

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

CIENA CORPORATION,
Petitioner,

v.

K. MIZRA LLC,
Patent Owner.

U.S. Patent 8,782,282

Title: NETWORK MANAGEMENT SYSTEM

Inter Partes Review No.: IPR2025-01362

**PETITION FOR *INTER PARTES* REVIEW OF U.S. PATENT 8,782,282
UNDER 35 U.S.C. §§311-319 AND 37 C.F.R. §§42.1-.80, 42.100-.107**

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. OVERVIEW	1
III. GROUNDS FOR STANDING UNDER 37 C.F.R. §42.104(a).....	5
IV. REASONS FOR THE REQUESTED RELIEF	6
A. Overview of the '282 Patent.....	6
B. Prosecution History of the '282 Patent	9
C. Priority Date of the Challenged Claims	11
D. Claim Construction	11
E. Level of Ordinary Skill in the Art	11
F. State of the Art	12
1. Network Management Systems/Operations Support Systems .	12
2. Adapters	18
3. Load Balancing	19
4. Failover and Recovery	24
V. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE.....	29
A. Ground 1: Claims 1-22 Are Obvious over <i>Secer</i> in View of <i>Dinker</i>	29
1. <i>Secer</i> (EX-1004)	29
2. <i>Dinker</i> (EX-1005)	33
3. Motivation to Combine <i>Secer</i> and <i>Dinker</i>	35
4. Detailed Application of <i>Secer</i> in Combination with <i>Dinker</i> to the Challenged Claims	43
Claim 1	43
[1pre] A method, comprising:.....	43
[1ai] receiving, at a first application server instance selected from a plurality of application server instances based on a load balancing process, first adapter processed information from a first adapter,.....	43

[1aii] wherein the first adapter processed information comprises event information received by the first adapter from a network element and processed by the first adapter based on a first communication protocol;50

[1b] processing, by the first application server instance, the first adapter processed information based on an event management service to produce application processed information;.....52

[1ci] sending, by the first application server instance, the application processed information to a gateway device,.....54

[1cii] wherein the gateway device is one of a plurality of gateway devices respectively associated with the plurality of application server instances and is configured to transfer the application processed information to a second adapter of a plurality of second adapters configured to process the application processed information based on a second communication protocol to produce second adapter processed information and transfer the second adapter processed information to an operation support system device; and.....57

[1d] in response to determining that the first application server instance has become disabled, facilitating establishing an association between the first adapter and a second application server instance of the plurality of application server instances and between the gateway device and the second application server instance.61

Claim 267

[2] The method of claim 1, wherein the first application server instance is configured to execute on a separate physical machine from the first adapter67

Claim 368

[3pre] A system, comprising:.....68

[3a] a first application server instance configured to receive first adapter processed information from a first adapter[,] process the first adapter processed information based on an event management service to yield application processed information, and68

[3b] send the application server processed information to a gateway device,69

[3c] wherein the first adapter processed information comprises event information from a network element that has been processed by the first adapter based on a first communication protocol; and69

[3d] a load balancing component configured to select the first application server instance from a plurality of application server instances based on a load balancing process;69

[3e] wherein the gateway device is one of a plurality of gateway devices respectively associated with the plurality of application server instances and is configured to transfer the application server processed information to a second adapter of a plurality of second adapters configured to process the application server processed information based on a second communication protocol to yield second adapter processed information, and.....69

[3f] send the second adapter processed information to an operation support system device, and69

[3g] wherein, the first adapter and the gateway device are further configured to, in response to disablement of the first application server instance, establish an association with a second application server instance of the plurality of application server instances.70

Claim 470

[4] The method of claim 1, further comprising converting, by the second adapter, a protocol specific message associated with the first application server instance to a formatted message associated with the second communication protocol.....70

Claim 572

[5] The method of claim 1, further comprising selecting the first application server instance based on a determination that the first application server instance has a lowest processing load of the plurality of application servers instances.72

Claim 673

[6] The method of claim 1, wherein the first communication protocol and the second communication protocol comprise one or more communication protocols associated with at least one of extensible markup language, simple network management protocol, common object request broker architecture, or transaction language 1.....73

 Claim 774

[7] The method of claim 1, further comprising at least one of collecting, recording, or publishing the event information in accordance with the event management service for access by the operation support system device.74

 Claim 875

[8] The method of claim 1 wherein the plurality of gateway devices comprise a plurality of physically or logically separated gateway devices.75

 Claim 977

[9] The method of claim 1, wherein the gateway device is a physically and logically separate machine from the first application server instance.77

 Claim 1079

[10] The method of claim 1, further comprising converting, by the first adapter, a message associated with the first communication protocol to a protocol specific message associated with the first application server instance.79

 Claim 1180

[11] The method of claim 1, further comprising processing, by the gateway device, a function associated with the application server processed information.80

 Claim 1281

[12] The method of claim 1, wherein the event information relates to a configuration of the network element.....81

 Claim 1381

[13] The method of claim 1, further comprising executing the first adapter and the second adapter on one or more virtual machines.....81
 Claim 1482

[14] The method of claim 1, wherein the first adapter and the second adapter are configured to communicate with the first application server instance over a remote method invocation interface.82
 Claim 1583

[15] The system of claim 3, wherein the event information is associated with a configuration of the network element.....83
 Claim 1683

[16] The system of claim 3, wherein the first application server instance comprises a performance manager component configured to collect performance data from a plurality of network elements respectively associated with a plurality of first adapters including the first adapter.....83
 Claim 1784

[17] The system of claim 16, wherein the performance manager component is further configured to deliver at least a subset of the performance data to one or more of the plurality of network elements.84
 Claim 1886

[18] The system of claim 3, wherein the first communication protocol and the second communication protocol comprise one or more communication protocols associated with at least one of extensible markup language, simple network management protocol, common object request broker architecture, or transaction language 1.....86
 Claim 1986

[19] The system of claim 3, wherein at least one of the first adapter or the second adapter are executed on one or more virtual machines.86
 Claim 2086

[20] The system of claim 3, wherein the first adapter and the second adapter are configured to communicate with the first application server instance over a remote method invocation interface.	86
Claim 21	87
[21] The system of claim 3, wherein the load balancing component is configured to select the first application server instance based on a determination that the first application server instance has a lowest processing load of the plurality of application server instances.....	87
Claim 22	87
[22] The system of claim 3, wherein the plurality of gateway devices comprises a plurality of physically or logically separated gateway devices.	87
VI. MANDATORY NOTICES – 37 C.F.R. §42.8.....	87
A. Real Parties-In-Interest Under 37 C.F.R. §42.8(b)(1).....	87
B. Related Matters Under 37 C.F.R. §42.8(b)(2)	87
1. Judicial Matters	87
2. Administrative Matters	88
3. Related Patents	88
C. Lead and Back-Up Counsel Under 37 C.F.R. §42.8(b)(3)	88
D. Service Information Under 37 C.F.R. §42.8(b)(4).....	89
E. Payment of Fees – 37 C.F.R. §42.103.....	90
VII. CONCLUSION.....	90

PETITIONER'S EXHIBIT LIST

Exhibit	Brief Description
1001	U.S. Patent 8,782,282 (“the ’282 Patent”)
1002	Prosecution History for U.S. Patent 8,782,282
1003	Declaration of Dr. Douglas Schmidt
1004	U.S. Patent No. 7,209,968 (“ <i>Secer</i> ”), issued April 24, 2007, filed May 29, 2001
1005	U.S. Patent Pub. No. 2003/0177411 (“ <i>Dinker</i> ”), published September 18, 2003, filed March 12, 2002
1006	WIPO Publication WO 00/08823 (“ <i>Keene</i> ”), published February 17, 2000
1007	U.S. Patent Pub. No. 2003/0028654 (“ <i>Abjanic</i> ”), published February 6, 2003, filed August 10, 2001
1008	U.S. Patent No. 6,718,377 (“ <i>Bischoff</i> ”), issued April 6, 2004, filed March 15, 2000
1009	U.S. Patent No. 5,742,762 (“ <i>Scholl</i> ”), issued April 21, 1998
1010	U.S. Patent No. 7,043,525 (“ <i>Tuttle</i> ”), issued May 9, 2006, filed December 14, 2001
1011	U.S. Patent No. 6,470,394 (“ <i>Bamforth</i> ”), issued October 22, 2002
1012	U.S. Patent Pub. No. 2004/0008717 (“ <i>Verma</i> ”), published January 15, 2004, filed July 12, 2002
1013	U.S. Patent 6,260,062 (“ <i>Davis</i> ”), issued July 10, 2001
1014	U.S. Patent Pub. No. 2003/0120502 (“ <i>Robb</i> ”), published June 26, 2003, filed April 23, 2002

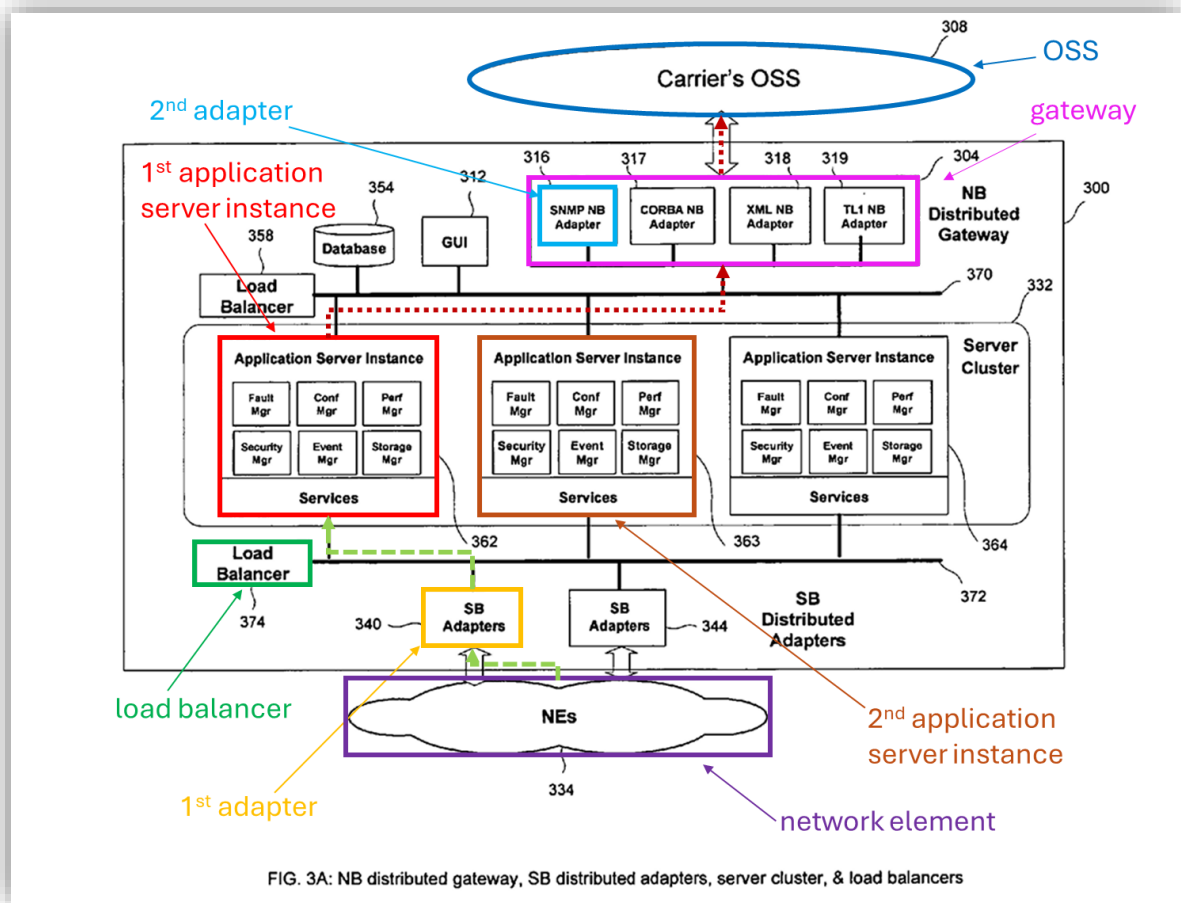
I. INTRODUCTION

Ciena Corporation (“Petitioner”) hereby petitions for an *inter partes* review (IPR) of U.S. Patent 8,782,282 (“the ’282 Patent”, EX-1001). Petitioner respectfully submits that Claims 1-22, (the “Challenged Claims”) of the ’282 Patent are unpatentable under Pre-AUA 35 U.S.C. §103 over in *Secer* (EX-1004) in view of *Dinker* (EX-1005).

