23* 32:*11* 40:*1* 41:*4* 64:*17*
giving 6:*11* 33:*24*
go 21:*23*, *24* 27:*24* 32:*1* 44:*6* 48:*13* 50:*8* 55:*11* 61:*21* 69:*23*, *24* 76:*6*
goes 26:*17*
going 7:*2* 15:*10* 18:*5* 31:*20* 44:*5* 53:*4* 57:*25* 61:*20*, *21* 69:*7*, *8*
Good 4:*6*, *8* 5:*9* 19:*12*
Goodrich 7:*5*, *11* 8:*21* 26:*8* 48:*2* 62:*8*
Goodrich's 6:*5* 7:*12* 17:*4*, *14*, *17* 48:*9* 64:*9* 70:*23*
Google.com 54:*10*
governments 29:*8* 38:*2*

great 4:*21* 5:*22* 44:*9*
guess 50:*7* 66:*20*

< H >
had 7:*19* 9:*6* 38:*6* 41:*16, 25* 42:*13* 43:*1* 44:*1, 2* 45:*2* 53:*19* 58:*1* 76:*7* 77:*2*
handled 28:*18* 41:*22* 42:*12*
happen 17:*24* 33:*19* 35:*24* 58:*11* 71:*13, 20*
happening 36:*3*
happens 31:*11* 60:*25* 71:*21* 75:*19*
hard 5:*1, 3, 12* 34:*13*
has 5:*7, 14, 17* 16:*24* 20:*2* 27:*16* 37:*3* 40:*20* 42:*3* 48:*2* 49:*16* 55:*17* 71:*25* 72:*8, 19* 73:*12* 75:*23* 76:*4*
HASTINGS 2:*6*
have 4:*11, 23* 5:*1, 4, 6, 11* 6:*24, 25* 7:*25* 9:*2* 10:*11, 24* 11:*7, 13, 20* 14:*22* 17:*5, 19* 19:*5, 7, 12, 14* 20:*6* 21:*2* 23:*1, 10, 23* 25:*8, 10, 20, 21* 27:*9* 28:*15, 16* 29:*23* 33:*2, 3, 15* 37:*5, 12, 17* 38:*14, 15* 40:*7, 10, 22* 41:*1, 14* 42:*1, 2, 5, 8, 24* 43:*13* 45:*11* 47:*2, 13* 49:*15, 16, 18, 19, 23* 51:*15, 16, 17, 19, 22, 23, 24* 53:*18* 54:*2, 8, 16* 57:*9, 13* 58:*8, 22* 60:*1, 4* 61:*1, 8, 20, 21* 65:*12* 67:*15* 69:*5* 70:*19* 72:*11* 73:*14, 24* 74:*17, 21* 77:*5, 8*
haven't 5:*16*
he 7:*19* 9:*3, 8, 24, 25* 10:*8, 16* 17:*7, 8* 26:*9* 48:*10*
heading 19:*18, 19* 22:*2*
healthcare 38:*3* 45:*22* 46:*1, 17* 47:*11*
heavily 40:*24*
her 51:*24*
HERBER 2:*12*
hereunto 78:*18*
he's 8:*14, 17* 17:*10*
Hewlett 41:*7*
high 34:*5*
him 48:*10*
himself 10:*8* 48:*10*
his 7:*17, 21* 8:*14, 17, 22* 9:*2, 10, 12, 15, 20* 10:*4, 7, 13, 15* 17:*6, 7* 48:*2* 60:*10* 66:*9* 70:*7* 74:*12*
hit 53:*16*
home 55:*2, 4, 6, 9*
host 16:*17, 24* 17:*2, 22, 25* 18:*2, 4, 19, 22, 23* 26:*19, 21* 27:*5, 7, 19, 21* 31:*15* 53:*7, 8, 12, 16*
hosted 27:*13*
hosting 25:*9*
hour 44:*6*
hours 6:*21*
housekeeping 4:*22*
HTML 18:*17, 18* 52:*9, 11*
HTTP 18:*16* 19:*1* 29:*2* 52:*9, 11* 53:*12* 58:*20, 25* 75:*12*
HTTP/1 56:*12, 22*
HTTP/1.0 57:*6, 24*
HTTP/2 72:*18, 19*
HTTP://localhost 18:*10*
HTTPS 28:*19* 40:*14, 16* 53:*24*
HTTPS://localhost 53:*13*
huge 38:*9*
hundred 10:*9* 47:*13*

< I >
IBM 41:*16, 21* 43:*4, 11, 12*
identified 7:*25*
identify 73:*24*
if 5:*9, 14, 19* 9:*3* 15:*17* 16:*8, 16* 17:*8* 18:*6, 11* 20:*8* 21:*18* 23:*5* 27:*5, 9* 28:*2, 15, 17, 25* 29:*2* 30:*7* 33:*5* 34:*12, 24* 37:*9* 38:*20* 39:*10, 21* 40:*13, 17* 49:*11* 54:*13* 55:*16* 59:*23* 61:*7* 62:*9* 64:*11* 65:*16* 66:*1, 10, 16* 68:*1, 4* 69:*14, 15, 20* 70:*23* 75:*18*
immediately 57:*22* 58:*14, 18* 59:*10* 60:*20, 23* 61:*2, 3, 15* 71:*18* 73:*7* 74:*5, 19*
immersion 33:*22*
impact 9:*25* 10:*7*
implement 28:*9* 33:*5* 35:*16* 36:*6, 7* 38:*25* 61:*5*
implementation 14:*19* 33:*3* 39:*24* 40:*2*
implementations 33:*8*
implemented 10:*24* 11:*7, 11* 12:*13* 13:*1, 13, 20* 14:*5* 28:*3* 32:*4* 39:*25* 41:*1* 51:*4* 72:*7*
implementing 11:*13* 12:*2, 5, 11* 28:*8* 43:*9, 10* 56:*7*
important 19:*17*
improve 32:*4*
improved 12:*5, 16, 17* 13:*6, 11* 14:*14*
improvement 14:*13*
improving 11:*25* 31:*20, 25* 32:*2*
include 64:*5*
including 29:*7* 39:*15* 40:*25* 47:*23* 55:*9*
incoming 54:*24*
incorporated 43:*23*
incorporates 41:*9*
incorrect 8:*1* 10:*17* 17:*15* 70:*24*
INDEX 3:*3*
India 47:*13*
industry 27:*3, 14, 15* 28:*19* 29:*5* 30:*4* 33:*7* 34:*14* 35:*14* 36:*6, 9* 37:*16* 39:*13* 41:*1* 42:*8* 43:*18, 21* 46:*15* 52:*9*

information 19:*17* 46:*24* 47:*1* 61:*9*
infrastructure 46:*11*
injected 33:*14*
injection 32:*7*, *24* 33:*15*, *20* 34:*3*, *24* 35:*8*
injections 31:*21* 32:*19*
inside 69:*1*
inspected 71:*12*
interested 78:*17*
interface 15:*10*, *25* 23:*6*, *23* 24:*10* 49:*15*, *24* 50:*1*
intermediary 63:*7*, *10*, *13*, *14*, *17*
Internet 11:*6* 54:*5*
intranet 12:*24*
introduce 37:*7*
introduced 37:*11*
invention 56:*13*
IP 26:*22* 27:*10*, *22* 53:*24* 54:*2*, *7*, *8*, *12*, *16*, *24* 55:*1*, *8*
IPR 1:*3*, *4* 4:*14*, *15*, *16*
IPRs 5:*8*
issue 75:*9*, *10*
issues 8:*11* 37:*8*, *11*, *19* 41:*14* 42:*11*, *12*, *13* 44:*4*
items 6:*8*
its 13:*11* 41:*17*

