Paper No. 1

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

APPLE INC.
Petitioner,

v.

VIRNETX, INC. AND SCIENCE APPLICATION INTERNATIONAL CORPORATION,
Patent Owner

Patent No. 7,490,151
Issued: Feb. 10, 2009
Filed: Sep. 30, 2002
Inventor: Edmund C. Munger, *et al.*

Title: ESTABLISHMENT OF A SECURE COMMUNICATIONS LINK BASED DOMAIN NAME (DNS) REQUEST

Inter Partes Review No. IPR2015-00187

PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT NO. 7,490,151 UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.1-.80 & 42.100-.123

TABLE OF CONTENTS

I. MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1) 1

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1) 1

B. Related Matters Under 37 C.F.R. § 42.8(b)(2) 1

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3) 2

II. PAYMENT OF FEES – 37 C.F.R. § 42.103 2

III. REQUIREMENTS FOR IPR UNDER 37 C.F.R. § 42.104 2

A. Grounds for Standing Under 37 C.F.R. § 42.104(a) 2

B. Challenge Under 37 C.F.R. § 42.104(b) and Relief Requested 3

C. Claim Construction under 37 C.F.R. §§ 42.104(b)(3) 5

1. Domain Name 5

2. DNS Request 6

3. Secure Server 6

4. Automatically Initiating/Creating an Encrypted/Secure Channel 6

5. Client 7

6. Between [A] and [B] 7

IV. SUMMARY OF THE ‘151 PATENT 8

B. Summary of the Prosecution History of the ‘151 Patent 9

C. The Effective Priority Date of the Claims of the ‘151 Patent 9

V. MANNER OF APPLYING CITED PRIOR ART TO EVERY CLAIM FOR WHICH AN IPR IS REQUESTED, THUS ESTABLISHING A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE ‘151 PATENT IS UNPATENTABLE 11

A. [GROUND 1] – Kiuchi Anticipates Claims 1, 2, 6-8, and 12-14 11

1. Kiuchi Anticipates Claim 1 18

2. Kiuchi Anticipates Claim 7 24

3. Kiuchi Anticipates Claim 13 24

4. Kiuchi Anticipates Claims 2, 8, and 14 26

5. Kiuchi Anticipates Claims 6 and 12 27

B. [GROUND 2] – Kiuchi In View of RFC 2660 Renders Obvious Claims 1, 2, 6-8, and 12-14 28

VI. CONCLUSION 32

Attachment A. Proof of Service of the Petition

Attachment B. List of Evidence and Exhibits Relied Upon in Petition

Apple Inc. ("Petitioner" or "Apple") petitions for *Inter Partes* Review ("IPR") under 35 U.S.C. §§ 311–319 and 37 C.F.R. § 42 of claims 1, 2, 6-8, and 12-14 ("the Challenged Claims") of U.S. Patent No. 7,490,151("the '151 patent"). By its accompanying Motion for Joinder, Petitioner seeks to join this petition to IPR2014-00610, a proceeding instituted on the same patent and the same prior art. The grounds identified in this petition are identical to the grounds in which the Board instituted trial on in IPR2014-00610.

I. MANDATORY NOTICES UNDER 37 C.F.R § 42.8(a)(1)

A. Real Party-In-Interest Under 37 C.F.R. § 42.8(b)(1)

The real party of interest of this petition pursuant to § 42.8(b)(1) is Apple Inc. ("Apple") located at One Infinite Loop, Cupertino, CA 95014.

B. Related Matters Under 37 C.F.R. § 42.8(b)(2)

The '151 patent is the subject of a number of civil actions including: (i) Civ. Act. No. 6:13-cv-00211-LED (E.D. Tex.), filed February 26, 2013; (ii) Civ. Act. No. 6:12-cv-00855-LED (E.D. Tex.), filed November 6, 2012; (iii) Civ. Act. No. 6:10-cv-00417-LED (E.D. Tex.), filed August 11, 2010; (iv) Civ. Act. No. 6:11-cv-00018-LED (E.D. Tex), (v) Civ. Act. No. 6:13-cv-00351-LED (E.D. Tex), filed April 22, 2013 ("the 2013 VirnetX litigation"); (vi) Civ. Act. No. 6:10-cv-00094 (E.D. Tex); and (vii) Civ. Act. No. 6:07-cv-00080 (E.D. Tex).

The '151 patent is also the subject of merged *inter partes* reexamination nos. 95/001,697 and 95/001,714. In the merged proceedings, the Office issued a Non-

Final Action on April 20, 2012 rejecting all 16 claims of the ’151 patent, including rejections based on several prior art references relied upon in this Petition.

The ’151 patent is the subject of an inter pates review filed by Microsoft Corporation (IPR2014-00610), instituted on October 15, 2014. The ’151 patent was also the subject of petitions for *inter partes* review filed by: New Bay Capital, LLC (IPR2013-00376, dismissed); Apple Inc. (IPR2013-00354, not instituted); and RPX Corporation (IPR2014-00173, not instituted).

C. Lead And Back-Up Counsel Under 37 C.F.R. § 42.8(b)(3)

Lead Counsel	Backup Lead Counsel
Jeffrey P. Kushan (Reg. No. 43,401) **jkushan@sidley.com** (202) 736-8914	Joseph A. Micallef (Reg. No. 39,772) **jmicallef@sidley.com** (202) 736-8492

D. Service Information

Service on Petitioner may be made by e-mail, or by mail or hand delivery to: Sidley Austin LLP, 1501 K Street, N.W., Washington, D.C. 20005. The fax number for lead and backup counsel is (202) 736-8711.

II. PAYMENT OF FEES – 37 C.F.R. § 42.103

The Director is authorized to charge the fee specified by 37 CFR § 42.15(a) to Deposit Account No. 50-1597.

III. REQUIREMENTS FOR IPR UNDER 37 C.F.R. § 42.104

A. Grounds for Standing Under 37 C.F.R. § 42.104(a)

Petitioner certifies that the ’151 patent is available for *inter partes* review by Petitioner. The Petitioner is not barred or estopped from requesting an *inter* *partes* review challenging the patent claims on the grounds identified in the petition. The ’151 patent was asserted against Petitioner in proceedings alleging infringement more than one year ago, but because this petition is accompanied by a motion for joinder to IPR2014-00610, the one-year period in 35 U.S.C. § 315(b) does not apply to this petition pursuant to 35 U.S.C. § 315(c). *E.g., Dell Inc. v. Network-1 Security Solutions, Inc.*, IPR2013-00385, Paper 17 at 4-5; *Microsoft Corp. v. Proxyconn, Inc.*, IPR2013-00109, Paper 15 at 4-5. Trial was instituted in IPR2014-00610 on October 15, 2014, less than one month prior to the filing of the present petition, as required by § 122(b). For the reasons detailed in the accompanying motion for joinder, this petition should be joined to the IPR2014-00610 proceeding.