This Petition demonstrates by a preponderance of the evidence that there is a reasonable likelihood that Petitioner will prevail with respect to at least one of these claims. Accordingly, it is respectfully requested that the Board institute an *inter partes* review.

II. OVERVIEW

The ’282 Patent is directed to a distributed network management system (“NMS”) in which communication adapters are distributed from the application servers of the NMS. The distributed architecture facilitates scalability in managing network elements and communicating information to a network administrators’ operations support systems (OSS). FIG. 3A shows one example of an NMS, annotated relative to pertinent limitations of the Challenged Claims:



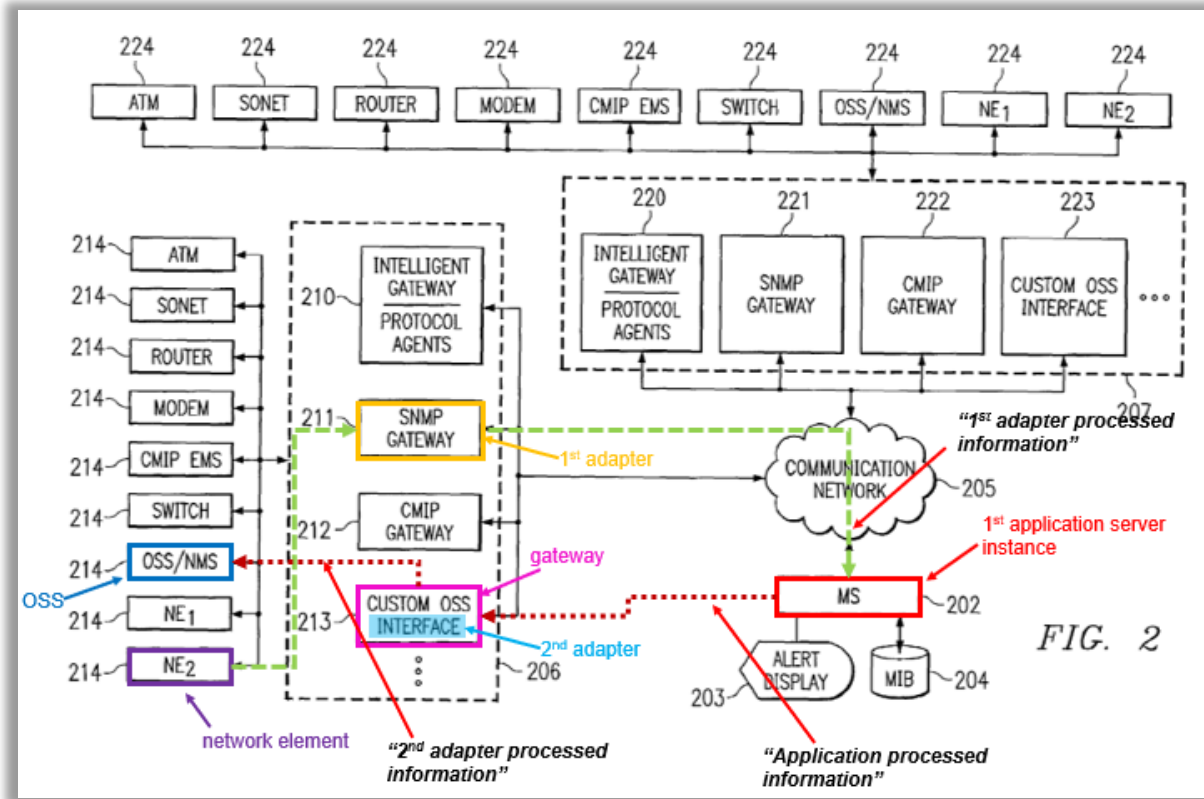
EX-1001 FIG. 3A (annotated). EX-1003 ¶31.

Network elements (NEs) (purple) provide management information (e.g., data, messages or events) to an application server (red) via an adapter (gold). The adapter facilitates transferring and translating between NEs and the application servers using one of any number of different protocols (dashed green arrow). A load balancer (green) uses any appropriate load balancing algorithm to distribute the management information received by the adapter, from the NE, to one of the application servers of the cluster. The application server processes the information and transfers the processed information to a gateway (pink), which includes an

adapter (light blue) that processes the information and translates and transfers the information to an OSS (blue) using one of any number of different protocols (dashed red arrow). EX-1003 ¶32.

During prosecution, the Applicant amended the claims to recite a “failover” feature (*see, e.g.*, [1d]) before allowance. The failover feature required that, in response to a first application server becoming disabled, an association is established between a second application server and the adapter and gateway that were previously associated with the first application server. However, such a failover feature was well known in the prior art as discussed in Ground 1 and §IV.F (State-of-the-Art Section). EX-1003 ¶33.

For example, *Secer* (EX-1004) disclosed a distributed NMS that includes a nearly identical architecture to the distributed NMS disclosed in the '282 Patent that facilitates scalability in managing NEs and communicating information to network administrator's OSS. FIG. 2 of *Secer* disclosed an NMS where elements in common with the '282 Patent's NMS are annotated in the same color used above:



EX-1004 FIG. 2 (annotated). EX-1003 ¶34.

NEs (purple) provided management information including messages or events to an application server (*Secer's* management server MS 202) (red) via an adapter (*Secer's* "gateway") (gold) (dashed green arrow). *Secer's* adapters facilitated transferring and translating information between NEs and the application server using different protocols. The application server processed the information and transferred the information to a gateway (pink) (dashed red arrow). The gateway included an adapter (light blue) that processed the information and transferred and translated the information to an OSS (blue) using

an appropriate protocol (dashed red arrow). *Secer*'s NMS also used load balancing and addressed failover. EX-1003 ¶35.

While *Secer* did not expressly disclose the use of multiple application server instances, and its load balancer and failover feature were expressly discussed for its gateways, failover and load balancing of multiple application servers was widely known and routinely implemented, as demonstrated by *Dinker* (EX-1005), as corroborated by the State-of-the-Art section (§IV.F). For example, *Dinker* demonstrated that it was known to use load balancing and failover protection with distributed application server clusters. A POSITA would have found it obvious to extend *Secer*'s load balancing and failover concepts to *Secer*'s application server 202 following *Dinker*'s teaching of multiple application server clustering, load balancing, and failover protection teachings to achieve improved scalability, performance, efficiency, and fault-tolerance, as explained further in Ground 1 below. EX-1003 ¶35.

Thus, the Challenged Claims are obvious over *Secer* in view of *Dinker* as demonstrated by this Petition and the evidence cited herein. EX-1003 ¶36.

III. GROUNDS FOR STANDING UNDER 37 C.F.R. §42.104(a)

Petitioner certifies that the '282 Patent is available for IPR.

IV. REASONS FOR THE REQUESTED RELIEF

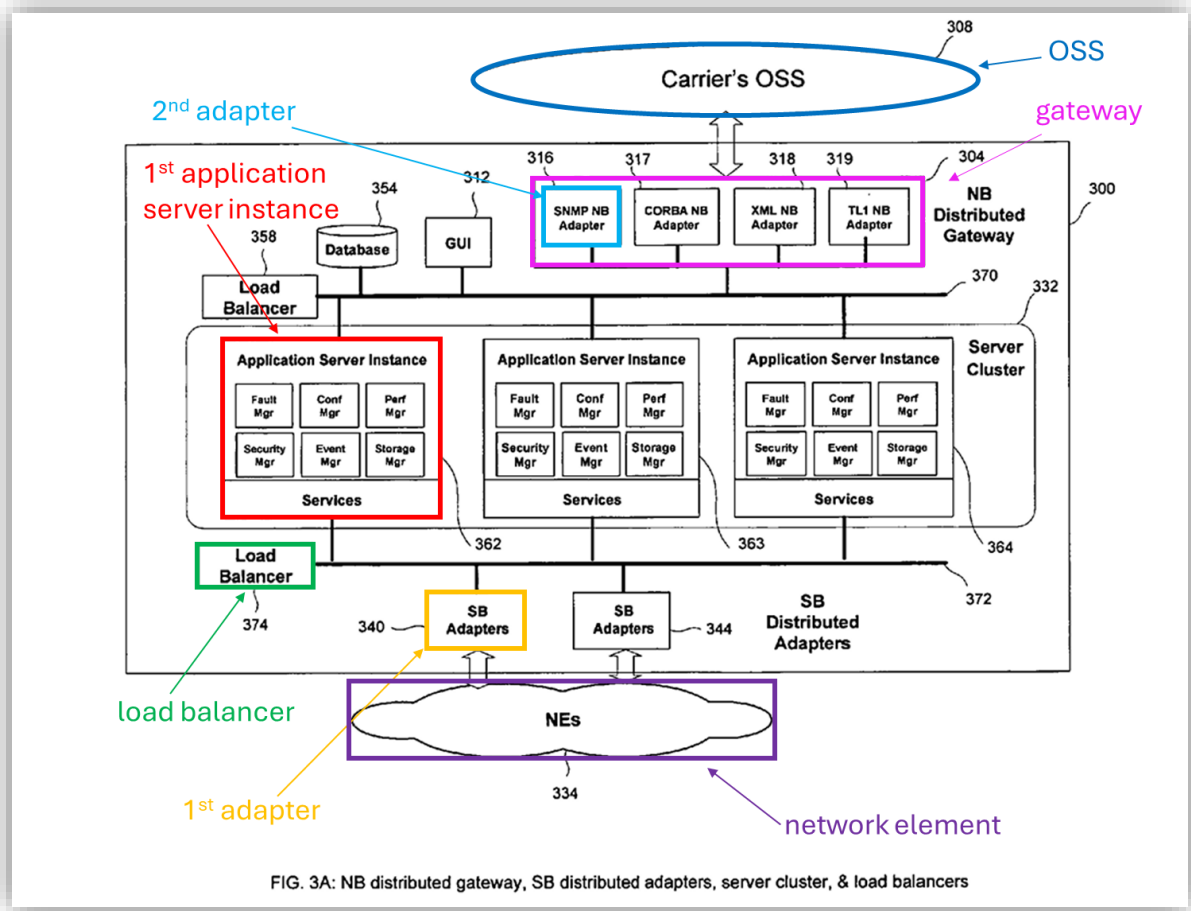
As explained in §§II and IV-V of this Petition and in the attached Declaration of Petitioner's Expert, Dr. Schmidt (EX-1003), the subject matter claimed in the '282 Patent was obvious over the prior art to a POSITA at the time of the invention.