< J >
January 1:*17* 78:*19*
Java 41:*7*
JEFFREY 2:*13*
jeffrey.price@hsfkramer.com 2:*15*
JENNA 2:*12* 15:*16* 65:*12*
jenna.fuller@hsfkramer.com 2:*15*

< K >
KAMILAH 2:*6*
kamilahalexander@paulhastings.com 2:*9*
keep 56:*9*, *21* 70:*10*
kin 78:*14*
kind 9:*8* 17:*23* 22:*20* 32:*5*, *8* 35:*22* 57:*10* 61:*15* 68:*18*
kinds 47:*21*
knew 75:*17*
know 4:*25* 5:*9*, *16* 6:*20* 16:*10* 19:*6*, *20* 25:*6* 34:*7* 36:*5* 38:*25* 39:*5* 54:*16*, *20* 61:*20* 68:*25* 76:*1*
knowledge 24:*1* 52:*25*
knowledgeable 62:*11*
known 11:*13*, *15* 14:*23* 26:*19* 27:*19* 35:*15* 36:*7* 37:*18* 39:*7*, *14* 40:*15*, *21*, *22* 42:*21* 45:*6*, *9* 46:*8* 53:*6* 56:*8*, *23* 57:*5*, *9* 58:*10*, *22* 59:*16* 67:*17* 72:*11*
knows 17:*1*
KRAMER 2:*12*
Kumar 11:*5*, *16* 12:*24* 14:*21*, *22* 24:*2* 53:*1*
Kumar's 12:*1* 14:*6*

< L >
lag 69:*7*
language 28:*7*
laptop 51:*16*, *18*, *22*, *24* 52:*2*, *16*, *17*, *19* 55:*5*
large 29:*17*
largest 38:*6*
last 13:*19*
leaving 56:*5*
legal 62:*5*
level 34:*5*
license 38:*7* 41:*15*, *18* 42:*9* 43:*3*, *12*, *15*
life 71:*22*
limited 14:*2* 23:*6* 63:*21* 66:*13* 68:*7* 72:*6*
limiting 64:*4* 70:*24*
line 32:*1* 36:*12* 65:*20* 66:*11*
lines 17:*9*
listening 18:*14*
literally 55:*4*
little 4:*22* 15:*17*, *18*
LLP 2:*6*
local 15:*10*, *12* 16:*17*, *23* 17:*2*, *22*, *25* 18:*2*, *3*, *19*, *22*, *23* 26:*19*, *21* 27:*4*, *7*, *19*, *21* 31:*15* 53:*7*, *8*, *12*, *16*
located 15:*8* 29:*15*, *18*
location 20:*23* 29:*15* 47:*3*
locations 15:*1* 20:*20*
logged 55:*21*
logic 24:*18* 49:*17*, *25* 50:*2* 52:*14*
log-in 42:*3*
logistics 46:*17*, *20* 47:*17*
long 40:*21* 41:*5* 42:*22* 46:*7* 56:*8*, *21*, *23* 58:*9*, *25* 59:*6*, *7*, *22* 60:*14*, *16*, *18* 61:*5*, *14* 71:*17*, *18* 72:*12* 73:*1*, *9*, *19* 74:*18*, *20*, *23* 75:*1*, *5*, *12*, *21*
look 14:*20* 17:*9* 49:*11* 59:*23* 68:*1* 69:*1*
looked 5:*16*
looking 16:*8* 22:*8*, *10* 62:*24* 71:*7*
looks 5:*7*
lot 26:*12* 36:*21* 46:*14* 67:*14* 68:*16* 71:*19*
lying 47:*14*

< M >
machine 11:*22* 15:*9*, *11* 16:*4*, *6*, *22* 17:*6*, *11*, *20*, *24* 18:*1*, *3*, *20*, *24* 19:*1*, *4* 23:*13*, *21* 25:*9* 26:*11*, *23* 27:*6*, *8*, *10*, *23* 28:*10*, *16* 41:*19*, *20* 48:*5*, *12* 49:*8*, *14* 53:*11* 54:*4* 56:*19* 57:*15*
machines 25:*11* 47:*14* 49:*6*, *7*
magic 60:*1*
magically 74:*1*
making 29:*8* 31:*1*, *5* 36:*4*

43:*20* 57:*19*
malicious 32:*8*
mall 36:*22*
manage 56:*19*
management 26:*12* 29:*22* 30:*10, 11, 18* 31:*10, 12, 16, 17* 46:*18* 47:*17, 18*
managing 26:*13* 30:*21* 31:*7* 56:*17*
manipulation 23:*7, 24* 24:*21* 49:*17, 19, 25* 50:*4*
MARKED 3:*10*
materials 5:*4, 8, 24* 6:*2*
mean 6:*14* 10:*1* 11:*10* 13:*21* 15:*24* 21:*9* 22:*12* 30:*11, 17* 32:*21* 52:*6* 55:*24* 57:*16* 58:*6* 63:*9, 13, 17* 66:*10* 72:*15* 73:*22*
meaning 55:*18, 20* 72:*23*
means 20:*7* 63:*22*
meant 21:*19* 36:*16* 74:*14*
mechanism 75:*24*
MEDICAL 1:*6* 47:*11*
memorized 66:*15*
mentioned 16:*7* 38:*1* 40:*23* 45:*19* 46:*7* 48:*8* 52:*11* 57:*7* 58:*9*
mentioning 47:*16* 58:*5*
message 16:*18* 18:*14* 26:*23* 27:*23*
Mexico 47:*13*
middle 36:*21, 22*
middleware 42:*21* 43:*6, 17* 46:*13*
millions 28:*23* 38:*8* 39:*19*
mind 6:*8*
mischaracterizes 9:*20* 60:*10* 66:*9* 70:*7* 74:*12*
misunderstanding 51:*11* 73:*13*
mitigate 39:*1, 5* 40:*3*
mitigated 28:*13*
MOBILE 1:*6, 13* 21:*4* 29:*9* 30:*6, 7* 45:*23*
modern 73:*24*
modified 14:*4*
morning 4:*6, 8* 55:*4* 67:*14*
multiple 15:*1* 22:*2* 25:*8, 24*
multitude 25:*11* 39:*18*