B. Challenge Under 37 C.F.R. § 42.104(b) and Relief Requested

Petitioner requests an IPR of the Challenged Claims on the grounds set forth in the table shown below, and requests that each of the Challenged Claims be found unpatentable. An explanation of how these claims are unpatentable under the statutory grounds identified below is provided in the form of a detailed description that indicates where each element can be found in the cited prior art, and the relevance of that prior art. Additional explanation and support for each ground of rejection is set forth in Exhibit 1003, the Declaration of Dr. Roch Guerin (“Guerin Declaration”), referenced throughout this Petition.

Ground	**'151 Patent Claims**	**Basis for Rejection**
Ground 1	**1, 2, 6-8, and 12-14**	Anticipated under § 102 by Kiuchi
Ground 2	**1, 2, 6-8, and 12-14**	Obvious under § 103 based on Kiuchi in view of RFC 2660

The '151 patent issued from a string of applications allegedly dating back to an original application filed on October 30, 1998. However, as outlined in Section IV(C), the effective filing date for the embodiments recited by claims 1, 2, 6-8, and 12-14 of the '151 patent is no earlier than February 15, 2000.

Kiuchi qualifies as prior art under 35 U.S.C. § 102(b). Specifically, Kiuchi (Ex. 1018) is a printed publication that was presented at the 1996 Symposium on Network and Distributed Systems Security (SNDSS) on February 22 & 23, 1996, and published by IEEE in the Proceedings of SNDSS 1996. Ex. 1018.

Aventail qualifies as prior art under §§ 102(a) and (b). Specifically, Aventail (Ex. 1007) is a printed publication that was publicly distributed no later than January 31, 1999. Ex. 1005 at ¶¶ 11-36; Ex. 1006 at ¶¶ 11-24.

RFC 1034 qualifies as prior art under 35 U.S.C. § 102(b). Specifically, RFC 1034 (Ex. 1008) was published in November 1987 by the Internet Engineering Task Force (IETF). RFC 1034 was publically distributed no later than November 1987. Ex. 1008.

RFC 2660 qualifies as prior art under 35 U.S.C. § 102(b). Specifically, draft 01 of RFC 2660 (Ex. 1010) was published in February 1996 by the Internet

Engineering Task Force (IETF). RFC 2660 was publically distributed no later than February 1996. Ex. 1010.

C. Claim Construction under 37 C.F.R. §§ 42.104(b)(3)

A claim subject to IPR is given its "broadest reasonable construction in light of the specification of the patent in which it appears."[1] 37 C.F.R. § 42.100(b). For purposes of this proceeding only, Petitioner submits constructions for the following terms. All remaining terms should be given their plain meaning.

1. Domain Name

The Patent Owner has asserted to the PTAB that that a "domain name" means "a name corresponding to network address." Ex. 1019, p. 28. In view of the Patent Owner's assertions, it is reasonable, for purposes of this proceeding in which the broadest reasonable construction standard applies, to consider the term "domain name" as encompassing "a name corresponding to a network address."

[1] Because the standards of claim interpretation applied in litigation differ from PTO proceedings, any interpretation of claim terms in this IPR is not binding upon Petitioner in any litigation related to the subject patent. *See In re Zletz*, 13 USPQ2d 1320, 1322 (Fed. Cir. 1989). Additionally, Petitioner does not acquiesce to Patent Owner's or the district court's (or anyone else's) constructions, and otherwise reserves all of its rights to argue, contest, and/or appeal the constructions.

2. DNS Request

The Patent Owner has asserted to the PTAB that that a “DNS request” means “a request for a resource corresponding to a domain name.” Ex. 1019 at pp. 28-29. In view of the Patent Owner’s assertions, it is reasonable, for purposes of this proceeding in which the broadest reasonable construction standard applies, to consider the term “DNS request” as encompassing “a request for a resource corresponding to a domain name.”

3. Secure Server

The Patent Owner has asserted to the PTAB that a “secure server” means “a server that requires authorization for access and that can communicate in an encrypted channel.” Ex. 1019 at 38-39. In view of the Patent Owner’s assertions, it is reasonable, for purposes of this proceeding in which the broadest reasonable construction standard applies, to consider the term “secure server” as encompassing “a server that requires authorization for access and that can communicate in an encrypted channel.”

4. Automatically Initiating/Creating an Encrypted/Secure Channel

The Patent Owner has asserted to the PTAB “automatically initiating/creating an encrypted/secure channel” means “initiating/creating the encrypted/secure channel without involvement of a user.” Ex. 1019 at 41-42. In view of the Patent Owner’s assertions, it is reasonable, for purposes of this proceeding in which the broadest reasonable construction standard applies, to consider "automatically initiating/creating an encrypted/secure channel" as encompassing "initiating/creating the encrypted/secure channel without involvement of a user."

5. Client

Petitioner proposes that a "client," under the broadest reasonable interpretation of that term, encompasses "a computer or program from which a data request to a server is generated." This is not inconsistent with the '151 patent's specification and the understanding one of ordinary skill in the art would ascribe to this term when identifying the broadest reasonable construction. *See* Ex. 1003 at ¶ 16.

In particular, the '151 patent describes that "user's computer 2501 includes a client application 2504 (for example, a web browser) and an IP protocol stack 2505." Ex. 1001 at 37:1-3. Notably this sentence uses the term "client" with regard to an application, not the "user's computer." Thus, under this term's broadest reasonable interpretation, the '151 patent supports that "client" may refer to an application, not just a physical machine.

6. Between [A] and [B]

In prior litigation involving the '151 patent, the Patent Owner argued against a defendant's construction that "between" should mean "extend from one endpoint to the other," and instead stated that "between" should only apply to the "public communication paths." Ex. 1015, p. 10. Under the Patent Owner's contentions, an encrypted/secure channel is "between" a client and a secure server where the channel is on the public communication paths between the client and the secure server, regardless of whether the encrypted/secure channel extends completely from the client to the secure server. In view of the Patent Owner's assertions, it is reasonable, for purposes of this proceeding in which the broadest reasonable construction standard applies, to consider an encrypted/secure channel "between [A] and [B]" to encompass an encrypted/secure channel on the public communication paths between the client and the secure server, regardless of whether that channel fully extends from the client to the secure server.