A. Overview of the '282 Patent

The '282 Patent is directed to a network management system (NMS). Network administrators conventionally use NMSs to facilitate the exchange of management information between Network Elements (NEs) and network administrators' Operations Support Systems (OSS) to, e.g., install, configure, and manage NEs. EX-1001 1:11-17; EX-1003 ¶37.

The patent purported to provide a more robust and scalable NMS. EX-1003 ¶38.

The '282 Patent discloses an example NMS 300 in FIG. 3A:



EX-1001 FIG. 3A (annotated). EX-1001 7:43-46. NMS 300 includes an application server instance (red), load balancer (green), southbound (SB) adapters (gold), “[northbound] NB gateway (gateway)” (pink) having an adapter (light blue) and OSS (blue). EX-1001 7:43-46; EX-1003 ¶39.

The '282 Patent describes that, in one embodiment, NEs (purple) provide management information (e.g., traps/alarms) in the uplink (northbound) direction to a southbound adapter (gold). The '282 Patent describes that adapters facilitate transferring and translating information between NEs, the application server(s) and OSS (blue) using different protocols. EX-1001 2:12-16; 2:53-61. The arriving

information at the SB adapter is then transferred to a selected server (red) according to load balancer 374 (green). “Load balancer 374 may be a software module physically located on any device in NMS 300” and “may use any load balancing algorithm that is appropriate.” For example, load balancer 374 may use a round robin scheme or distribute the load based on CPU usage. In some embodiments, all information from the SB adapter is passed to the selected server (red) for processing. The information is then transferred from the server to a selected NB gateway (pink). When the information arrives at the NB gateway, it is transferred to a NB adapter (light blue) that uses the NB interface protocol for communicating with the OSS (blue). The information is then transferred from the NB adapter to the OSS. EX-1001 9:1-24; EX-1003 ¶40.

The '282 Patent claims were allowed after the addition of a “failover” feature (*see, e.g.*, [1d], §IV.B) where, in response to a first instance of the application server being disabled, a second application server instance gets associated with the software modules (e.g., gateways and adapters) that were associated with the disabled server. The '282 Patent provides sparse description of this “failover” feature:

An instance of the server can also be shut down for any reason without interrupting the server functionality as a whole. When a server is shutdown, its associated software modules (NB gateway 304, GUI 312, or SB adapters 340 or 344) automatically re-establish the association

with another server instance (one of existing servers 362-364 or a standby server) based on certain criterion, such as selecting the lightest loaded server.

EX-1001 9:58-65. These two sentences do not include the technical details of how or why the application server may become disabled, nor do they include the technical details of how the software modules (gateway and adapter) are associated with the second application server instance. As the '282 Patent states: “technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.” EX-1001 2:8-11. It is not surprising that the '282 Patent does not include a detailed description of the “failover” feature because this feature was well known to a POSITA. §IV.F.4; §V; EX-1003 ¶41.

As demonstrated below, the Challenged Claims, including the “failover” feature, are disclosed and/or rendered obvious by *Secer* in view of *Dinker*. EX-1003 ¶43.

B. Prosecution History of the '282 Patent

The '282 Patent was filed December 19, 2003 with 32 claims, including independent Claim 18 (which would be amended and ultimately issue as Claim 1):

18. A method of communicating in a network management system, including:
communicating an event from a network element to a configured adapter wherein
the configured adapter is preselected to handle events from the network element; and
transferring information associated with the event from the configured adapter to
an application server.

EX-1002, 0484-487, 0079; EX-1001 (21-22).

Following multiple rejections, claim amendments, and arguments, the applicant finally added the “failover” feature of limitation [1d] shown below, after which the Examiner allowed the claims, highlighting this limitation:

2. The Office deems Applicant’s latest claim amendments persuasive to overcome the rejection of the claims over the applied prior art references and/or any other candidate prior art, and the claims are accordingly considered in condition for allowance (**MPEP § 1302.14**).

In addition, Applicants remark and/or argument that the current prior art references do not sufficiently teach or disclose *all* of the recited limitations of the amended independent claims, and claim 1 in particular, including the recited feature of “in response to determining that the first application server instance has become disabled, facilitating establishing an association between the first adapter and a second application server instance of the plurality of application server instances and between the gateway device and the second application server instance...” as recited [Remarks: par 2, pg. 9 & par 6, pg. 13], is also considered persuasive by the Office.

EX-1002, 0041.¹

Again, this tersely described “failover” feature (§IV.A) was already well known. §§II, IV.F, V; EX-1003 ¶46.

C. Priority Date of the Challenged Claims

The ’282 Patent does not contain any priority claim. This Petition assumes the Priority Date for the ’282 Patent is the filing date of U.S. Application 10/742,573, December 19, 2003. EX-1001 (21-22).

D. Claim Construction

Petitioner proposes that each claim term in the Challenged Claims be given its plain and ordinary meaning in this proceeding, and that no specific construction of any claim term is required because the prior art relied on in this Petition meets each of the claim terms under any reasonable construction. EX-1003 ¶48.

E. Level of Ordinary Skill in the Art

A Person of Ordinary Skill in the Art (“POSITA”) in December 19, 2003 would have been someone knowledgeable and familiar with computer network management systems. A POSITA would have gained knowledge of these concepts through a mixture of training and work experience, such as by having at least a bachelor’s degree in electrical engineering, computer science or related field, and

¹ The Notice of Allowance incorrectly references “claim 1” and should reference claim 18, as claim 1 was canceled during prosecution. EX-1002, 0308-313.

at least two to three years of training or additional work experience in the domain of computer network management systems, or a related field. Additional hands-on and design experience could compensate for less formal education, and vice versa. The knowledge and skill of a POSITA is further reflected in the prior art references themselves, as well as the State of the Art, discussed below. EX-1003 ¶¶50-53.

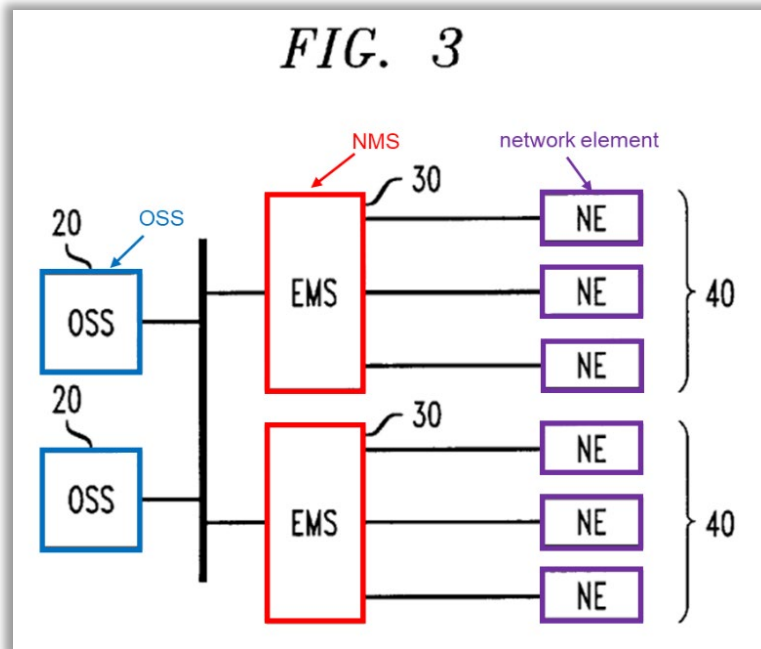
F. State of the Art

This section describes the state of the art as of the Priority Date. This section, and the expert testimony and documentary evidence cited, provide additional factual support for the general knowledge and skill of a POSITA at the Priority Date. Additionally, this section provides additional factual support for and motivation to modify and combine the teachings of *Secer and Dinker*, and further demonstrates why doing so would involve a reasonable expectation of success. EX-1003 ¶¶54-73.

1. Network Management Systems/Operations Support Systems

By December 2003, network management systems (NMS) and operations support systems (OSS) were well known. For example, U.S. 6,718,377 (“*Bischoff*,” EX-1008) disclosed that OSSs allowed users to “efficiently and effectively manage (i.e., monitor, regulate, configure, etc.)” network elements. EX-1008 1:17-26; EX-1014 ¶77 (OSS functionality included “system management, network management”). The OSS (blue) controlled, monitored,

managed, maintained, and performed the functions that kept the network operating efficiently:



EX-1008 1:24-30, FIG. 3 (annotated). OSSs “typically” interfaced with other components to manage NEs, e.g., the element management systems (EMSs) (red).

EX-1008 1:44-46. *Bischoff* disclosed the EMS in connection with the management and administration of one or more NEs. EX-1008 1:29-31. *Bischoff* further disclosed communications between the OSS and EMS used the Common Object Request Broker Architecture (CORBA) protocol. EX-1008 4:63-67. Thus, the EMS performed network management tasks and acted as an intermediary between the OSS and NEs. Moreover, it was known to use an OSS in the management of NEs using components between the OSS and the managed NEs. EX-1003 ¶55.

As another example, U.S. 5,742,762 (“Scholl,” EX-1009) disclosed

“traditional network management service applications”:

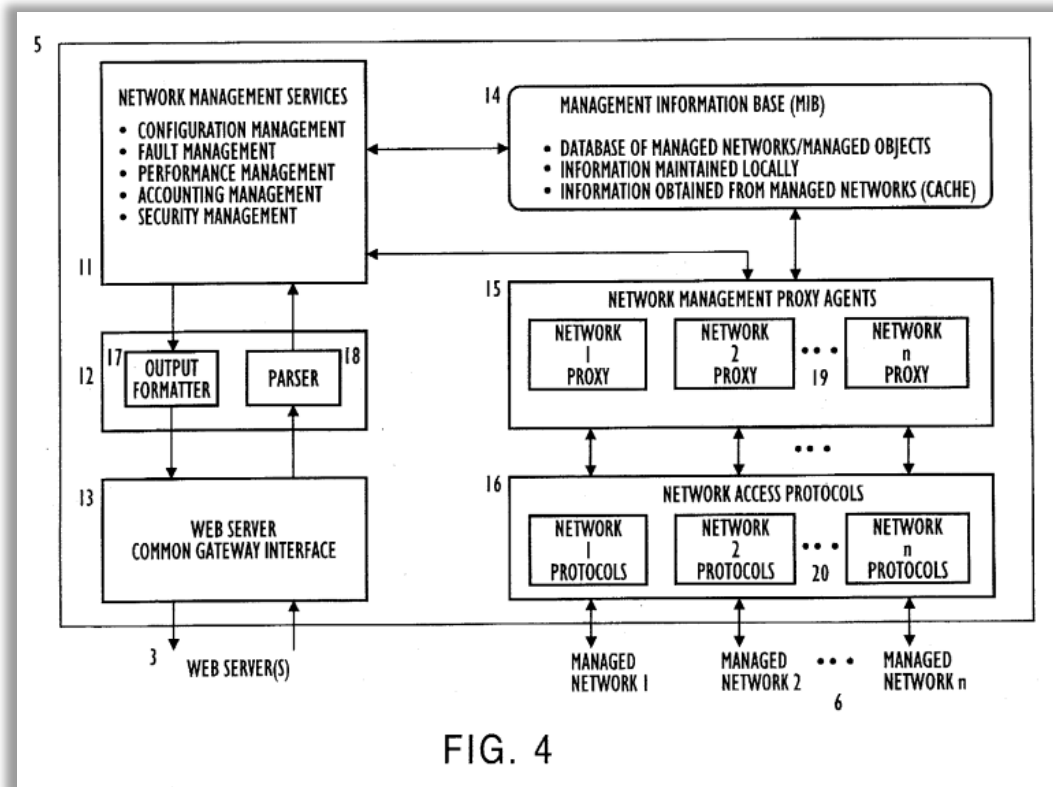
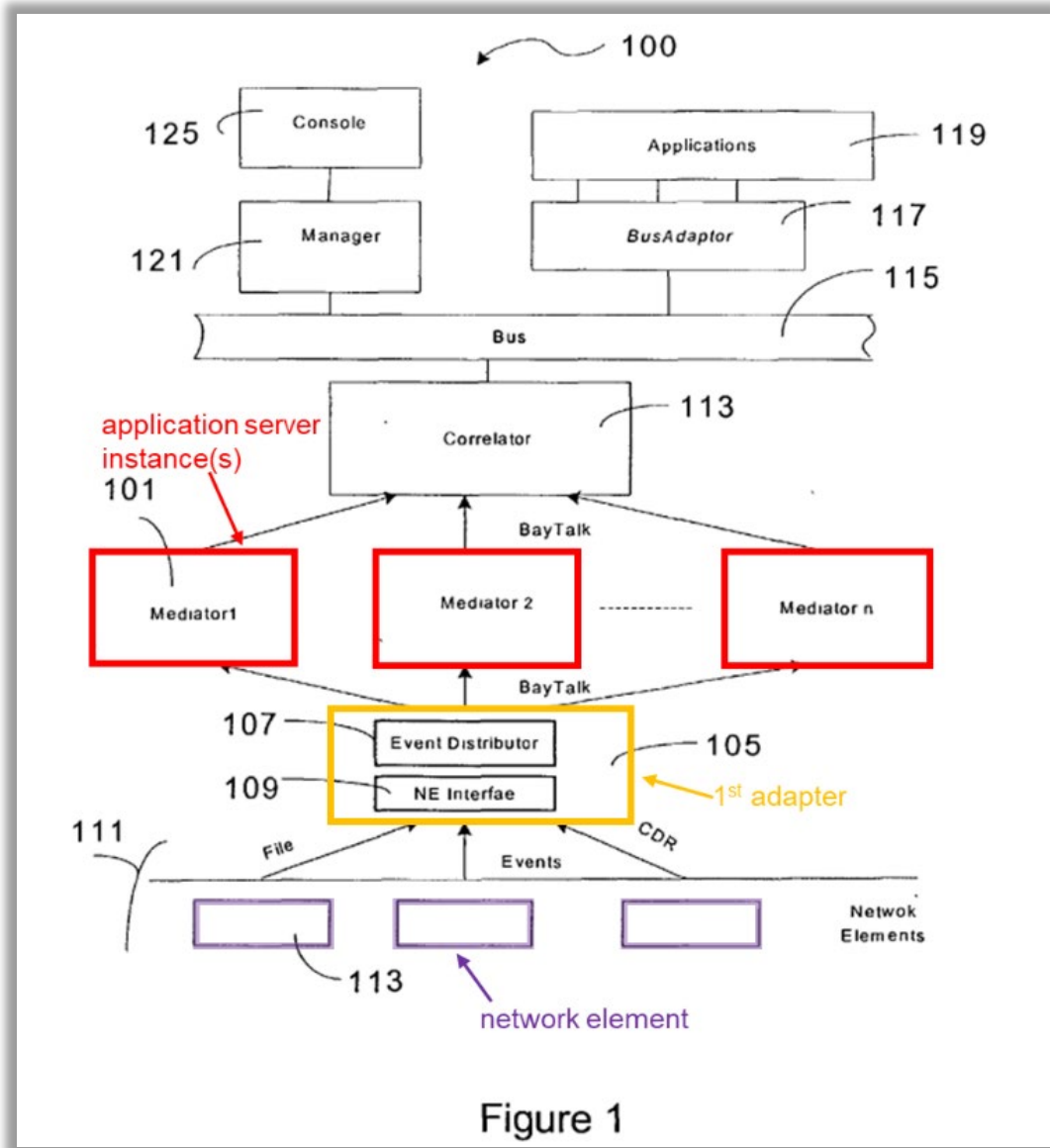


FIG. 4

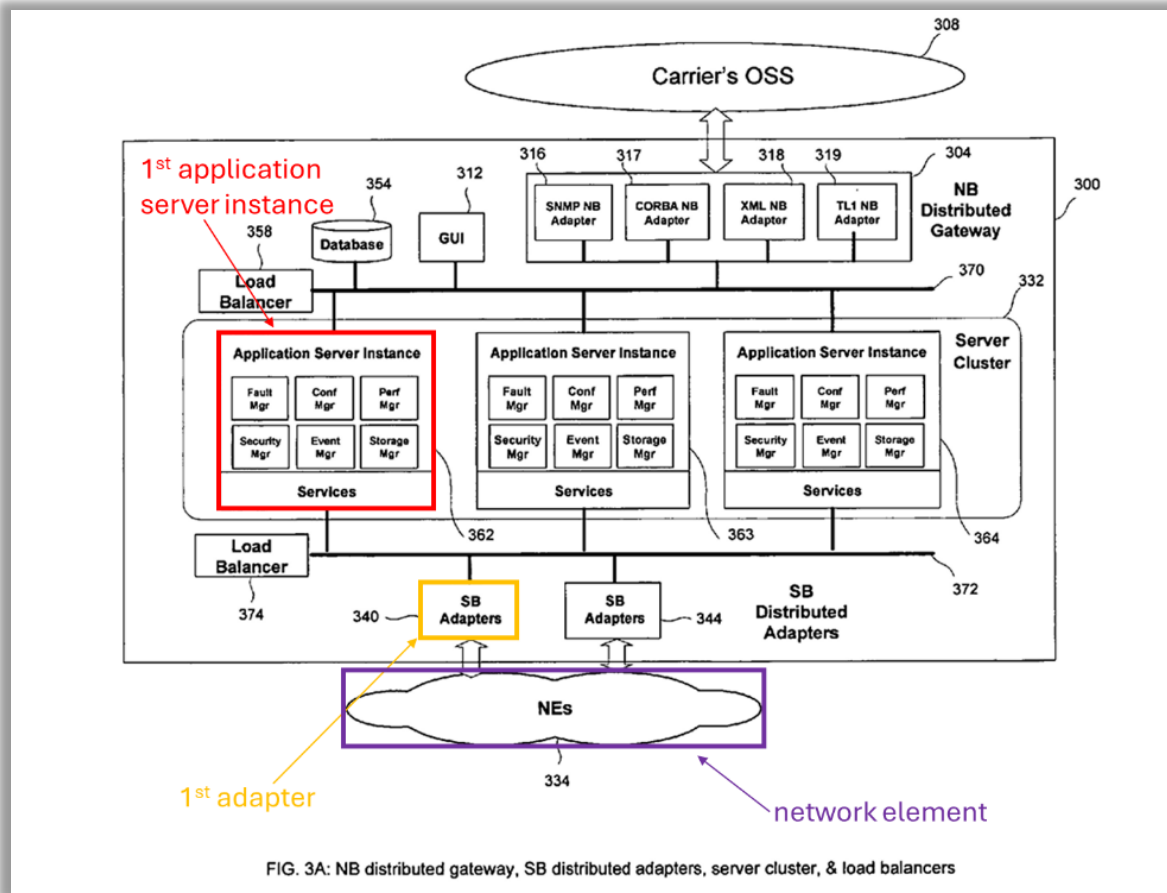
EX-1009 9:1-12, FIG. 4. The network management service applications included “configuration management” that tracked network configuration from remote locations, fault management, and performance management (optimization of network performance through data collection and analysis). EX-1009 9:1-12, FIG. 4. *Scholl* further acknowledged the simple network management protocol (SNMP) as a standard “network management protocol.” EX-1009 3:49-54. EX-1003 ¶56.

It was also known to use NMSs that monitored elements using the same architecture described in the '282 Patent. For example, U.S. 2004/0008717

(“Verma,” EX-1012) depicted distributed application servers called “mediators” (red) that received messages and events from NEs (purple):

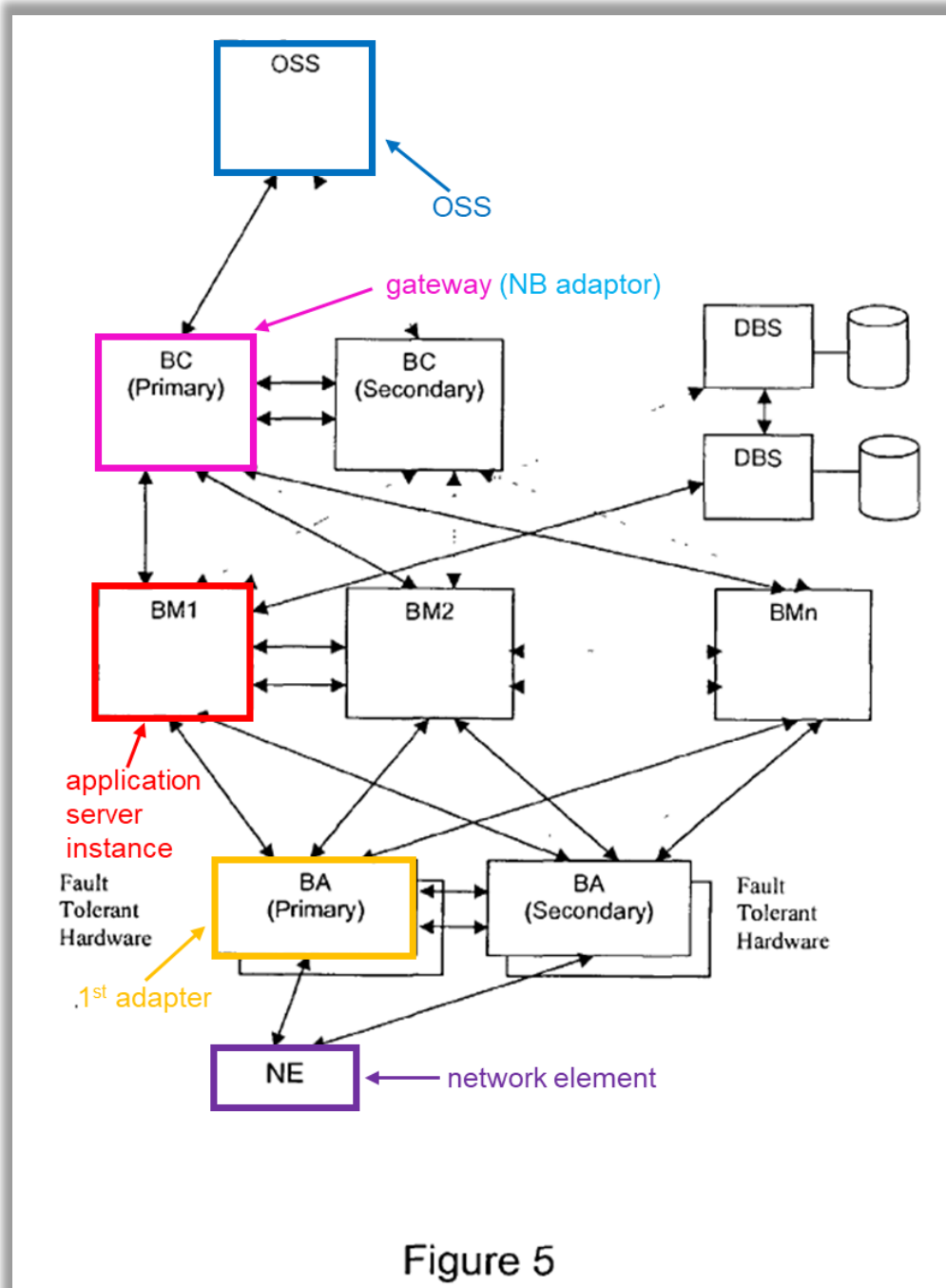


EX-1012 FIG. 1 (annotated); cf. EX-1001 FIG. 3A:



EX-1003 ¶57.