< N >
name 40:*19* 42:*4*
NAT 7:*21* 8:*2, 10* 9:*4, 7, 16, 23, 24* 10:*1, 9, 12* 54:*22* 55:*9*
NATs 55:*2*
naturally 14:*20*
Nebraska 1:*16* 78:*2, 6*
need 5:*19* 7:*24* 10:*3* 46:*1*
network 8:*10, 22* 16:*25* 20:*14* 37:*4, 5, 7* 38:*14* 54:*14, 15, 16, 21* 55:*6, 7* 69:*23, 24*
network-based 39:*16*
New 2:*14* 28:*21* 55:*16* 56:*11, 25* 57:*2, 3, 4, 21, 22* 58:*2, 15, 16* 59:*4, 9, 10* 60:*5, 20, 22, 24* 61:*2, 9, 16* 69:*2, 17, 18* 70:*16* 75:*20*
news 61:*11, 13*
NO 1:*3, 5* 3:*10* 7:*12, 24* 10:*3* 18:*22* 38:*23* 54:*19* 62:*23* 77:*5, 8*
nonweb 52:*21*
Notary 1:*15* 78:*5, 22*
November 78:*24*
number 29:*17*
numbers 4:*15, 16* 38:*9*
numerous 14:*17*
NY 2:*14*

< O >
Objection 8:*24* 9:*19* 12:*18* 13:*7, 16* 15:*14* 20:*9* 21:*7, 16* 24:*15* 31:*2* 32:*9* 33:*17* 35:*2* 38:*19* 51:*8* 60:*9* 62:*4* 65:*11* 66:*8* 70:*6* 72:*3* 74:*11*
obvious 11:*6* 12:*24* 14:*18* 62:*10* 64:*13* 71:*1, 5*
obviously 6:*6*
occur 16:*3, 6*
off 11:*4* 44:*6* 76:*6, 7*
OFFICE 1:*1* 46:*5, 21, 22* 47:*8, 17*
oftentimes 30:*14* 31:*12* 48:*6*
Oh 15:*19* 19:*10*
okay 4:*18* 5:*9, 18* 6:*25* 10:*21* 15:*5* 19:*10, 14, 25* 21:*24* 22:*4, 17* 28:*1* 44:*10* 50:*9, 16* 53:*19* 55:*13* 61:*19, 25* 65:*18* 71:*23*
ongoing 31:*6, 10*
Online 1:*15*
onward 63:*16*
open 6:*22* 19:*14* 29:*23* 30:*14* 36:*20* 55:*22, 25* 56:*5, 10, 21* 57:*20* 65:*15* 68:*25* 73:*12*
opening 8:*3* 31:*13*
operating 16:*17* 25:*16* 47:*6* 54:*14, 18*
opined 8:*22* 23:*17*
opining 10:*23*
opinion 10:*7* 11:*7* 62:*1* 71:*4*
opinions 7:*5, 9, 13* 8:*22* 9:*11, 12, 15* 10:*4, 7* 17:*15, 18*
opposed 7:*17*
optical 70:*17*
optional 68:*9*
order 11:*23* 60:*3* 73:*16*
orders 77:*10*
ordinary 10:*10* 11:*12* 14:*22* 62:*3*
original 6:*7, 16* 14:*16* 26:*6*
outside 5:*10* 8:*25* 14:*21* 15:*15* 20:*10* 31:*3* 54:*15* 65:*12, 23*
overly 64:*10, 11*

70:*24*
overriding 34:*9*
OWNER 1:*7* 2:*11* 7:*4* 47:*18*

< P >
p.m 77:*1, 15*
Packard 41:*7*
packet 68:*25* 69:*1* 71:*12, 14*
packets 69:*22*
PAGE 3:*2, 3, 4* 18:*17, 18* 19:*15, 20* 21:*24* 22:*1* 42:*3* 43:*10* 44:*2* 48:*15* 50:*13* 65:*18, 20* 66:*11*
pages 11:*18* 28:*21, 23* 37:*24* 46:*12* 78:*11*
paper 50:*21*
Paragraph 7:*3* 10:*19, 20* 11:*4* 12:*4, 5, 21, 23* 15:*3* 16:*14* 22:*10* 26:*18, 24* 27:*24* 28:*2* 33:*4* 48:*13* 53:*4* 55:*11, 13, 14* 61:*18, 22* 71:*7, 23* 72:*5*
pardon 12:*4* 64:*17, 24* 66:*4*
parenthetical 48:*17*
parking 36:*21*
part 13:*19* 18:*8* 24:*10, 14, 18, 22, 23* 31:*24* 39:*3*
particularly 32:*3*
parties 41:*25* 43:*17* 45:*1* 77:*1* 78:*15*
parts 56:*6*
party 41:*3*
password 40:*18* 42:*4*
PATENT 1:*1, 2, 3, 5, 7* 2:*11* 7:*4*
patents 4:*17* 5:*7* 6:*6* 30:*2*
path 53:*15*
PAUL 2:*6*
pay 43:*13, 16*
payload 71:*14*
PC 46:*6, 25* 47:*3, 15*
PCs 46:*4, 20*
pending 12:*8*
people 17:*4* 26:*10* 31:*7* 38:*4* 40:*19* 41:*11* 42:*7* 46:*4, 5, 14*
percent 10:*9* 66:*25*
perform 29:*12*
performing 38:*1* 70:*15*
performs 24:*19, 24*
period 11:*15* 26:*15* 38:*6* 39:*20* 45:*20* 56:*9*
person 43:*9*
personal 13:*24* 14:*2, 3* 47:*7*
perspective 7:*14*
PETITIONER 1:*4* 2:*5*
Ph.D 1:*7, 14* 2:*3* 4:*1* 78:*8*
phone 51:*17, 20, 23, 25* 52:*17, 18*
physical 69:*23*
physician 11:*23* 32:*14* 36:*16* 45:*13, 14* 54:*13* 55:*19, 21* 58:*7* 74:*15, 21*
physicians 37:*1*
physician's 45:*7* 55:*15*
physician-side 10:*24* 11:*8, 13, 20, 22* 12:*3, 12, 25* 13:*12, 23* 14:*10, 12* 25:*12, 15, 25* 26:*2* 27:*11, 14* 34:*18* 35:*20* 36:*13, 15, 19, 25* 37:*3* 39:*24* 40:*8* 45:*16* 51:*13*
picking 10:*5*
piece 67:*19*
plain 7:*22* 62:*2*
play 19:*17* 26:*5*
plumbing 41:*24* 42:*19* 43:*6*
point 9:*24* 33:*4* 34:*11* 36:*4* 37:*9, 14* 47:*25* 65:*13*
pointed 9:*10* 48:*6*
pointing 23:*15* 66:*21*
points 36:*11* 48:*10*
polling 56:*23* 58:*9, 23* 59:*7, 22* 60:*17, 18* 61:*5, 14* 71:*17, 19* 72:*12* 73:*1, 3, 9, 19* 74:*18, 20, 23, 24* 75:*1, 5, 8, 11, 13, 21, 25*
poor 33:*2*
port 16:*11, 18* 18:*11, 13, 15* 53:*17*
portals 61:*12*
portion 17:*8*
portions 37:*20*
POSITA 10:*23* 11:*7* 12:*5, 10* 14:*20* 16:*10* 24:*2* 28:*9* 36:*5, 17, 23* 37:*17* 38:*25* 39:*5* 40:*22, 25* 52:*25* 57:*9* 58:*22* 62:*12* 63:*22* 68:*5* 72:*11* 75:*17*
possible 17:*19*
Potentially 30:*20* 34:*9* 67:*20* 70:*15*
practices 36:*8* 37:*18* 39:*6, 9, 10, 13* 40:*24* 41:*2*
precisely 30:*3* 55:*3* 64:*7*
preclude 51:*13, 18*
precluded 23:*12*
premise 40:*4*
preparation 5:*25* 6:*9, 12, 16, 18*
preparing 5:*23* 6:*14*
prepping 19:*13*
present 45:*1* 77:*1*
pretty 5:*15*
prevalently 10:*1*
previously 7:*19*
PRICE 2:*13*
principal 28:*4*
printed 5:*1*
private 54:*14, 15, 20* 55:*6*
probably 6:*20*
problems 48:*4*
proceedings 45:*2* 77:*2*
process 22:*13, 14* 48:*19, 20* 49:*13* 68:*22* 70:*2, 3*
processed 70:*4*
processees 23:*8, 24* 24:*21*
processes 68:*12, 13*
processing 68:*3, 9* 70:*15* 71:*6, 9*
Professional 78:*5, 22*
proper 28:*3, 8, 11* 32:*4* 33:*6* 39:*1*