IV. SUMMARY OF THE '151 PATENT

A. Brief Description

Generally, the '151 patent purportedly provides a secure mechanism for communicating over the internet. Ex. 1001 at Col. 3, line 8. In particular, the '151 patent purportedly describes a domain name service system configured to perform the operations of: (1) intercepting a DNS request sent by a client; (2) determining whether the intercepted DNS request corresponds to a secure server, (3) when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer, and (4) when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server. Ex. 1001 at Col. 37, lines 25-38.

The ‘151 patent includes 16 claims, of which claims 1, 7 and 13 are independent.

B. Summary of the Prosecution History of the ‘151 Patent

U.S. 7,490,151 issued on February 10, 2009 from U.S. Patent Application No. 10/259,494 (“the ‘494 application”), which was filed on September 30, 2002 with 20 claims as a division of U.S. Patent Application No. 09/504,783 (“the ’783 application”). *See* Ex. 1002, pp. 8, 79-82.

No reasons for allowance are expressly stated in the ‘151 patent file history. Moreover, the Patent Owner never explicitly distinguished the ultimately allowed and issued claims from the cited prior art, instead focusing its responsive arguments on claims that were later cancelled. *See, e.g.,* Ex. 1002, pp. 359-363, 385-388, 398-399, 559-561. Ultimately the patent issued on February 10, 2009, though it is not clear whether a particular feature led to the allowance.

C. The Effective Priority Date of the Claims of the ‘151 Patent

The ’151 patent issued from U.S. Application No. 10/259,494, filed September 30, 2002. The ’494 application is a division of U.S. Application No. 09/504,783, filed on February 15, 2000, now U.S. Patent No. 6,502,135, which is a continuation-in-part of U.S. Application No. 09/429,653, filed on October 29, 1999, now U.S. Patent No. 7,010,604. The ’494, ’783 and ’653 applications each attempt to claim priority under 35 U.S.C. 119(e) to Provisional Application Nos. 60/106,261, filed October 30, 1998 and 60/137,704, filed June 7, 1998.

Claims 1, 7 and 13 of the ’151 patent are independent claims. Claims 2-6 depend from claim 1, claims 8-12 depend from claim 7, and claims 14-16 depend from claim 13. Accordingly, claims 2-6, 8-12, and 14-16 cannot enjoy an effective filing date earlier than that of claims 1, 7 and 13, respectively, from which they depend.

Claims 1, 7 and 13 of the ’151 patent rely on information first presented in the ’783 application. For example, claim 1 of the ’151 patent specifies “determining whether the intercepted **DNS request** corresponds to a secure server” and subsequent steps involving **the DNS request.** Similarly, claims 7 and 13 include limitations involving **DNS requests** (*e.g.*, “intercepting a **DNS request** sent by a client . . .” and “determining whether the intercepted **DNS request** corresponds to a secure server . . .”, respectively). The first application that recites the term “DNS” is the ’783 application. Because none of the ’653, ’261 or ’704 applications disclose or even suggest use in any manner of DNS requests or proxy servers, these earlier filed applications do not describe or enable the subject matter defined by at least claims 1, 7 and 13 of the ’151 patent. Accordingly, the effective filing date of claims 1-16 of the ’151 patent is no earlier than **February 15, 2000.**

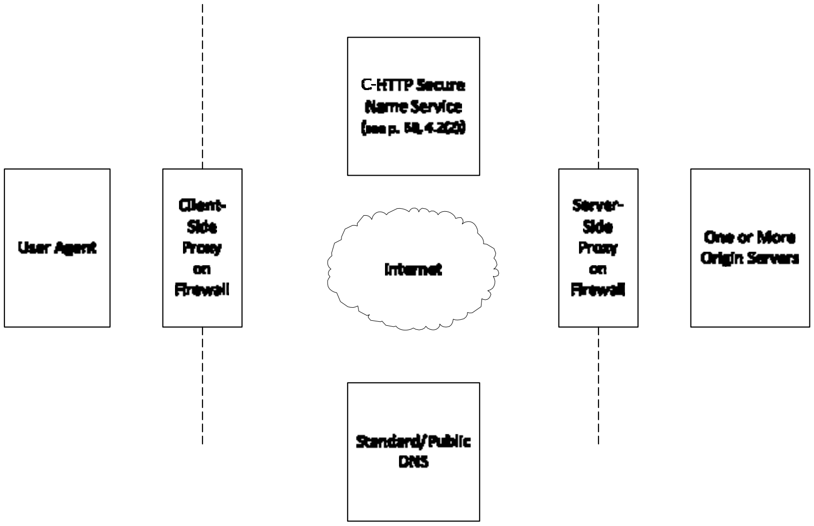
V. MANNER OF APPLYING CITED PRIOR ART TO EVERY CLAIM FOR WHICH AN IPR IS REQUESTED, THUS ESTABLISHING A REASONABLE LIKELIHOOD THAT AT LEAST ONE CLAIM OF THE ‘151 PATENT IS UNPATENTABLE

A. [GROUND 1] – Kiuchi Anticipates Claims 1, 2, 6-8, and 12-14

Kiuchi is a printed publication that was presented at the 1996 Symposium on Network and Distributed Systems Security (SNDSS) on February 22 & 23, 1996, and published by IEEE in the Proceedings of SNDSS 1996. Ex. 1018. Kiuchi is therefore prior art to the ‘151 patent at least under § 102(b), regardless of which effective filing date in the priority chain is applied to the claims.

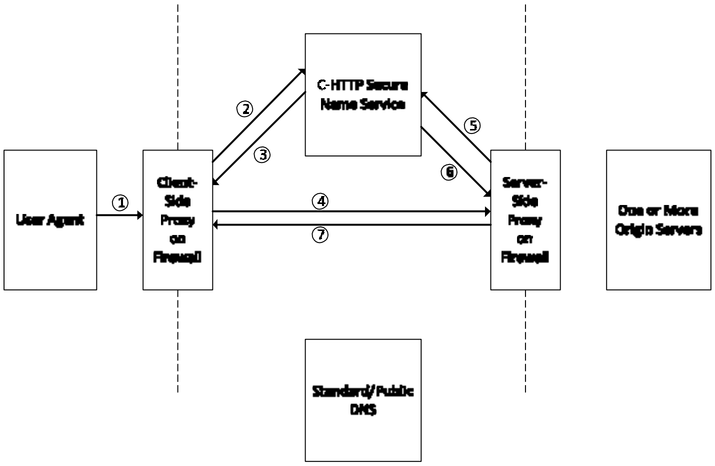
Overview of Kiuchi

Kiuchi describes a system and a protocol called “C-HTTP” that “provides secure HTTP communication mechanisms within a closed group of institutions on the Internet, where each member is protected by its own firewall.” Ex. 1018, p. 64, abstract. Kiuchi describes that C-HTTP can be used to create “a <u>closed</u> HTTP-based <u>virtual network</u> . . . for closed groups; for example, the headquarters and branches of a given corporation.” Ex. 1018, p. 69, § 5. The following Diagram 1 illustrates relevant parts within the C-HTTP system described by Kiuchi, and will be used to describe the C-HTTP system. *See* Ex. 1003, ¶ 56.