Each of the distributed application servers (red) monitored network activity or events through an adaptor (gold). The adaptor received event information, possible in different formats, from the NEs (purple), converted the information into a common protocol, and sent the converted event information to an application server instances. EX-1012 ¶30. The application server processed (e.g., converted) events are fed by application servers to different OSS after doing necessary correlation/filtering and formatting as per the OSS (blue) requirements via one of a plurality of correlators (BC) (pink):



EX-1012 ¶¶18, 32, FIG. 5 (annotated). *Verma* stated that having an adapter “interfac[ing] with the network elements and distribut[ing] the events to one or more mediator[s],” (distributed applications servers) “support[ed] higher traffic,

“ma[de] the system scalable,” and improved reliability in case of server failures.

EX-1012 ¶¶17, 31-79. In other words, it was known to add more application servers to handle more traffic and further improve fault tolerance. *Id.*; EX-1003 ¶58.

2. Adapters

Adapters were well known by December 2003. For example, *Verma* stated, “[t]he adaptor receives events, which may be in different formats, from the network and converts them into a common protocol” for the “mediator systems.” EX-1012 ¶30. Indeed, *Verma* emphasized the use of the common protocol for adapter-mediator communications, stating that it was preferable. EX-1012 ¶77. The adapters also formed part of a “scalable, fault tolerant” system. EX-1012 ¶17; EX-1003 ¶59.

As another example, U.S. 6,260,062 (“*Davis*,” EX-1013) disclosed adapters that “translated” network management messages and network element-dependent protocol messages “into a common element-independent message protocol, as known to one of skill in the art.” EX-1013 13:39-43, 15:19-22. *Davis* disclosed that its adapters supported the management of “very large, heterogenous telecommunications networks and support[ed] rapid, low-cost integration of new and different network element types having a variety of protocols and a variety of manufacturers.” EX-1013 5:11-16; 13:55-60.

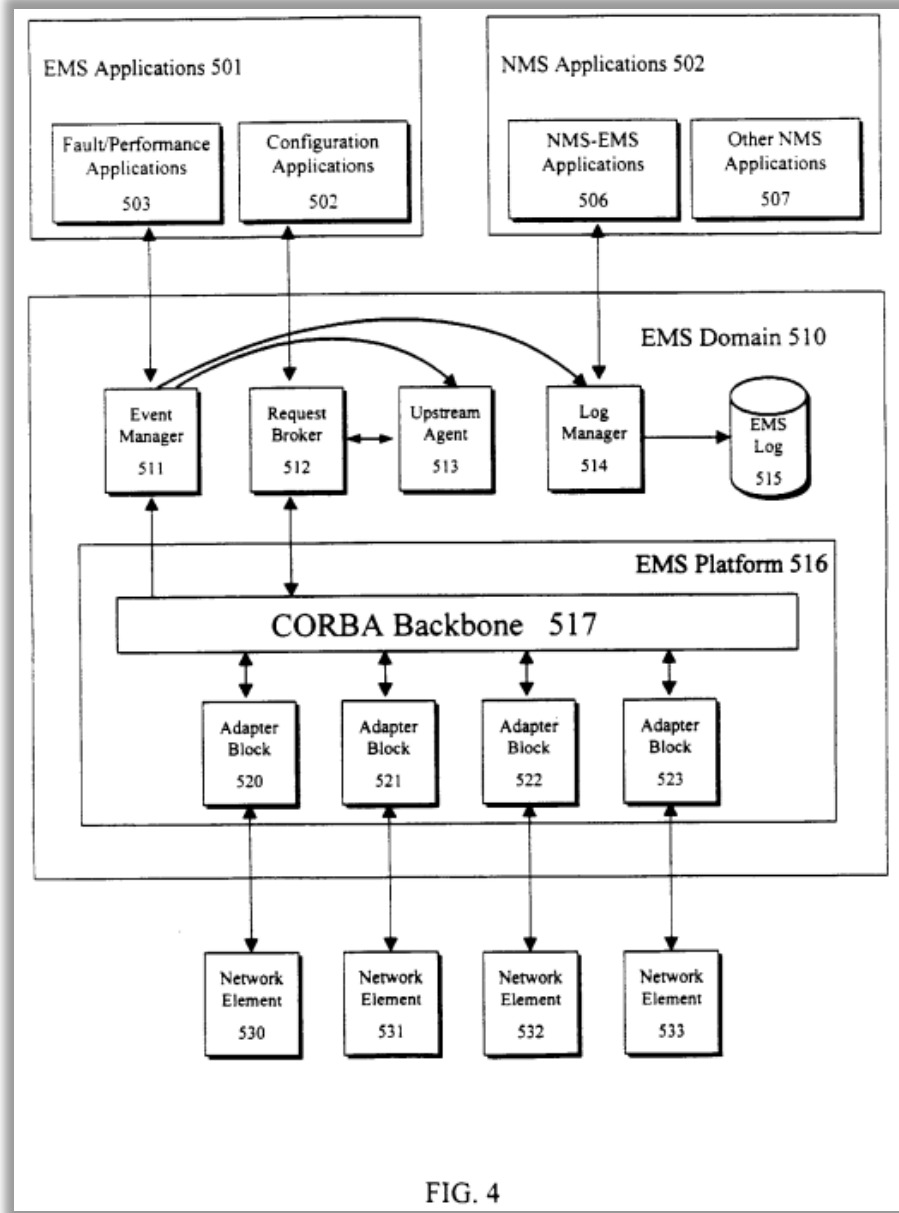


FIG. 4

EX-1013 FIG. 4; EX-1003 ¶60.

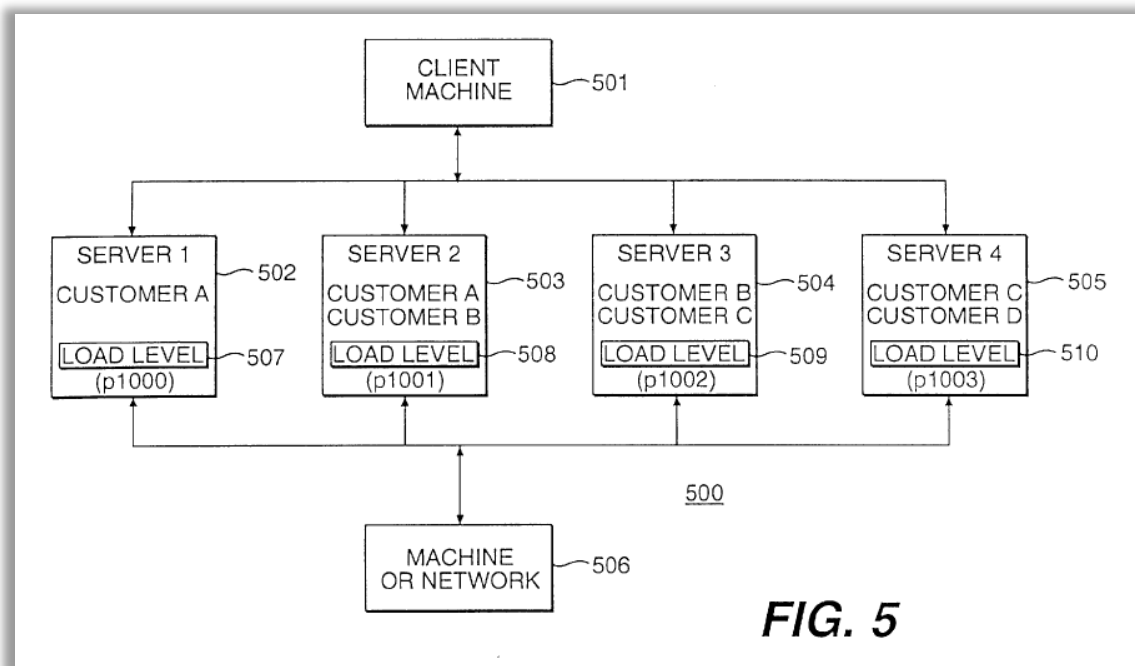
3. Load Balancing

By December 2003, load balancing, including, among distributed application servers, was well known. For example, in *Verma's* NMS, the adapter forwarded the events to the distributed mediators (application servers) “evenly” or “based on

load.” EX-1012 ¶¶31, 33 (load balanced mediators), 42 (adapters load balanced).

Verma explained that **distributing the load evenly to the application servers** “**save[d] costs**” and **avoided congestion**. EX-1012 ¶31, 42. Thus, load balancing and its benefits were known in the context of NMS application servers. EX-1003 ¶61.

As another example, U.S. 6,470,394 (“*Bamforth*,” EX-1011), expressly described that load balancing of distributed servers was important. *Bamforth* depicted a multi-application server environment:

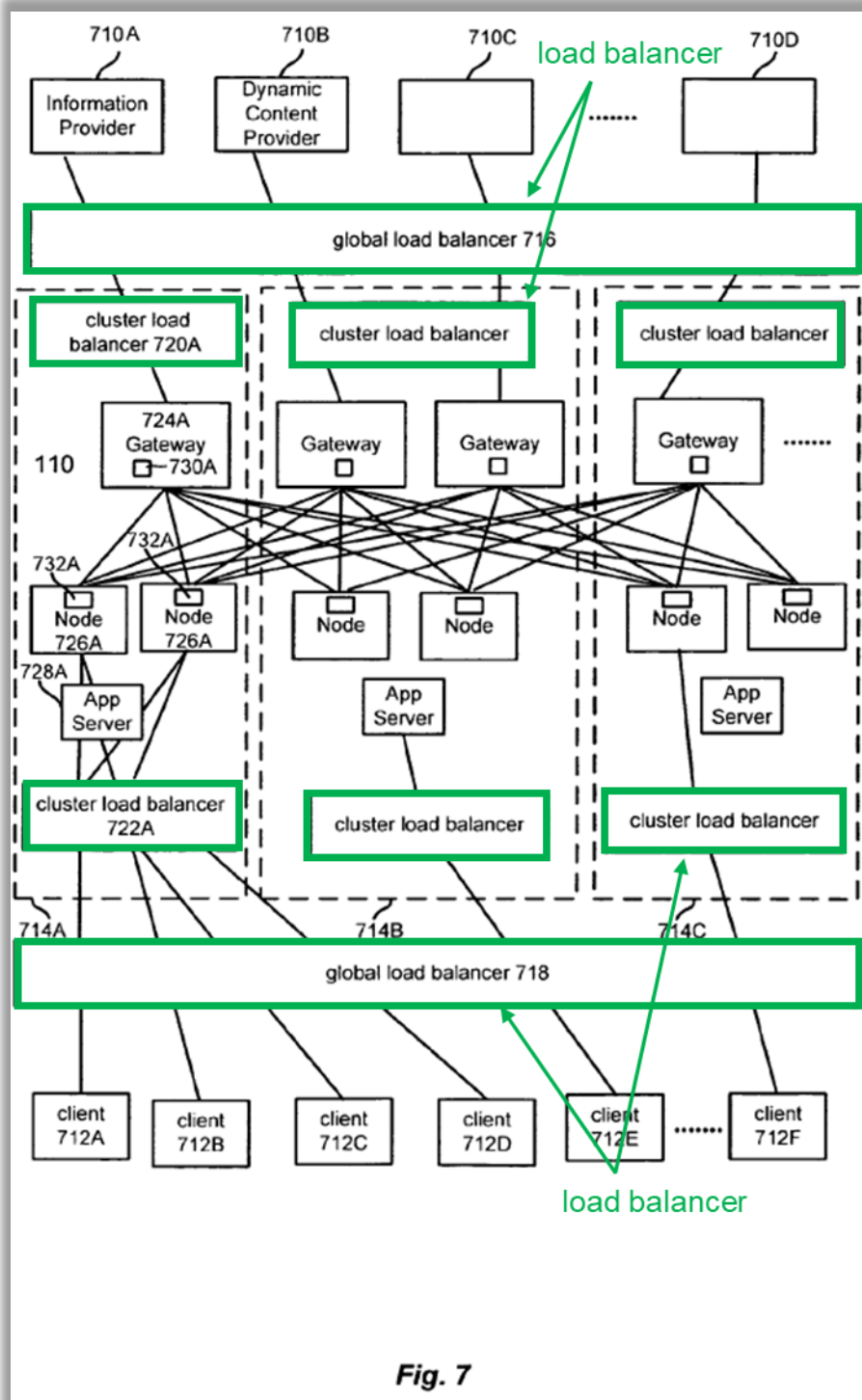


EX-1011 FIG. 5. *Bamforth* stated that “[b]alancing the customer load among the servers is important, for example, to maintain service to the customers and

avoid downtime.” EX-1011 8:15-20.² *Bamforth* further disclosed that “the **load balancing** achieved by [its] servers ... **may be used in any applicable computer network for any applicable processing.**” EX-1011 9:1-4; EX-1003 ¶62.

Moreover, the mechanisms for load balancing were well known, amounting to routine skill in the art. For example, U.S. 7,043,525 (“*Tuttle*,” EX-1010) disclosed that load balancers (green) for application server clusters were well known. EX-1010 15:16-16:21. *Tuttle* disclosed a “cluster load balancer”:

² Unless stated otherwise, all emphasis in this Petition is added.

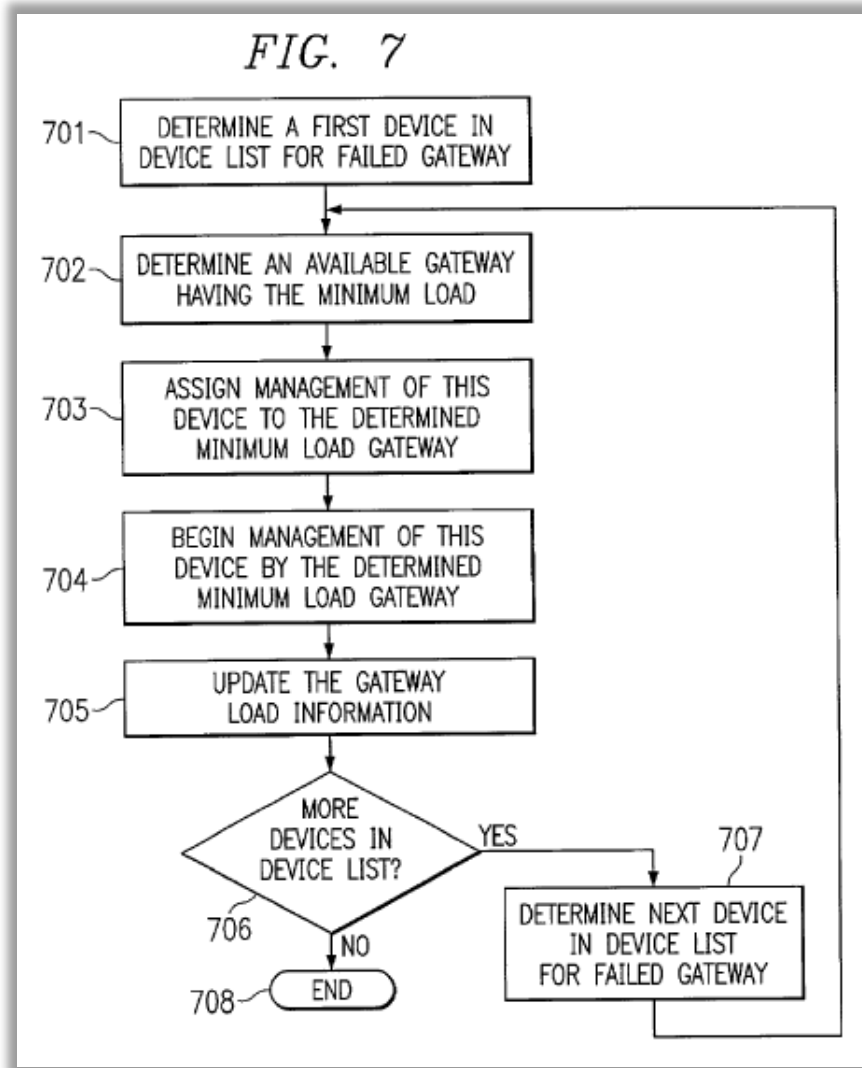


EX-1010 FIG. 7 (annotated). *Tuttle* explained that load balancers 716, 718 were designed to ensure that the load was distributed among the clusters 714A-714C. For example, the load may be distributed evenly among the clusters 714A-714C or

a more powerful cluster may be distributed a majority of the load. EX-1010 15:35-40, 16:4-11. *Tuttle* further disclosed cluster load balancers 720A, 722A which distributed messages once received within the cluster 714. EX-1010 15:44-53; EX-1003 ¶63.

As another example, WO 00/08823 (“*Keene*,” EX-1006) stated that load balancing across a multitude of servers lessens the work required by a single server and “**effectively speeds the response of the system.**” EX-1006 5:8-20; EX-1003 ¶64.

As further example, *Secer* disclosed “various types of **load balancing algorithms** may be utilized” in an NMS. EX-1004 17:1-3. One of these ways *Secer* disclosed load balancing was to find “an available **gateway having the minimum load is determined,**” and then assign work to that gateway:



EX-1004 FIG. 7, 17:9-18; EX-1003 ¶65.

Thus, a POSITA was well aware of various ways of implementing load balancing for application servers, including NMS application servers. EX-1003 ¶66.

4. Failover and Recovery

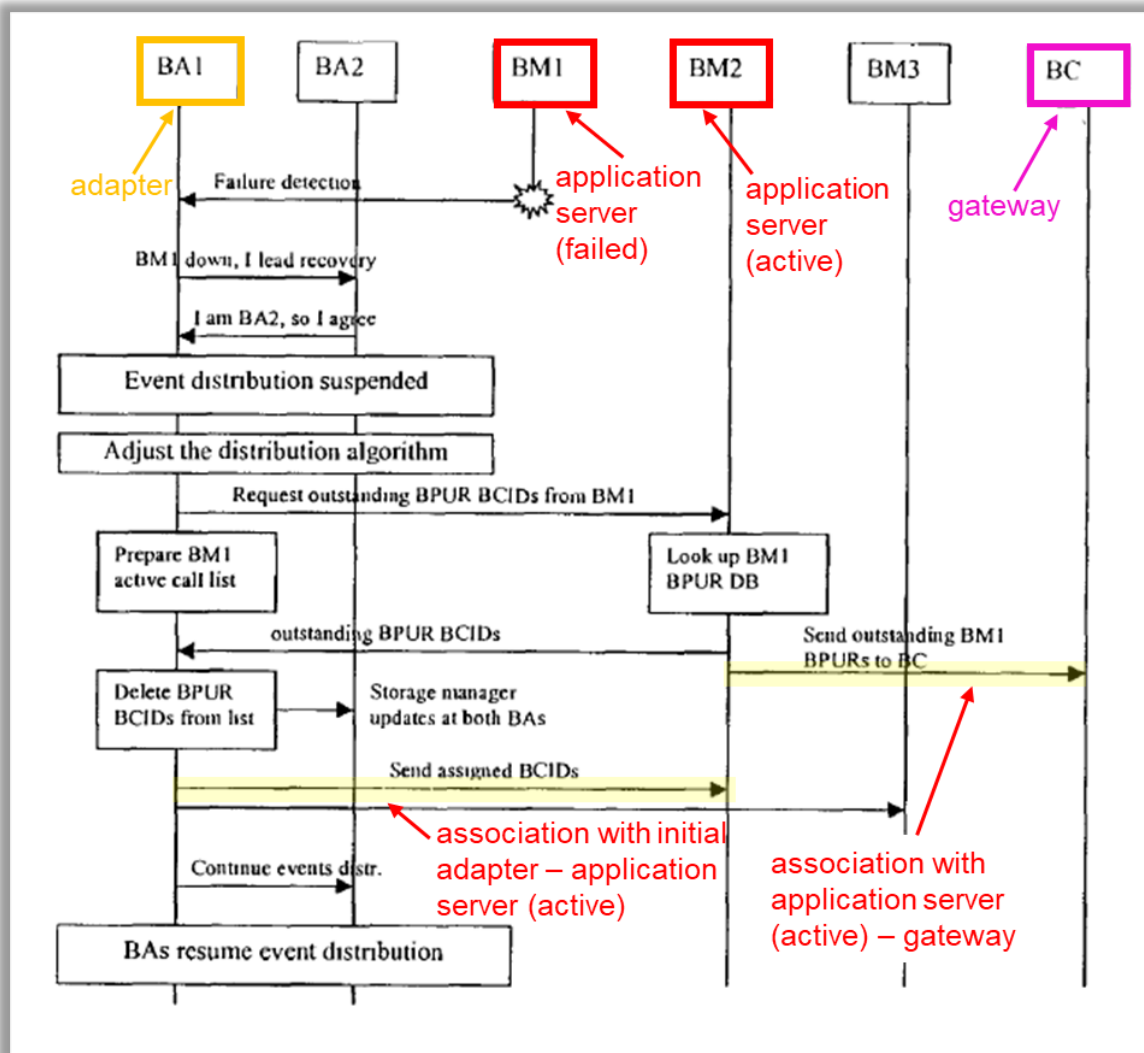
Failover and recovery techniques were well known by December 2003. EX-1003 ¶67. For example, *Verma* disclosed a fault tolerant NMS. EX-1012 Abstract.

Verma expressly disclosed that, if an application server (*Verma*'s mediator) failed, *Verma*'s adaptor redistributed event information, collected from the NEs, from the failed application server to another application server. EX-1012 ¶31; EX-1003 ¶67.