40:*2, 17*
properly 35:*16*
protocol 56:*12* 72:*18, 19, 21* 75:*24*
protocols 52:*9, 10* 72:*13, 15* 73:*5, 11, 14, 24* 74:*3* 76:*1*
provide 7:*5, 9* 9:*17* 14:*19* 34:*4* 42:*18* 46:*24* 56:*10*
provided 7:*11* 39:*17* 41:*24* 42:*19* 43:*17* 60:*6* 71:*18* 72:*24*
provides 57:*22*
providing 34:*9* 47:*6* 49:*23, 24* 73:*7*
PST 1:*17*
Public 1:*15* 36:*20* 37:*2, 6* 54:*16* 78:*5, 22*
publicized 39:*14*
published 39:*8* 40:*24*
pull 58:*20, 25* 59:*6* 60:*14, 16* 72:*6*
purposes 30:*12* 52:*22*
push 57:*13, 14, 16* 58:*19, 24* 59:*23* 60:*3, 15* 71:*24* 72:*7, 13, 15, 18, 19* 73:*11, 13, 23, 24*
pushed 72:*1, 20* 73:*17*
pushes 63:*16, 20*
pushing 63:*21* 64:*2* 72:*22, 23*
push-type 56:*7*
put 18:*9, 10, 11, 12, 13* 34:*7* 53:*14*
putting 25:*24*

< Q >
query 55:*16* 56:*4*
querying 74:*7, 10*
question 11:*3* 12:*8, 15* 13:*3, 19* 15:*24* 21:*9, 18, 21* 28:*15* 31:*24* 32:*17* 33:*11, 23* 45:*17* 50:*25* 66:*13, 17, 23* 75:*15*
questioning 32:*2* 36:*12*
questions 8:*5, 19* 9:*23* 77:*6, 9*
quick 76:*8*
quickly 19:*25*
quite 6:*8* 72:*10, 25*
quote 49:*11* 73:*16*

< R >
rarely 22:*21*
reachable 9:*7*
read 8:*15* 12:*22* 19:*21* 66:*11*
readily 14:*18, 23*
reading 17:*3* 36:*23* 65:*1*
ready 31:*1, 5*
real 71:*22*
reality 70:*12*
really 10:*3, 7* 23:*14* 30:*17* 31:*4* 32:*16* 36:*2* 48:*1* 63:*13, 19*
reason 28:*22* 41:*11* 61:*6*
reasoning 9:*17*
reasons 21:*13* 61:*4*
recall 4:*15*
receive 58:*2* 59:*19* 67:*18*
received 55:*17* 71:*10, 25* 72:*8* 74:*5, 19*
receives 59:*13* 63:*15* 64:*23* 67:*3* 70:*2*
receiving 59:*18* 62:*25* 64:*17* 65:*3* 66:*4* 67:*13* 73:*8*
Recess 44:*11* 76:*10*
record 44:*6, 7* 76:*6, 7* 77:*11*
reduced 78:*9*
refer 18:*24*
reference 5:*10* 11:*17* 12:*2, 12* 30:*10* 36:*19*
references 62:*11*
referencing 4:*18*
refers 20:*12* 52:*8*
regardless 37:*18*
regards 11:*16* 16:*13* 52:*24* 58:*9* 62:*24* 65:*24* 67:*8* 71:*16*
Registered 78:*4, 22*
related 4:*16*
relation 8:*6*
relevance 23:*15* 24:*16* 34:*20* 35:*12*
relevant 10:*14* 26:*14*
remains 57:*20*
remember 54:*11*
remembering 66:*21*
remote 11:*6* 12:*24* 45:*24*
remotely 1:*14* 2:*2, 3, 9* 29:*18*
renders 11:*5* 12:*24*
rent 41:*20*
rephrase 65:*2*
reply 4:*13, 20* 5:*10, 17* 6:*4, 23* 7:*16* 9:*11, 17* 15:*4* 26:*7* 27:*17, 25* 37:*20* 53:*5, 9* 56:*6* 57:*7* 64:*8*
REPORTER 3:*4* 77:*10* 78:*5, 22*
request 18:*2, 16* 19:*1* 55:*10* 56:*24* 57:*2, 4, 19* 58:*12, 18* 59:*3, 8, 13* 60:*7, 21, 24* 61:*5* 64:*1* 73:*10, 19, 20, 23* 74:*20* 75:*2, 9, 11, 13, 20* 76:*3*
Requests 77:*10*
residing 15:*7*
RESMED 1:*3*
resolved 54:*7, 10, 12*
resources 20:*13*
respect 71:*9*
respond 7:*24* 10:*3* 18:*15* 57:*1* 58:*13, 14* 59:*9*
responding 18:*21* 19:*1*
response 7:*4* 58:*16* 71:*18*
restrictive 64:*10, 12*
result 73:*6* 74:*4* 75:*22* 76:*3*
retransmit 62:*3, 15* 63:*2, 21* 64:*3, 5, 16* 67:*20* 70:*21*
retransmits 63:*6* 67:*4*
retransmitted 71:*15*