(Diagram 1)

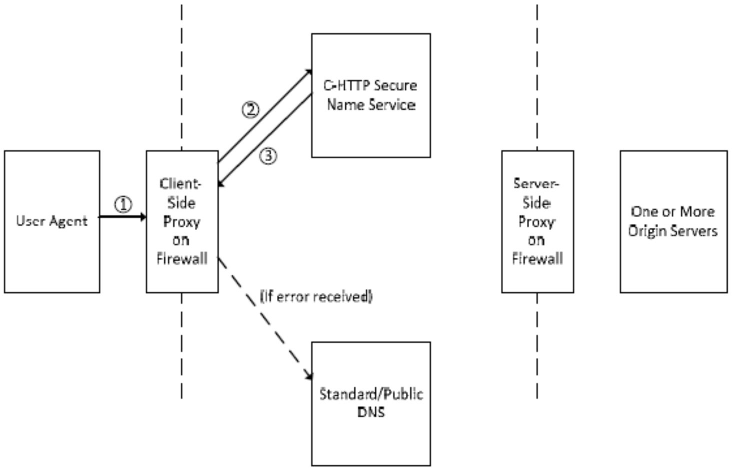
Leveraging these parts, Kiuchi describes a process by which a client-side proxy establishes a secure connection with a server-side proxy using the C-HTTP protocol over the Internet (i.e., a C-HTTP connection), thus establishing a closed virtual network including a user agent and one or more origin servers. *See* Ex. 1018, p. 64, § 2.1; p. 69, § 5; *see also* Ex. 1003, ¶ 57. Through the C-HTTP connection, a user agent associated with the client-side proxy may request information stored on one or more origin servers associated with the server-side proxy. *See id*. In order to establish a C-HTTP connection, Kiuchi teaches discrete steps that will be described using the following block diagram. *See* Ex. 1018, pp. 65-66, § 2.3; *see also*, Diagram 2, where each step is numbered to indicate a temporal sequence of the steps taught by Kiuchi (Ex. 1003, ¶ 58).


C-HTTP Secure Name Service
②
③
⑤
⑥
User Agent
①
Client-Side Proxy on Firewall
④
⑦
Server-Side Proxy on Firewall
One or More Origin Servers
Standard/Public DNS

(Diagram 2)

To enable initiation of this set of steps, the user agent displays HTML documents to an end-user. *See* Ex. 1018, p. 65, § 2.3. Through interaction with the user agent, the end user selects a hyperlink URL included within an HTML document. *See id.* Kiuchi provides an example of the selected URL: "http://server.in.current.connection/sample.html=@=6zdDfldfcZLj8V!i", where "server.in.current.connection" is the hostname, "sample.html" is the name of the resource being requested, and "6zdDfldfcZLj8V!i" is a connection ID. *See* Ex. 1018, p. 65, § 2.3*;* Ex. 1003, ¶ 59.

Thereafter, as illustrated by Diagram 3, initial steps are performed by Kiuchi's system in response to user selection of the hyperlink. These steps include: (1) a request being sent from the user agent to the client-side proxy for the selected URL; (2) a request being sent from the client-side proxy to the C-HTTP name server for an IP address corresponding to the hostname included in the selected URL; and (3) a response being returned from the C-HTTP name server that either includes the IP address associated with the server-side proxy or an error message. Ex. 1003, ¶ 63. If the C-HTTP name server returns an error message (i.e., if the hostname does not correspond to a secure server in the closed network, or the connection is not permitted), then the client-side proxy performs a DNS lookup using the standard/public DNS, as illustrated by the dashed line. *See* Ex. 1018, p. 65, § 2.3; *see also* Ex. 1003, ¶ 63.


C-HTTP Secure Name Service
②
③
User Agent
①
Client-Side Proxy on Firewall
(If error received)
Standard/Public DNS
Server-Side Proxy on Firewall
One or More Origin Servers

(Diagram 3)

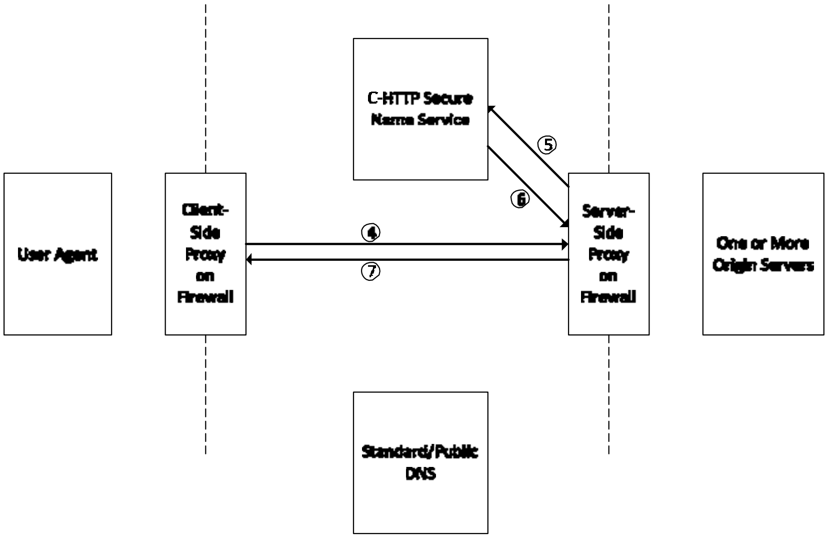
Analyzing these steps in further detail, when the end user selects the hyperlink in the displayed HTML document, the user agent sends a request for the URL to the client-side proxy, as illustrated by (1) in Diagram 3. *See* Ex. 1018, p. 65, § 2.3. When the client-side proxy receives the URL (including the hostname) from the user agent, it determines whether the connection ID included in the URL matches the IDs of any current connections being maintained by the client-side proxy. *See id.* If the connection ID is not found in the current connection table in the client-side proxy, the client-side proxy attempts to establish a new connection with the host corresponding to the hostname included in the URL. *See id.*

To establish a new connection with the corresponding host, as illustrated by (2) in Diagram 3, the client-side proxy sends a request to ask the C-HTTP name server whether the client-side proxy can communicate with the host associated with the hostname and, if so, to resolve the hostname included in the URL such that the corresponding IP address is returned by the C-HTTP name server to the client-side proxy. *See* Ex. 1018, p. 65, § 2.3(2). In some instances, the hostname corresponds to an origin server behind a server-side proxy and is associated with an IP address for the server-side proxy. *See* Ex. 1018, p. 65, § 2.3; *see also* Ex. 1003, ¶¶ 59-61. In other instances, the hostname corresponds to a server on the Internet outside the C-HTTP network. Ex. 1018, p. 65, § 2.3; Ex. 1003, ¶ 61.