Unlike the '282 Patent, *Verma* disclosed at least one detailed example of detecting failure and performing recovery of an NMS application server, demonstrating the knowledge and skill in the art at the time of the purported invention. *Id.* at ¶81. For example, *Verma* disclosed that an adaptor could detect the failure based on loss of "heartbeat signals" transmitted from the application server. *Id.* at ¶81. The adapter detecting the failed application server would then lead the recovery process. *Id.* at ¶81. The adapter suspended the event distribution activity and adjusted the distribution algorithm so as not to send events to the failed application server (*i.e.*, load balanced). *Id.* at ¶81. The adapter prepared a list of events which were active at the time the application server failed and divided this list, and sent sending the events to other application servers based on the current load of the application servers. *Id.* at ¶81. This ensured that the events, which were being processed by the failed application server, were distributed to active mediators (application servers) based on their current load. *Id.* at ¶81. Thus, *Verma* disclosed an exemplary system having improved fault tolerance and

reliability in case of application server failures. EX-1012 ¶¶17, 31, 79, 81; EX-1003 ¶68.

Verma disclosed its failure recovery process in FIG. 6:



EX-1012 FIG. 6 (annotated). During the recovery process, an association between the adapter (“BA1”), which processed NE events and was previously associated with the failed application server (“BM1”), was established with the active application server (“BM2”), and the association with the gateway (BC) was

established with the active application server (“BM2”). EX-1012 ¶81, FIG. 6; EX-1003 ¶69.

A POSITA understood that establishing associations with the adapter and gateway that were associated with the failed application server, as exemplified by *Verma*, simplified the recovery process compared to establishing new associations with a different adapter and gateway. EX-1012 ¶81, FIG. 6; EX-1003 ¶70. The adapter and gateway that initially serviced the NEs used one protocol while other adapters and gateways used different protocols. EX-1012 ¶¶31-32, 51-52. Instead of searching for a new adapter and gateway that used the same protocols, the same adapter and gateway would be associated with the new application server. EX-1012 ¶81, FIG. 6; EX-1003 ¶70. This would save time and computing resources by eliminating the need to search for a new adapter or gateway and forego the associations between the new adapter and the NE, and the new gateway and OSS. EX-1003 ¶70. Additionally, the geographic location between the initial adapter and gateway likely would have been selected based on having a minimal distance to the NEs and OSS, respectively, in order to reduce network traffic and latency within the network, and thus a POSITA understood it would be beneficial to use the initial adapter and gateway with a replacement application server to maintain these benefits. EX-1004 17:67-18:11; EX-1012 ¶81; EX-1003 ¶70.

Likewise, *Dinker* confirmed that failover techniques used in application server clusters were well known and desirable:

In computer systems, “failover” refers to a backup operational mode in which the functions of a system component (such as a processor, server, network, or database, for example) are assumed by secondary system components when the primary component becomes unavailable through either failure or scheduled down time. It would be **desirable to provide** various types of **failover mechanisms to increase the fault-tolerance of an application server cluster**.

EX-1005 ¶33, Abstract, ¶¶ 2, 34, 57; EX-1003 ¶71.

Keene (EX-1006) is yet another example showing failover systems and techniques were well known: “When a system fails, the **remaining available systems take over the failed system’s load**.” EX-1006 1:17-21. *Keene* further confirmed the widely-understood, benefit of fault tolerance: “The ability of a peer group to allocate work amongst available servers, and **then reallocate work if a particular server should become unavailable, provides fault tolerance**.” EX-1006 5:19-22; EX-1003 ¶72.

U.S. 2003/0028654 (“*Abjanic*,” EX-1007) is yet another example that further confirmed how well-known failover systems, including for servers, and their obvious benefits were to a POSITA:

[A] program may detect the **failure of one or more servers...and then...account for these changes in the network (e.g., **redirect certain messages...from servers which have failed to the available servers**)**.

EX-1007 ¶47; EX-1003 ¶73.

V. IDENTIFICATION OF HOW THE CLAIMS ARE UNPATENTABLE

A. Ground 1: Claims 1-22 Are Obvious over *Secer* in View of *Dinker*

1. *Secer* (EX-1004)

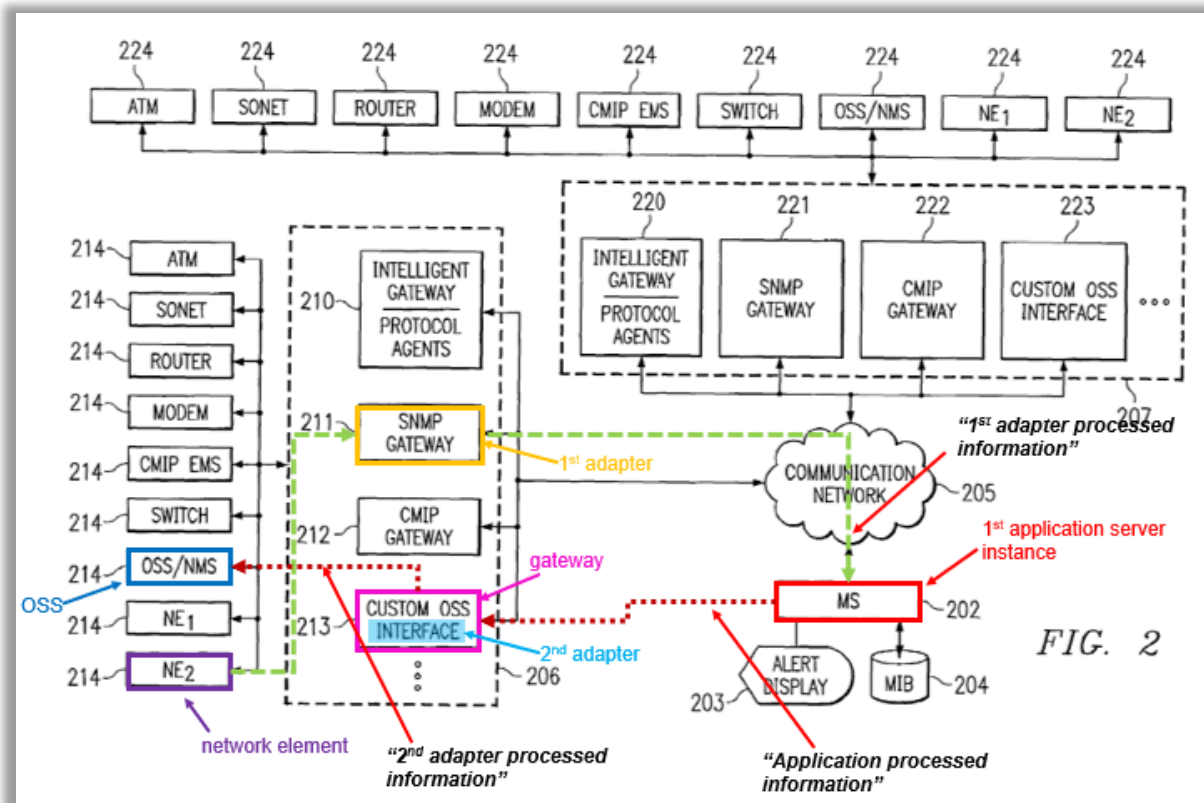
U.S. Patent 7,209,968 to *Secer* (“*Secer*,” EX-1004) is directed to a “System and Method for Recovering Management of Network Element(s) Responsive to Failure of a Distributed Gateway.” EX-1004 (10), (45), (75), (54); EX-1003 ¶74. *Secer* issued April 24, 2007 and was filed on May 29, 2001 qualifying as prior art under pre-AIA 35 U.S.C. §102(e). EX-1004 (22). *Secer* was not before the Examiner during prosecution of the ’282 Patent. EX-1002.

Secer disclosed a system and method for efficient recovery of the management of NEs responsive to the failure of a distributed “gateway” coupled between the NEs and a central management system. EX-1004 Title, 1:25-31, Abstract. EX-1003 ¶75.

Secer acknowledged that “legacy management systems,” including NMS and OSSs, managed networks and NEs and “commonly recognize[d] faults (or traps)” from the NEs and/or polled NEs. EX-1004 1:52-60, 2:15-35. *Secer* explained that the prior art management systems faced challenges recovering from

failures, resulting in the loss of “many messages (or events)” from NEs, stating that prior NMS “often fail[ed] to efficiently resolve the failure.” EX-1004 7:30-47; EX-1003 ¶76.

Secer illustrated an NMS for addressing these issues:



EX-1004 FIG. 2 (annotated); see also FIGs. 3-6 (providing additional detail).

Specifically, *Secer* disclosed a central management system (MS) 202 (*first application server instance*)³ (red) comprising a server for network management that received data from *network elements* 214 (purple) collected by “gateway group

³ This Petition uses italics for language from the Challenged Claims.

206” containing “gateways” 210-213. EX-1004, 9:17-18. *Secer* further disclosed that each gateway 210-213 (*first adapter*) (gold) could facilitate communication (*i.e.*, processing, filtering, translating, etc. communications and data) between central MS 202 and network elements 214 using different communication protocols. EX-1004 9:17-16. For example, the gateway 211 processed and converted communications from SNMP-based devices. EX-1004 1:64-2:8, 8:60-9:18, FIG. 2. These communications include messaging, events, and polling information received at the gateway. EX-1004, 9:13-26. Therefore, the gateways (each a *first adapter*) processed communications from network elements, converted the communications to the correct protocol, and sent the converted communication to central MS 202 for further processing. EX-1004 9:22-25; EX-1003 ¶¶77.

Secer disclosed that the MS (*application server instance*) used various software applications to process messages and polling information received from the gateways to pull “management behavior objects” that controlled the behavior of the gateways and NEs. EX-1004 9:26-10:11. For example, the “management behavior objects” described the operation(s) to be performed by a gateway(s) in response to various messages from the network elements. EX-1004 9:62-10:4; EX-1003 ¶¶78.

Secer described that the MS identified and pushed the management behavior object to the “appropriate gateways.” EX-1004 10:4-11. *Secer* also disclosed that gateways 210-213 included functionality to recognize and process different protocols. EX-1004 2:29-35; 9:3-16. *Secer* also disclosed that the gateways included functionality for fault management and performance management. EX-1004 2:62-3:17; 10:12-30; EX-1003 ¶79.

Secer disclosed that gateway 213 (pink) included the particular protocols for communicating with OSS 214 (blue) using a custom OSS interface gateway 213 (light blue). EX-1004 9:3-16. When a management behavior object identified, e.g., OSS 214, the management behavior object(s) for OSS 214 would be sent to gateway 213 (pink) for processing using the custom OSS interface protocol (second adapter) (light blue) to be sent from custom OSS interface gateway 213 to OSS 214 (blue). EX-1003 ¶80.

Secer also disclosed enabling efficient recovery of the management of NEs responsive to a failure of a managing gateway. EX-1004 4:7-21; 10:12-30. Specifically, *Secer* disclosed monitoring the operation of distributed gateways that managed the NEs. EX-1004 4:7-21, 10:31-52, FIG 3, FIG 4 (monitor 401, 402), FIG 6 (monitor 603, 604). Through such monitoring, failure of one of the distributed gateways may be efficiently detected, and management of the affected NEs was efficiently recovered by assigning management responsibility to another

distributed gateway. EX-1004 4:7-21. While the details of this failover mechanism were expressly described with respect to gateways, *Secer* stated that “the detection and recovery techniques described herein may be utilized within any client/server environment and may be applied to devices other than gateways.” EX-1004 18:12-15. Thus, a POSITA understood that *Secer* would be readily improved by extending its failover concepts to its other components, including its MS. EX-1004 18:12-28; EX-1003 ¶81.