retransmitting 62:*18*, *21* 63:*1*, *4* 65:*2* 67:*2*, *22* 70:*1*
review 5:*24*
reviewed 6:*3*, *4*, *5*, *6*, *9* 7:*3*
RF 69:*15* 70:*17*
right 5:*21* 19:*18* 39:*22* 65:*7* 66:*24*
robust 43:*22*
roles 19:*17*
Rooney 1:*15* 78:*4*, *21*
rough 77:*13*
route 54:*25*
router 54:*23* 55:*9*
routers 55:*2*
RPR 1:*15* 78:*21*
rules 23:*7*, *23* 24:*13*, *17*
run 25:*24* 26:*10*, *13* 41:*19*, *20* 42:*16* 45:*6*, *9*, *12* 46:*20* 47:*23* 48:*11* 49:*12* 51:*21* 52:*16* 55:*4*
running 11:*21* 14:*10*, *11* 17:*5*, *24* 26:*1* 27:*11*, *13* 29:*20*, *24* 31:*7* 33:*21*, *22* 34:*17* 35:*1*, *19* 38:*10* 40:*8* 46:*4*, *25* 47:*3*, *15*, *20*, *21* 52:*1*, *16* 53:*17*, *21* 56:*18*, *19* 67:*16*, *19*

< S >

safe 29:*11* 39:*17*, *22*
San 2:*8*
SANDEEP 1:*7*, *13* 2:*3* 4:*1* 78:*7*
saying 8:*12*, *13*, *14*, *17* 10:*13* 13:*3*, *20* 14:*9* 23:*5* 26:*9* 30:*14* 38:*23* 49:*12*, *21* 55:*19* 58:*5* 59:*1* 62:*20*, *23* 63:*3* 66:*12* 69:*21*
says 17:*10* 19:*16* 22:*20* 34:*7* 48:*11*, *23* 50:*10*
scanned 19:*25*
scenario 52:*20*
science 8:*16* 10:*5*
scope 8:*25* 15:*15* 20:*10* 31:*3* 65:*12*
scratch 42:*9* 43:*3*
se 54:*8*
seal 78:*19*
second 19:*22* 21:*24* 39:*3* 61:*21* 64:*17*
section 10:*22* 11:*5* 19:*23* 22:*8*, *11* 62:*7* 64:*8*
secure 28:*24* 29:*1*, *11* 30:*9* 36:*13*, *14* 38:*1* 39:*17*, *22*
secured 34:*19* 35:*20*
security 28:*5*, *11*, *18* 31:*21*, *25* 32:*2*, *5* 37:*8*, *11*, *22* 40:*3*, *5* 41:*14*, *22* 42:*10*, *18*, *25* 43:*23* 44:*4*
see 7:*6* 19:*10* 20:*3* 22:*7*, *24* 28:*6* 30:*15*, *16* 34:*20* 35:*12*, *21* 48:*15*, *21* 49:*2* 55:*16* 66:*1*
send 18:*1*, *4*, *13* 58:*15* 63:*12* 69:*2* 70:*20* 74:*20* 75:*2*, *3*
sending 63:*24*, *25* 64:*6*, *18* 65:*4*
sends 26:*22* 27:*22* 59:*11* 75:*20*
sense 21:*21*
sent 61:*15* 71:*9* 72:*19*
sentence 19:*21*, *22* 20:*2* 22:*12*, *15*, *20* 66:*13*, *20*
serve 38:*8*
server 9:*3* 11:*1*, *8*, *11*, *14*, *20*, *24* 12:*1*, *2*, *13* 13:*1*, *13*, *20* 14:*6*, *9*, *12* 15:*7*, *8*, *12* 16:*3*, *7*, *9*, *19*, *22* 17:*20* 18:*2*, *4*, *6*, *7*, *14*, *15*, *19*, *22*, *23*, *25* 19:*18* 20:*22* 21:*6*, *11* 22:*3*, *6*, *13*, *16*, *19* 23:*2*, *10*, *11*, *13*, *19*, *20* 24:*3*, *5*, *14*, *19*, *22*, *24* 25:*2*, *10*, *12*, *17*, *22* 26:*1*, *11*, *25* 27:*6*, *8*, *12*, *13* 28:*3*, *10*, *12*, *16* 29:*4*, *14*, *20*, *22*, *23*, *24* 30:*16*, *19*, *22*, *23* 31:*1*, *7*, *11*, *13* 32:*3*, *19*, *22* 33:*2*, *5*, *9*, *14* 34:*17* 35:*1*, *19* 38:*10*, *11*, *15* 40:*1*, *9*, *11*, *20* 41:*8*, *9*, *15*, *16*, *18* 42:*17* 43:*10* 45:*6*, *9*, *12* 46:*9*, *21*, *23*, *25* 47:*4*, *8*, *15*, *21* 48:*5*, *11*, *18*, *19*, *24* 49:*4*, *5*, *9*, *13*, *14*, *16*, *22*, *24* 50:*3*, *5*, *6*, *10*, *11*, *14*, *17*, *22* 51:*5*, *6*, *21* 52:*1*, *13*, *15*, *18* 53:*11*, *16*, *20* 54:*1*, *4*, *18*, *25* 55:*5*, *7*, *10*, *16* 56:*2*, *4*, *10*, *17*, *19*, *20*, *25* 57:*20*, *21*, *22* 58:*13*, *14* 59:*8*, *14*, *21* 60:*2* 72:*20* 73:*8*, *15*, *25* 74:*6*, *7* 75:*2*, *3*, *23* 76:*4*
servers 9:*6* 10:*11* 14:*23* 16:*10* 17:*5*, *11*, *24* 18:*24* 19:*16*, *19* 22:*2* 26:*13*, *19* 28:*20* 31:*12* 32:*23*, *25* 33:*12* 38:*10* 42:*21* 46:*4* 50:*19*, *22*, *23*, *25* 57:*12* 72:*13*
server's 33:*20*
services 46:*2*
session 55:*18* 59:*20*
set 75:*18*
seventy-eight 78:*11*
shorthand 16:*23* 17:*22* 18:*24* 26:*22* 27:*18*, *21* 53:*6* 78:*8*
show 8:*17* 26:*8* 74:*23*
shows 10:*15* 75:*3*
side 19:*12*
signals 68:*20*, *21*, *22*, *24* 69:*2*, *5*, *6*, *8*, *12*, *13*, *15*, *16*, *17*, *18* 70:*14*, *16*, *17*
signature 78:*19*
similar 71:*11*
simplest 48:*24* 49:*22* 50:*6*, *11*
simply 17:*12* 18:*23* 30:*14* 31:*13* 41:*15* 42:*9* 43:*3* 49:*12*, *21*