Upon receipt of the request, the C-HTTP name server first authenticates the client-side proxy (and, by association, the user agent) to determine if the request is legitimate. *See* Ex. 1018, p. 65, § 2.3; *see also* Ex. 1003, ¶ 62. When the request is legitimate, the C-HTTP name server determines whether a "server-side proxy [associated with the hostname] is registered in the closed network." *See id.* As illustrated by (3) in Diagram 3, if a server-side proxy associated with the hostname is not registered in the closed network, or the connection is not permitted, then the C-HTTP name server returns an error message, in response to which the client-side proxy performs a look-up with a standard/public DNS server, behaving like an ordinary HTTP proxy. *See id.* The standard/public DNS server then returns an IP address of the host corresponding to the hostname, which the client-side proxy uses to connect to the host on behalf of the user agent. *See id.*

On the other hand, if the server-side proxy is registered in the closed network and is permitted to accept a connection from the client-side proxy, then the C-HTTP name server sends a response to the client-side proxy's request that includes "the IP address and public key of the server-side proxy and both request and response Nonce values," as illustrated by (3) in Diagram 3. *See* Ex. 1018, p. 65, § 2.3; Ex. 1003, ¶¶ 62-63. Notably, the C-HTTP name server never provides the IP address of the origin server to the client-side proxy. Ex. 1018 at 65, § 2.2; Ex. 1003, ¶ 64. Rather, when the C-HTTP name server returns the IP address of the server-side proxy along with the server-side proxy's public key and the nonce values, the client-side proxy attempts to establish a C-HTTP connection with the server-side proxy using the IP address received from the C-HTTP name server. *See* Ex. 1018, p. 65, § 2.3; Ex. 1003, ¶ 64. The steps for doing so are illustrated in Diagram 4. *See* Ex. 1003, ¶ 64.

In particular, Kiuchi describes that the client-side proxy, in response to receiving the IP address of the server-side proxy and other information from the C-HTTP name server, sends a "[r]equest for connection to the server-side proxy" (4), the server-side proxy performs a "[l]ookup of client-side proxy information" with the C-HTTP name server (5 and 6), and the server-side proxy sends confirmation of the connection to the client-side proxy (7), if the server-side proxy is able to properly authenticate the client-side proxy. *See* Ex. 1018, pp. 65-66, § 2.3, steps 3-5; Ex. 1003, ¶ 66.


C-HTTP Secure Name Service
⑤
⑥
User Agent
Client-Side Proxy on Firewall
④
⑦
Server-Side Proxy on Firewall
One or More Origin Servers
Standard/Public DNS

(Diagram 4)

Considering these steps in further detail, the client-side proxy, in response to receiving the IP address and associated information from the C-HTTP server, sends a request for connection to the server-side proxy, as illustrated by (4) in Diagram 4. *See* Ex. 1018, p. 65, § 2.3; Ex. 1003, ¶ 67. The client-side proxy encrypts the request for connection using the server-side proxy's public key and includes in the request "the client-side proxy's IP address, hostname, request Nonce value and symmetric data exchange key for request encryption." *See* Ex. 1066, p. 65, § 2.3; Ex. 1003, ¶ 67. After receiving the request, the server-side proxy "asks the C-HTTP name server whether the client-side proxy is an appropriate member of the closed network," as illustrated by (5) in Diagram 4, and, in response, the C-HTTP name server "examines whether the client-side proxy is permitted to access the server-side proxy." Ex. 1018, pp. 65-66, § 2.3; *see* Ex. 1003, ¶ 67. If the C-HTTP name server determines that "access is permitted, the C-HTTP name server sends [to the server-side proxy] the IP address and public key of the client-side proxy and both request and response Nonce values," as illustrated by (6) in Diagram 4. Ex. 1018, p. 66, § 2.3; *see* Ex. 1003, ¶ 67.

After "the C-HTTP name server provides both client-side and server-side proxies with each peer's public key," the proxies establish a C-HTTP connection. Ex. 1018 p. 66, § 2.3; *see* Ex. 1003, ¶ 68. The C-HTTP connection "provides [a] secure HTTP communication mechanisms" in which communications over the C-HTTP connection are encrypted. Ex. 1018, p. 64-66, abstract; *see* Ex. 1003, ¶ 68.

1. Kiuchi Anticipates Claim 1

Kiuchi discloses "*[a] data processing device, comprising memory storing a domain name server (DNS) proxy module that intercepts DNS requests sent by a client,*" as recited in claim 1. *See* Ex. 1003 at ¶¶ 57, 59-60. For example, Kiuchi's client-side proxy is a domain name server (DNS) proxy module that intercepts DNS requests sent by a user agent acting as a client. *See id.* Moreover, Kiuchi describes that the client-side proxy is a program installed on a firewall, so the client-side proxy is stored in the memory of a data processing device. *See* Ex. 1018 at p. 65, § 2.2.

In particular, as described above, Kiuchi's client-side proxy receives a request from the user agent. *See* Ex. 1018 at p 65, § 2.3; *see also* Ex. 1003, ¶¶ 57, 59-60. The user agent's request is a request for content corresponding to a hostname in a URL (*e.g.*, an HTML document). *See id.* The request from the client-side proxy therefore requests resources corresponding to the hostname.

In this case, the hostname is a "domain name," under that terms broadest reasonable construction, because the hostname corresponds to an IP address. The user agent is a "client," under that term's broadest reasonable construction, because the user agent is a computer or program from which a data request to a server is generated. The request from the user agent sent to the client-side proxy is a "DNS request," under that term's broadest reasonable construction, because the request is a request for a resource (*e.g.*, an HTML document) corresponding to a domain name (the hostname).

Kiuchi discloses the DNS proxy module performing the step of "*determining whether the intercepted DNS request corresponds to a secure server,*" under the broadest reasonable construction, as recited in claim 1. For example, as described above, the client-side proxy receives a request from the user agent, and the request includes a hostname. Ex. 1018, p. 65, § 2.3. In some instances, the hostname corresponds to an origin server behind a server-side proxy and designates the server-side proxy. Ex. 1018, p. 65, § 2.3; Ex. 1003, ¶ 61.