52:*8* 53:*23* 54:*11*, *23*, *25* 55:*8* 56:*15*, *24* 58:*12* 62:*17*, *18* 63:*2*, *22* 67:*23* 68:*7* 71:*8*, *21* 74:*3* 75:*20*, *23*
single 16:*22* 47:*15*
single-tier 22:*6*, *15*, *18* 23:*2*, *19*
sit 70:*19*
site 11:*6* 12:*24*
sites 61:*12*
sitting 58:*7* 74:*15*, *21*
situation 25:*20*
skill 10:*10* 11:*12* 14:*22*
slash 53:*14*
SMITH 2:*12*
software 14:*9*, *11*, *13* 16:*8*, *9*, *16*, *25* 17:*1*, *20* 18:*14*, *15* 26:*12* 30:*4*, *22* 33:*23*, *25* 36:*10* 41:*3*, *6*, *11*, *18*, *19*, *21* 42:*13*, *16*, *17* 43:*5*, *11*, *13*, *14* 46:*21* 47:*4*, *22* 48:*12* 58:*12* 67:*15*, *19*
solution 40:*6*
somebody 43:*20* 47:*18* 74:*2*
soon 38:*13*, *14* 56:*11* 57:*3*, *21* 59:*12*, *18* 61:*3*, *16* 72:*23* 75:*23* 76:*4*
Sorry 13:*17*
sounds 5:*22* 44:*9*
space 38:*3* 46:*16*, *17*
speak 11:*25* 12:*3*, *10* 14:*13* 26:*18* 27:*17*, *18*
speaking 73:*18*, *19*
specific 16:*9*, *21* 26:*22* 33:*2* 35:*18*, *24* 39:*24*
specifically 16:*8* 19:*16*
specify 42:*2*
spend 6:*12*
spending 67:*14*
spoke 6:*23* 42:*15* 57:*25* 67:*11* 74:*8*
SQ 34:*3*
SQL 31:*21* 32:*6*, *7*, *19*, *23* 33:*15*, *19* 34:*24* 35:*7*
Ss 78:*2*
stand 65:*4* 66:*1*, *24* 67:*1*
standard 27:*4*, *15* 28:*20* 29:*3*, *5* 33:*7* 34:*14* 35:*14* 36:*6*, *9*, *10* 37:*16* 39:*13* 40:*13* 41:*1* 43:*19*, *21* 46:*15* 47:*25* 52:*9* 77:*13*
start 10:*20*
started 4:*23*
starting 11:*4* 66:*11*
State 1:*16* 7:*3* 27:*21* 55:*14* 78:*2*, *6*
stated 9:*21* 10:*8*
statements 9:*8*, *11* 17:*4*, *7*, *18*
STATES 1:*1* 12:*23* 22:*5*
stating 11:*12*, *19* 17:*12* 62:*8* 71:*8* 72:*5*
stored 71:*9*, *15*
strike 35:*5*
strokes 35:*8*
structure 48:*24*, *25* 49:*4*, *10*, *22* 50:*7*, *11*, *12*
submit 7:*15*
submitted 4:*13*
subtitle 12:*22*
Suite 2:*7*
support 43:*15*
sure 5:*14* 11:*2*, *10* 12:*14* 13:*3* 14:*1* 15:*23* 21:*8*, *20* 28:*14* 31:*23* 32:*13*, *21* 33:*11* 39:*2* 52:*6* 57:*16* 58:*4* 63:*13*, *19* 73:*22*
surprised 17:*3*
surprising 48:*1*
sworn 4:*2*
system 12:*1*, *6*, *11*, *17* 13:*6*, *11*, *24* 14:*5*, *14* 16:*17* 24:*1* 30:*8* 32:*15* 34:*1*, *22* 39:*16* 51:*6* 72:*6*
systems 19:*18* 30:*7* 34:*12* 39:*12* 45:*20*, *22*

< T >
take 21:*3*, *14* 22:*21* 23:*3* 33:*9* 44:*7* 68:*23* 70:*23* 77:*12*
taken 1:*14* 4:*11*, *12* 44:*11* 76:*10* 78:*8*
talk 14:*16* 53:*8* 56:*6* 58:*21* 62:*12* 64:*13*
talked 20:*18* 26:*5*, *6* 43:*25* 46:*10* 47:*10* 54:*21*, *22* 56:*16* 59:*25* 64:*6*, *7* 67:*12* 68:*1* 71:*2*
talking 16:*14*, *20* 23:*9*, *16*, *22*, *25* 24:*2*, *6*, *8*, *25* 25:*1*, *2*, *3*, *13*, *14*, *19* 30:*1*, *3* 31:*18* 32:*12*, *14*, *15* 34:*16*, *17*, *21*, *23* 35:*13*, *14*, *18*, *22*, *25* 40:*6*, *7* 41:*23* 45:*15* 47:*12* 50:*21* 51:*1* 52:*10*, *24* 53:*1* 56:*3*, *5* 60:*11* 62:*7* 63:*20* 67:*7*, *14* 72:*23*
talks 11:*17* 22:*18* 28:*8* 39:*8* 40:*16*
targets 33:*16*, *20*
TCP/IP 54:*6*
teaching 12:*1* 14:*6*
team 19:*12*
technical 7:*14* 9:*12* 10:*15*, *16* 17:*14*
technologies 11:*18* 14:*24* 28:*20* 29:*3*, *4*, *5*, *10*, *11* 30:*9* 33:*7*, *10* 34:*15* 35:*15* 36:*6*, *8*, *9* 37:*10*, *13*, *16* 38:*12* 39:*21* 40:*14* 41:*2* 42:*24* 43:*8*, *18*, *19*, *22* 46:*16* 52:*8*, *21*, *22* 53:*2*, *3* 59:*24*
technology 7:*21* 8:*2* 14:*24* 27:*4*, *16* 28:*25* 29:*9* 33:*5*, *6* 38:*5*, *7*, *11* 39:*11* 41:*9*, *17* 42:*6* 43:*4*, *20* 46:*9* 47:*25* 57:*5*, *18* 58:*10*, *19* 60:*1* 68:*18* 70:*13* 71:*22*
telemedicine 46:*1*
telling 36:*2*
term 20:*2*, *4* 34:*6* 35:*10* 62:*15* 64:*3*,

10 69:*10* 70:*10*, *21*, *25* 74:*14*
terminate 56:*13* 75:*13*
terminated 69:*13*, *17*
terminating 70:*14*
terms 65:*13* 72:*22*
testified 4:*3* 35:*7*, *12* 48:*2* 64:*15* 65:*3*, *9* 67:*8*
testifying 70:*5*
TESTIMONY 3:*3* 8:*8*, *9* 9:*20* 60:*10*, *16* 65:*5* 66:*2*, *6*, *9*, *24* 70:*7* 74:*12* 78:*7*, *12*
testing 30:*12*, *13*
textbook 8:*16* 10:*6*
Thank 4:*9* 15:*20*, *21* 77:*5*, *7*, *14*
thereof 78:*17*
thick 5:*15*
thing 60:*25* 73:*2*, *4*
things 7:*24* 29:*21* 40:*17* 43:*1* 70:*12* 71:*13*, *20*
think 6:*21* 7:*18* 12:*21* 14:*15* 32:*25* 36:*17*, *23* 37:*19* 39:*4* 40:*4*, *12* 47:*9* 50:*20*, *24* 51:*10* 52:*11* 56:*6* 59:*25* 62:*16* 63:*21* 64:*3*, *5* 66:*18* 67:*12*, *25* 68:*1* 70:*8*, *11* 72:*9* 75:*15*
third 41:*3*, *25* 43:*17*
thousand 66:*25*
Tiers 22:*2*
time 6:*12* 10:*2*, *10* 11:*15* 26:*15* 29:*9* 30:*4*, *5* 31:*11* 37:*13*, *17*, *25* 38:*5*, *18* 39:*12*, *20* 45:*5*, *20* 53:*23* 56:*9*, *12*, *14* 57:*12* 60:*8*, *19* 67:*14* 70:*20* 77:*5*
time-out 75:*18*, *19*
time-outs 75:*12*
times 66:*15*
title 13:*2*
today 4:*7*, *10*, *18* 5:*24* 6:*10*, *15*, *17* 22:*21* 49:*1* 50:*7*, *13* 65:*5* 66:*3* 72:*10* 77:*5*
today's 6:*13*, *18*
Toge 11:*5* 12:*6*, *11*, *16*, *17*, *23* 14:*18* 23:*17* 24:*1* 36:*18*, *24* 45:*15* 51:*5* 52:*24*, *25*
Toge/Kumar 72:*5*
Toge's 10:*24* 11:*8*, *25* 12:*3*, *25* 13:*6*, *11*, *23* 14:*10*, *11*, *14* 25:*15*
told 19:*7*
top 9:*23* 34:*15* 35:*17* 42:*1*
touch 37:*19*
to-wit 45:*3* 77:*3*
TRADEMARK 1:*1*
transaction 29:*13*
transactions 29:*12* 38:*2*
Transcription 78:*10*, *12*
translation 8:*11*, *23*
transmit 63:*2* 64:*22*
transmits 64:*25* 65:*9* 66:*5*
transmitting 62:*17*, *19*, *21* 63:*4* 64:*23*, *24* 65:*8* 66:*3*, *4*
transportation 46:*19*
TRIAL 1:*2*
truck 46:*22* 47:*2*
trucking 46:*22*
true 78:*11*
try 68:*12*
trying 13:*5* 33:*4* 34:*2*, *11* 37:*14* 38:*22* 39:*6* 59:*5* 66:*1*
turn 10:*19* 15:*3* 61:*17*
turnaround 77:*13*
turns 60:*24*
two 5:*17* 6:*4* 36:*11*
type 16:*9* 18:*8* 27:*4*, *6*, *10* 33:*1*, *13* 45:*9* 46:*1*, *22* 53:*23* 54:*2*, *8*, *9*, *23*
types 29:*7* 61:*13* 76:*1*
typewriting 78:*9*
typical 23:*11* 48:*18*
typically 16:*11* 20:*12*
typing 31:*14* 55:*8*

< U >
U.S 1:*3*, *5*
umbrella 34:*6* 35:*10*
unclear 68:*18*
uncommon 28:*22*
underlying 44:*3* 46:*13*
understand 4:*10* 10:*11* 12:*14* 13:*3*, *5*, *15*, *18* 15:*24* 16:*2* 21:*9* 23:*14* 28:*14* 32:*13* 33:*11*, *23* 36:*18*, *24* 37:*23* 38:*22* 45:*17* 59:*15* 68:*6* 69:*20*
understanding 10:*23* 13:*10* 19:*3* 20:*6* 21:*2*, *18* 23:*1*, *18* 32:*6*, *17* 59:*5* 62:*2*, *10* 64:*9*, *12* 66:*12* 70:*24*
understood 11:*3* 29:*5* 31:*23* 42:*22* 46:*8* 62:*17* 72:*11*
UNITED 1:*1*
unmodified 13:*24*
unprocessed 70:*4*
unquote 73:*16*
unreachable 9:*4*
update 46:*24*
updates 61:*15*
upload 5:*21* 65:*17*
uploaded 4:*25*
uploading 31:*8*
URL 18:*9* 25:*25* 54:*7*, *9*, *10*, *11*, *24* 55:*8*
use 11:*17* 17:*1* 28:*19*, *20* 37:*10* 39:*10*, *11*, *17*, *22* 40:*13*, *17* 41:*11* 43:*5*, *11* 48:*25* 50:*7*, *12* 52:*7*, *22* 53:*6*, *13*, *24* 54:*7* 74:*14* 78:*9*
user 23:*6*, *23* 24:*10* 40:*19* 42:*4* 47:*7* 49:*15*, *23* 50:*1*

users 38:*8*
uses 54:*6*

< V >
vacuum 66:*14*, *21* 67:*7*
Vague 12:*19* 13:*8* 21:*7*, *16* 32:*10* 33:*17* 35:*2* 38:*19* 51:*9* 72:*3*
VIDEOCONFERENCE 1:*6*, *13*
view 12:*23* 13:*2* 71:*1*
villages 45:*24* 47:*12*
virtual 15:*10*, *25*
vulnerabilities 28:*11*, *12*, *17* 32:*20* 33:*3* 37:*22* 38:*16*, *17*, *23*, *24* 39:*4*
vulnerability 32:*22*
vulnerable 32:*23* 33:*1*, *12*