The origin server is a secure server, under the broadest reasonable construction of secure server, because authorization is needed to access the origin server and the origin server can communicate in an encrypted channel. In particular, as described above, before the server-side proxy will accept a client-side proxy's request for connection, the server-side proxy "asks the C-HTTP name server whether the client-side proxy is an appropriate member of the closed network" and, in response, the C-HTTP name server "examines whether the client-side proxy is permitted to access to the server-side proxy" and therefore the origin server. Ex. 1018, pp. 65-66, § 2.3; *see* Ex. 1003, ¶¶ 66-67. If the C-HTTP server determines that "access is permitted, the C-HTTP name server sends the IP address and public key of the client-side proxy and both request and response Nonce values," which are used to establish the C-HTTP connection. Ex. 1018, p. 66, § 2.3. As described above, the C-HTTP connection is an encrypted channel that allows communications between the user agent and the origin server via the server-side proxy. *See* Ex. 1003, ¶ 70. Accordingly, authorization is required to establish the C-HTTP connection and therefore to access the server-side proxy and origin server, and the C-HTTP connection is an encrypted channel in which the server-side proxy (and through it the origin server) can communicate. As such, both the server-side proxy and origin server are a secure servers, under that term's broadest reasonable interpretation.

Furthermore, the client-side proxy determines whether the request from the user agent corresponds to a secure server. *See* Ex. 1003, ¶ 65. In particular, when the client-side proxy receives the request from the user agent, the client-side proxy determines whether the request corresponds to a secure server by asking "the C-HTTP name server whether it can communicate with the host specified in a given URL." Ex. 1018 at p. 65, § 2.3; *see* Ex. 1003, ¶¶ 62-63, 65. If "the requested server-side proxy [associated with the hostname] is registered in the closed network," then the client-side proxy receives, from the C-HTTP server, "the IP address and public key of the server-side proxy and both request and response Nonce values." *Id*. Otherwise, if the server-side proxy associated with the origin server is not registered in the closed network, then the client-side proxy receives a "status code which indicates an error." *Id*.

<u>Kiuchi</u> discloses the DNS proxy module performing the step of *"when the intercepted DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure*

computer," as recited in claim 1. *See* Ex. 1003, ¶ 62. For example, as described above, in Kiuchi, if the client-side proxy receives from the C-HTTP name server the status code that indicates an error, (and therefore the request does not correspond to a secure server), then the client-side proxy "behave[s] like an ordinary HTTP/1.0 proxy" by "perform[ing] DNS loockup." *See* Ex. 1018 at p. 65, § 2.3; *see also* Ex. 1003 at ¶ 62. To do so, the client-side proxy sends a request to a standard/public DNS. *Id.* The standard/public DNS server returns an IP address corresponding to the hostname in the URL to the client-side proxy. *Id.*

Kiuchi discloses the DNS proxy module performing the step of *"when the intercepted DNS request corresponds to a secure server, automatically initiating an encrypted channel between the client and the secure server,"* as recited in claim 1. For example, as described above, Kiuchi describes that, if the client-side proxy receives from the C-HTTP name server "the IP address and public key of the server-side proxy and both request and response Nonce values" (and therefore the request corresponds to a secure server), the client-side proxy uses this information to initiate a sequence of steps for a secure C-HTTP session. Ex. 1018, p. 65, §§ 2.2-2.3; *see* Ex. 1003, ¶ 65-67. In particular, the client-side proxy first sends a request for connection to the server-side proxy. *Id*. The client-side proxy encrypts the request for connection using the server-side proxy's public key and includes in the request "the client-side proxy's IP address, hostname, request Nonce value and symmetric data exchange key for request encryption." *Id.* After receiving the request for connection from the client-side proxy, the server-side proxy verifies that "the client-side proxy is an appropriate member of the closed network" and, if so, receives from the C-HTTP server "the IP address and public key of the client-side proxy and both request and response Nonce values." Ex. 1018, p. 66, § 2.3. After both client-side and server-side proxies obtain each peer's public key, the proxies establish a C-HTTP connection. *See* Ex. 1018 p. 66, § 2.3; *see also* Ex. 1003, ¶ 68. The C-HTTP connection "provides [a] secure HTTP communication mechanisms" in which communications over the C-HTTP connection are encrypted." Ex. 1018, p. 64-66, abstract.

As a result, the client-side proxy initiates an encrypted channel between the user agent and the one or more origin servers via the server-side proxy. *See* Ex. 1003, ¶¶ 68, 70. In particular, by sending a request for connection to the server-side proxy, the client-side proxy initiates an encrypted channel on public communication paths between the user agent and the origin server (i.e., the communication path over the Internet between the client-side proxy and the server-side proxy). *See id.* Furthermore, this process is performed without the involvement of a user. *See* Ex. 1003, ¶ 69. As such, this channel is automatically initiated, under that term's broadest reasonable construction. Therefore, Kiuchi discloses that the client-side proxy automatically initiates an encrypted channel between the user agent (acting as a client) and the origin server (a secure server) via the server-side proxy (also a secure server).

2. Kiuchi Anticipates Claim 7

Kiuchi discloses "*a computer readable medium… comprised of computer readable instructions that, when executed cause a data processing device to perform the steps*" specified by those instructions, as recited in claim 7. *See* Ex. 1018 at p. 65, § 2.2. In particular, the client-side proxy described in Kiuchi contains computer readable instructions that cause the client-side proxy to implement the C-HTTP protocol. *Id.*

Steps (ii), (iii), and (iv) of claim 7 are identical to steps (i), (ii), and (iii) of claim 1. As explained in § V.A.1, above, Kiuchi describes systems that perform steps (ii), (iii), and claim 7. Claim 7 further specifies the step of "*(i) intercepting a DNS request sent by a client.*" As explained in § V.A.1, above, Kiuchi describes that the client-side proxy receives a request sent by the user agent. *See* Ex. 1018, p. 65, § 2.3. The request is a "DNS request," under that term's broadest reasonable construction, because the request is a request for a resource (*e.g.*, an HTML document) corresponding to a domain name (the hostname). Thus, Kiuchi shows "*(i) intercepting a DNS request sent by a client.*" *See* Ex. 1003, ¶¶ 59-60. Kiuchi therefore describes all the elements specified in claim 7, and anticipates this claim.