< W >
wait 56:*25* 59:*4*
waiting 64:*1*
waits 59:*9*
walk 16:*4*
want 16:*2* 30:*19* 32:*1*, *5* 36:*11* 42:*2*, *4* 61:*10* 69:*9* 70:*9*, *11* 73:*25* 74:*1*
wants 74:*2*
water 5:*6*
way 28:*4*, *8*, *11*, *25* 32:*4* 33:*6* 34:*13* 39:*1*, *22* 40:*15* 54:*11* 58:*19*, *24* 59:*24* 60:*4* 61:*1* 75:*25* 78:*14*
ways 37:*15* 40:*2* 57:*8* 58:*21* 64:*5* 71:*19*
web 9:*5*, *6* 10:*11*, *25* 11:*14*, *18* 12:*1*, *13* 13:*1*, *13*, *20* 14:*5*, *9*, *12*, *23* 16:*9*, *10*, *15*, *19* 18:*7*, *18*, *23*, *25* 23:*11* 24:*3*, *5* 25:*1*, *3*, *6*, *7*, *9*, *10*, *16*, *23* 26:*11*, *13* 27:*7*, *11*, *13* 28:*12*, *15*, *16*, *20*, *21*, *22* 29:*3*, *10*, *22*, *23*, *24* 30:*9*, *16*, *19*, *22*, *23* 31:*1*, *7*, *11*, *12*, *13* 32:*19*, *22*, *23*, *25* 33:*2*, *5*, *6*, *9*, *10*, *12*, *14*, *20* 34:*17* 35:*1*, *19* 37:*10*, *12*, *24* 38:*9*, *10* 39:*21*, *25* 40:*8*, *11*, *20* 41:*9* 42:*23* 43:*9*, *10*, *18* 44:*2* 45:*6*, *9*, *12* 46:*4*, *8*, *12*, *16*, *21*, *23*, *25* 47:*4*, *15*, *21*, *23* 50:*19*, *22*, *25* 51:*5*, *21*, *22*, *23*, *24*, *25* 52:*1*, *7*, *8*, *21* 53:*11*, *16*, *20*, *25* 54:*18*, *25* 55:*5*, *7*, *10*, *15* 56:*2*, *17*, *19*, *20* 57:*12* 58:*1* 59:*20* 72:*13* 74:*7*, *9*
web-based 14:*24* 17:*1* 43:*7* 50:*18* 52:*4* 53:*1*
WebLogic 41:*17*
website 42:*1*, *10* 44:*2* 55:*22* 61:*8*, *13*
websites 28:*23* 37:*24* 39:*19* 46:*12*
well 4:*8* 5:*18* 7:*18* 9:*1*, *21* 11:*14*, *21* 14:*8* 15:*2* 16:*7* 20:*18* 21:*23* 23:*5*, *13*, *22* 25:*11*, *16* 26:*14*, *16* 29:*4*, *22* 33:*14* 34:*5*, *19*, *24* 35:*15*, *21* 36:*7*, *15* 38:*11* 39:*2*, *14* 40:*9*, *15*, *21* 41:*12* 42:*2* 45:*6*, *9*, *22* 46:*2*, *8* 47:*24* 49:*11*, *17*, *20* 52:*14* 53:*15* 54:*1*, *5* 56:*3*, *8*, *23* 57:*5* 58:*4*, *23* 61:*6* 64:*15* 66:*12* 67:*12*, *17*, *21* 72:*4* 75:*7*
well-built 28:*24*
well-known 27:*15* 53:*3*
we're 16:*8* 23:*9*, *16*, *25* 24:*2*, *8*, *25* 25:*1*, *13*, *14* 30:*3* 32:*15* 34:*16*, *17*, *22* 35:*13*, *14*, *18*, *24* 47:*12* 62:*24*
we've 20:*18* 43:*25* 44:*5* 59:*25* 67:*7*, *13* 68:*1*
whatsoever 38:*24*
WHEREOF 78:*18*
whitelist 31:*9*
whitelists 30:*20*
wife 51:*24*
wife's 52:*17*
Wi-Fi 69:*15*
within 1:*15* 21:*19* 55:*6* 62:*23* 71:*14* 78:*6*
without 33:*24* 44:*3*
WITNESS 2:*2* 9:*1*, *21* 12:*20* 13:*18* 15:*23* 19:*10* 20:*11* 21:*8*, *17* 24:*17* 31:*4* 32:*11* 35:*3* 38:*20* 44:*9* 51:*10* 60:*11* 62:*6* 65:*18*, *22* 66:*10* 70:*8* 72:*4* 74:*13* 77:*7* 78:*13*, *18*
words 15:*16*
work 44:*8* 59:*24* 69:*4* 70:*12* 73:*14* 74:*16*
worked 41:*5* 42:*13*
working 29:*14* 45:*21* 46:*19*
works 54:*6* 59:*25* 60:*3* 68:*19* 74:*18*
world 29:*6* 30:*7*, *8* 37:*25* 38:*7* 45:*24*, *25*
worry 41:*14* 43:*5* 44:*3*
would 4:*24* 6:*21* 8:*16* 10:*6*, *10*, *24* 11:*7*, *12* 14:*5*, *9*, *20*, *22*, *25* 16:*10*, *17* 18:*1*, *4*, *8*, *10*, *11*, *17* 21:*6*, *10* 28:*9*, *12* 30:*14* 32:*16* 35:*19*, *20*, *23* 36:*5*, *13*, *14*, *17*, *19*, *24* 37:*2*, *12*, *17*, *21* 38:*25* 39:*5*, *15* 40:*2*, *22*, *25* 41:*14* 42:*12* 43:*10*, *13* 47:*1*, *14*, *15*, *18* 49:*14* 51:*4* 52:*4*, *5*, *7* 53:*12*, *13* 54:*2*, *16*, *19*, *23*, *25* 55:*10* 56:*10*, *23*, *24*, *25* 57:*1*, *4*, *9* 58:*11*, *12*, *13*, *14*, *15*, *17*, *22*, *25* 59:*17*, *22* 61:*5*, *10*,

13 62:*10*, *16*, *25* 64:*4* 67:*21* 68:*5*, *13* 69:*1*, *23*, *24* 72:*1*, *11* 74:*22* 75:*1*, *3*, *13*
writing 62:*1*, *14*
wrong 7:*13*, *22* 8:*15*, *18*, *23* 9:*5*, *13*, *16* 10:*4*, *6* 17:*18* 26:*9* 38:*13* 62:*9*, *10* 64:*10*, *12*

< Y >
Yeah 19:*14* 52:*11* 60:*18* 66:*10*
Yes 4:*21* 6:*18* 15:*20* 18:*6* 19:*4*, *20*, *24* 21:*1* 25:*18* 28:*7*, *8* 30:*1* 37:*5* 42:*19* 43:*12*, *16* 45:*19* 46:*3* 52:*7* 53:*2*, *9*, *17* 59:*2*, *7* 60:*23* 61:*24* 64:*21* 65:*7* 68:*16* 69:*22* 75:*4*
York 2:*14*