3. Kiuchi Anticipates Claim 13

Kiuchi discloses "*a computer readable medium storing a domain name server (DNS) module comprised of computer readable instructions that, when executed cause a data processing device to perform the steps of,*" as recited in claim 13. *See* Ex. 1018 at p. 65, § 2.2. In particular, the client-side proxy described in Kiuchi contains computer readable instructions that cause the client-side proxy to implement the C-HTTP protocol, which was described in § V.A.1, above. *Id.*

As explained in § V.A.1, Kiuchi discloses "*(i) determining whether a DNS request sent by a client corresponds to a secure server*" and "*(ii) when the DNS request does not correspond to a secure server, forwarding the DNS request to a DNS function that returns an IP address of a nonsecure computer;*" as recited in claim 13. *See* Ex. 1003, ¶¶ 62-65.

Kiuchi also discloses "(iii) when the intercepted DNS request corresponds to a secure server, automatically creating a secure channel between the client and *the secure server,*" as recited in claim 13. As explained in § V.A.1, Kiuchi describes that, if the client-side proxy determines that the query request from the user agent corresponds to a secure destination, then the client-side proxy uses "the IP address and public key of the server-side proxy and both request and response Nonce values" to send an encrypted request for connection to the server-side proxy for a secure C-HTTP session. Ex. 1018, p. 65, §§ 2.2-2.3; *see* Ex. 1003, ¶¶ 65-67. In response to receiving this request for connection, the server-side proxy verifies that the client-side proxy is a member of the closed network. *Id*. After verification, both client-side and server-side proxies use each peer's public key to establish a C-HTTP connection. Ex. 1018 p. 66, § 2.3; *see* Ex. 1003, ¶ 68. The C-HTTP connection "provides [a] **secure** HTTP communication mechanisms" in which communications over the C-HTTP connection are encrypted. Ex. 1018, p. 64-66, abstract.

As a result, the client-side proxy creates a secure channel between the user agent and the one or more origin servers via the server-side proxy. *See* Ex. 1003, ¶¶ 68, 70. In particular, by sending a request for connection to the server-side proxy, the client-side proxy creates a secure channel on the public communication paths from the user agent to the origin server (i.e., the communication path over the Internet between the client-side proxy and the server-side proxy). *See id.* Furthermore, this process is performed without the involvement of a user. *See* Ex. 1003, ¶ 69. As such, this channel is automatically created, under that term's broadest reasonable construction. Therefore, Kiuchi discloses that the client-side proxy automatically creates a secure channel between the user agent (acting as a client) and the origin server (a secure server) via the server-side proxy (also a secure server).

4. Kiuchi Anticipates Claims 2, 8, and 14

Claims 2, 8, and 14 depend from claims 1, 7, and 13, respectively, and specify "*wherein step (iii) comprises the steps of: (a) determining whether the client is authorized to access the secure server; and (b) when the client is authorized to access the secure server, sending a request to the secure server to establish an encrypted[/secure] channel between the secure server and the client.*" Kiuchi shows that when the client-side proxy receives an HTTP request from a user agent, it sends a request to a C-HTTP name server. *See* Ex. 1018, p. 65. § 2.3; *see also* Ex. 1003, ¶ 61. The C-HTTP name server authenticates the request and then evaluates it to determine if the connection is permitted. *See* Ex. 1018, p. 65-66, § 2.3; *see also* Ex. 1003 at ¶ 62. If the C-HTTP name server determines the connection is not permitted, it returns an error code. *See* Ex. 1018, p. 65. § 2.3; *see also* Ex. 1003, ¶¶ 62-63. If, on the other hand, the C-HTTP name server determines the connection is permitted, it returns "the IP address and public key of the server-side proxy and both request and response Nonce values." Ex. 1018 at p. 65, § 2.3(2); *see* Ex. 1003, ¶ 62. When the client-side proxy receives the IP address, public key, and Nonce values (as opposed to an error message), the client-side proxy sends a request to the server-side proxy (a secure server) to establish an encrypted channel between the server-side proxy and the user agent. *See* Ex. 1018 at p. 65, § 2.3(3); *see also* Ex. 1003 at ¶¶ 65-68, 70.

5. Kiuchi Anticipates Claims 6 and 12

Claims 6 and 12 depend from claims 1 and 7, respectively, and specify "*wherein automatically initiating the encrypted channel between the client and the secure server avoids sending a true IP address of the secure server to the client.*" As explained in § V.A.1, above, Kiuchi discloses that automatically initiating/creating the encrypted/secure channel between the user agent (acting as a client) and the origin server (a secure server) avoids sending a true IP address of the origin server to the user agent. Instead, as described in § V.A.1, Kiuchi discloses that, after the client-side proxy intercepts the query request from the user agent, the client-side proxy determines that the request from the user agent is authorized and uses the received IP address of the server-side proxy to send a request for connection to the server-side proxy. Ex. 1018, p. 65, § 2.3; *see* Ex. 1003, ¶ 65. Kiuchi discloses that "[f]rom the view of the **user agent** or client-side proxy, all resources appear to be located in a **server-side proxy** on the firewall. In reality, however, the server-side proxy forwards requests to the origin server." Ex. 1018 (Kiuchi) at 66. Therefore, the client-side proxy avoids sending the true IP address of the secure server to the client.

B. [GROUND 2] – Kiuchi In View of RFC 2660 Renders Obvious Claims 1, 2, 6-8, and 12-14

Claims 1, 2, 6-8, and 12-14 are anticipated by Kiuchi for the reasons set forth in §§ V.A.1 to V.A.5, above. To the extent Patent Owner contends that Kiuchi does not expressly show automatically initiating/creating an encrypted/secure channel between the client and the secure server– a position that would be inconsistent with its prior interpretations of the limitation – the claims are still invalid as obvious. In particular, were the Patent Owner to contend that an encrypted/secure channel "between" a client and a secure server must be an encrypted/secure channel that extends from the client to the secure server, rather than just an intermediate portion there-between, a person of ordinary skill in the art in February of 2000 would have considered these claims obvious based on Kiuchi in view of, *inter alia*, the information in an early draft of RFC 2660 (hereinafter "RFC 2660" or Ex. 1010). *See* Ex. 1003 ¶¶ 71-75.

In particular, a person of ordinary skill would have considered it obvious to configure Kiuchi's user agent and origin server to implement "end-to-end secure transactions" using the Secure HTTP (S-HTTP) protocol disclosed in RFC 2660. Kiuchi opens the door to this possibility through the following: "[a]lthough the current C-HTTP implementation assumes the use of HTTP/1.0 compatible user agents and servers, it is possible to develop C-HTTP proxies which can communicate with other secure HTTP compatible user agents and servers." Ex. 1018 at p. 69, § 4.4; *see* Ex. 1003, ¶ 72. To permit this, Kiuchi describes that C-HTTP "can co-exist with" other secure HTTP proposals. *See id.* Kiuchi also describes the motivation to do so by describing the resulting benefit of assuring both institutional and personal level security: "[i]f C-HTTP is used with these protocols, which assure end-to-end or individual security, both institutional and personal level security protection can be provided." *Id.* As an example of a secure HTTP protocol that can be used at a user agent and at an origin server, Kiuchi refers to an early draft of RFC 2660. *See* Ex. 1018 at p. 70, n 12.

RFC 2660 discloses the use of encryption between clients and servers: "Secure HTTP (S-HTTP) provides secure communication mechanisms between an HTTP client-server pair in order to enable spontaneous commercial transactions for a wide range of applications." Ex. 1010 at § 1; see Ex. 1003, ¶ 73. In particular, RFC 2660 describes that "Secure HTTP supports a variety of security mechanisms to **HTTP clients** and **servers**" and that "[s]everal **cryptographic** message format standards may be **incorporated** into S-HTTP **clients** and **servers**." Ex. 1010 at § 1.1. "S-HTTP provides full flexibility of cryptographic algorithms, modes and parameters." Ex. 1010 at § 1.1.

The combination of Kiuchi and RFC 2660 would result in encrypted communications between the user agent and origin server using S-HTTP messages instead of standard HTTP/1.0 messages. *See* Ex. 1003 at ¶ 73. In this way, one of ordinary skill in the art would understand that the use of S-HTTP could simply replace the HTTP 1.0 messages otherwise employed in the C-HTTP security scheme described by Kiuchi. *See id.* As described by Kiuchi, "[t]his means that even if individual security management is not sufficient, data security can be guaranteed. In this case, administrators of proxies on the firewall cannot know the contents of any information exchanged." Ex. 1018 at p. 69, § 4.4.

Thus, upon receipt of an S-HTTP compliant request from the user agent for information stored on an origin server, the client-side proxy would automatically establish a C-HTTP connection with the server-side proxy, as described above, and the exchange of the S-HTTP messages would ensure end-to-end encryption between the user agent and origin server. *See* Ex. 1003, ¶ 74. If, on the other hand, the user agent is requesting information from a non-secure server outside the C-HTTP network that does not implement S-HTTP, the user agent would communicate using standard HTTP. *See* Ex. 1010 at § 1.1 ("S-HTTP supports interoperation among a variety of implementations, and is compatible with HTTP. S-HTTP aware clients can communicate with S-HTTP oblivious servers and *vice-versa*, although such transactions obviously would not use S-HTTP security features."); *see also* Ex. 1003, ¶ 74.

Therefore, based on the motivation provided in Kiuchi, it would have been an obvious design choice to one of ordinary skill in the art to incorporate the cryptography provided by Secure HTTP, as taught by RFC 2660, into Kiuchi's user agent and origin server, in order to provide end-to-end encryption and personal-level security. *See* Ex. 1003, ¶ 75. Therefore, Kiuchi (which discloses an encrypted/secure C-HTTP connection from the client-side proxy to the server-side proxy) in view of RFC 2660 (which discloses encrypted/secure end-to-end communications between the user agent and origin server) discloses an encrypted/secure channel that starts at the user agent (acting as a *client*) and ends at the origin server (a *secure server*). *See* Ex. 1003, ¶ 75.

VI. CONCLUSION

The cited prior art references identified in this Petition contain pertinent technological teachings (both cited and uncited), either explicitly or inherently disclosed, which were not previously considered in the manner presented herein, or relied upon on the record during original examination of the '151 patent. In sum, these references provide new, noncumulative technological teachings which indicate a reasonable likelihood of success as to Petitioner's assertion that the Challenged Claims of the '151 patent are not patentable pursuant to the grounds presented in this Petition. Accordingly, Petitioner respectfully requests institution of an IPR for those claims of the '151 patent for each of the grounds presented herein.

Dated: October 31, 2014 Respectfully Submitted,

/Jeffrey P. Kushan/
Jeffrey P. Kushan
Registration No. 43,401
Sidley Austin LLP
1501 K Street NW
Washington, DC 20005

PETITION FOR INTER PARTES REVIEW

OF U.S. PATENT NO. 7,490,151

Attachment A:

Proof of Service of the Petition

CERTIFICATE OF SERVICE

I hereby certify that on this 31st day of October 2014, a copy of this Petition, including all attachments, appendices and exhibits, has been served in its entirety by Federal Express on the following counsel of record for patent owner:

Joseph E. Palys
Paul Hastings LLP
875 15th Street NW
Washington, DC 20005
Telephone: (202) 551-1996
E-mail: josephpalys@paulhastings.com

Naveen Modi
Paul Hastings LLP
875 15th Street NW
Washington, DC 20005
Telephone: (202) 551-1990
E-mail: naveenmodi@paulhastings.com

Jason Stach
FINNEGAN, HENDERSON, FARABOW, GARRETT & DUNNER LLP
901 NEW YORK AVENUE, NW
WASHINGTON DC 20001-4413

Dated: October 31, 2014

Respectfully submitted,

/Jeffrey P. Kushan/
Jeffrey P. Kushan
Reg. No. 43,401
Attorney for Petitioner

PETITION FOR INTER PARTES REVIEW

OF U.S. PATENT NO. 7,490,151

Attachment B:

List of Evidence and Exhibits Relied Upon in Petition

Exhibit #	Reference Name
1001	U.S. Patent 7,490,151 (Munger)
1002	Excerpts from the Prosecution History of U.S. Patent 7,490,151
1003	Declaration of Dr. Roch Guerin
1004	Curriculum Vitae of Dr Roch Guerin
1005	Declaration of Chris A. Hopen
1006	Declaration of James Chester
1007	Aventail Connect v3.01/2.51 Administrator's Guide
1008	Mockapetris - RFC 1034 - Domain Names-Concepts and Facilities
1009	[RESERVED]
1010	Rescorla - Draft 01 of RFC 2660
1011	Bradner - RFC 2026 - The Internet Standards Process
1012	[RESERVED]
1013	[RESERVED]
1014	[RESERVED]
1015	VirnetX's Reply Claim Construction Brief
1016	[RESERVED]
1017	[RESERVED]
1018	KIuchi - C-HTTP-The Development of Secure Closed HTTP-based Network on the Internet
1019	Patent Owner's Preliminary Response in IPR2013-00354
1020	[RESERVED]
1021	[RESERVED]

Exhibit #	**Reference Name**
1022	[RESERVED]
1023	[RESERVED]
1024	[RESERVED]
1025	[RESERVED]
1026	[RESERVED]
1027	Excerpts from the History of Inter Partes Reexamination No 95001697
1028	Petition for Inter Partes Review filed in IPR2013-00376