2. *Dinker* (EX-1005)

U.S. 2003/0177411 to *Dinker* et al. (“*Dinker*,” EX-1005) is titled “A System and Method for Enabling Failover for an Application Server Cluster.” EX-1005 (10), (12), (54), Title. *Dinker* was filed on March 12, 2002 and published on September 18, 2003 (EX-1005 (43) (22)), and qualifies as prior art under at least pre-AIA 35 U.S.C. §§ 102 (a) and (e). *Dinker* was not before the Examiner during prosecution of the ’282 Patent. EX-1002.

Dinker was directed to managing application server clusters, including “failover for an application server cluster.” EX-1005 Title, ¶2. For example, *Dinker*’s application server cluster included application servers 108A, 108B (each an *application server instance*) (red) and a broker/web server 104 that load balanced its client requests among the servers of the cluster (*load balancer*) (green):

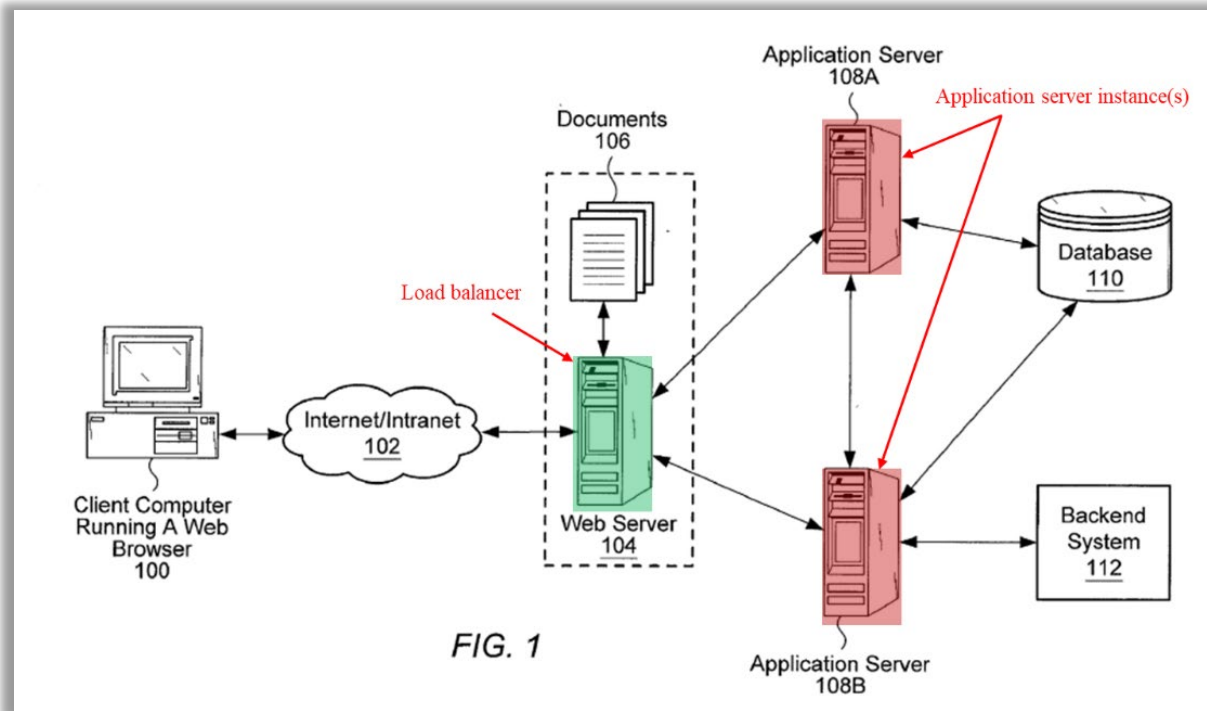


FIG. 1

EX-1005 FIG. 1 (annotated). *Dinker's* broker/web server 104 (*load balancer, load balancing component*) was coupled to a plurality of application servers and selected an application server (*application server instance*) from the server cluster using “load balancing techniques.” EX-1005 ¶49; EX-1003 ¶83.

Dinker further disclosed various failover procedures for the server clusters, including a “backup operational mode in which the functions of a system component (such as a processor, **server**, network, or database, for example), are assumed by secondary system components when the primary component becomes unavailable through either failure or scheduled down time.” EX-1005 ¶33. Thus, *Dinker* disclosed well-known application server clustering, load balancing and

failover and their well-known benefits of improved scalability, fault tolerance and reliability. §IV.F.4; EX-1003 ¶84.

3. Motivation to Combine *Secer* and *Dinker*

Secer disclosed a single application server (MS 202) in its distributed NMS, with load balanced and failover for its gateways. A POSITA would have found it obvious to extend *Secer*'s load balancing and failover concepts to its application server following *Dinker*'s teachings of application server clustering that used load balancing and failover, for several reasons. EX-1003 ¶¶85-97.

First, *Secer* and *Dinker* are analogous art. *Secer* is in the same field of endeavor as the '282 Patent because it is directed to managing NEs, including for fault detection and performance management of the same, using an NMS having distributed network components to increase the efficiency, including during recovery from failures network-management components. EX-1004 1:25-35. *Secer* is reasonably pertinent to the problem addressed by the '282 Patent's claimed invention because *Secer*'s solution detailed distributing network-management components (e.g., adapters) and implementing failover techniques for the same. EX-1001 Abstract, 1:5-7; 2:12-13; EX-1004 Title, Abstract; EX-1003 ¶86.

Dinker is in the same field of endeavor as the '282 Patent because it relates to “a system and method for enabling failover for an application server cluster.”

EX-1005 ¶¶2; EX-1001 2:52-54, 3:65-66. *Dinker* is reasonably pertinent to the problem addressed by the '282 Patent's claimed invention because *Dinker*'s application-server-cluster solution "provide[d] various types of failover mechanisms to increase the fault-tolerance of an application server cluster," enabling a "robust" network. EX-1005 ¶33; EX-1001 1:22-25; EX-1003 ¶87.

Moreover, a POSITA readily recognized that *Secer*'s load balancing and failover techniques (discussed in detail for its gateways) would also be applicable to its MS 202, as suggested by *Secer*, and in view of *Dinker*'s teachings. *Secer*'s NMS managed its network elements in a client/server environment similar to the client/server environment of *Dinker*. As described above, *Secer* used distributed gateways in an NMS and provided for assigning network element management responsibility to another gateway when one gateway failed. EX-1004 4:7-21. Indeed, *Secer* stated that it was desirable to "efficiently recover[]" the management of network elements "to minimize the time in which such network element(s) are without management." EX-1004 1:32-35. *Secer* expressly disclosed that its failover teachings extended to other network management components, stating that its "detection and recovery techniques...may be utilized within **any client/server environment** and may be applied to devices **other than gateways**." EX-1004 18:12-22. A POSITA would have realized that such a client/server environment existed between *Secer*'s gateways (both the claimed *first* and *second adapters* and

gateway) and MS (*application server instance*) through the exchange of management behavior objects and, in any case, that *Secer's* MS was one such “other” device, given the well-known technique and benefits of clustering application servers, such as taught by *Dinker*. EX-1003 ¶88.

Second, *Dinker* suggested using a load-balanced, fault-tolerant application server cluster in a wide variety of applications. For example, *Dinker* stated that the application server clusters of FIG. 1, which shows client computer communicating with a server (i.e., a client/server environment), “may be utilized in **any of various types of systems.**” EX-1005 ¶45. A POSITA would have readily recognized an NMS, like that of *Secer*, was a system to which *Dinker's* teachings applied. EX-1003 ¶89. Indeed, a POSITA was familiar with load-balanced, fault-tolerant application server clusters, both generally and specifically in the context of NMS. §§IV.F.1, IV.F.3-4 (citing, *inter alia*, EX-1011 8:15-20; EX-1010 15:35-40, 16:4-11; EX-1012). Moreover, *Secer* suggested that failover systems could be applied to other components and in a client/server environment, like those disclosed in *Dinker*. EX-1004 18:12-22. Therefore, a POSITA would have found it obvious to use a load-balanced, fault-tolerant application server cluster as taught and suggested by *Dinker* for *Secer's* NMS. EX-1003 ¶89.

Third, a POSITA would have been motivated to modify *Secer's* NMS system to include a load-balanced, fault-tolerant application server cluster as taught

by *Dinker* for *Dinker's* expressly stated benefits of improved scalability. EX-1003 ¶90. For example, *Dinker* disclosed that “**application servers may be added** to a cluster in order **to scale up** the available processing power **by distributing work.**” EX-1005 ¶31. Indeed, the benefit of scale brought by clustering servers was well known to the POSITA. §§IV.F.1, IV.F.2-3 (citing EX-1012 (describing adding more servers dynamically to handle more traffic); EX-1006 7:16-19 (“Utilizing a multitude of servers to process work effectively lessens the work required by a single server and effectively **speeds the response of the system.**”)); EX-1003 ¶90. *Dinker* explained that the increased processing power, provided by the additional application servers in the application server cluster, allowed the network management system to manage more network elements and process more event information to ensure the proper functioning of the network. EX-1005 ¶31. Furthermore, application server clustering provided additional benefits including increased memory capacity and the ability to provide more application services which enabled more flexibility to the customers and user of the network. EX-1005 ¶¶21, 30-31 EX-1003 ¶90.

Fourth, modifying the teaching of *Secer's* NMS system to include a load-balanced, fault-tolerant application server cluster provided for improved performance and a more efficient use of resources as expressly taught by *Dinker*. EX-1005 ¶¶31, 49. For example, *Dinker* stated that “[a]pplication server clustering

may also facilitate application performance.” EX-1005 ¶¶31. Additionally, like scalability, this improved performance and a more efficient use of resources provided for by a load-balanced, fault-tolerant application server cluster was well understood by the POSITA outside of *Dinker*. §IV.F.3 (citing EX-1012 ¶31; EX-1011 8:15-20 (“**[b]alancing the customer load among the servers is important, for example, to maintain service to the customers and avoid downtime.**”); EX-1006 5:8-20 (load balancing servers lessens the work required by a single server and “**effectively speeds the response of the system.**”); EX-1012 ¶31, 42 (distributing the load evenly among application servers “save[d] costs” and avoided congestion). In addition, *Secer* also taught the benefits of load balancing across its gateways, discussing various load balancing algorithms, as well as failover. EX-1004 4:39-49; 16:61-17:8. Thus, a POSITA would readily apply the same techniques for the same benefits — taught in *Secer*, known in the art, and exemplified by *Dinker* — to MS 202 implemented as a server cluster following *Dinker*. EX-1003 ¶91.

Moreover, the ’282 Patent acknowledges that load balancing across multiple application servers was well known as it provides no details for implementing load balancing, merely stating that “**any** load balancing algorithm” may be used and work can be distributed “in **any desired fashion.**” EX-1001 8:63-64, 9:10-11, 9:54-58. As the ’282 Patent states, “technical material that is known in the

technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.” EX-1001 2:8-11; EX-1003 ¶92.

Fifth, a POSITA would have been motivated to modify the teaching of *Secer*’s NMS to include a load-balanced, fault-tolerant application server cluster as taught by *Dinker* to increase fault-tolerance. EX-1005 ¶33. Indeed, *Dinker* stated that it was “**desirable** to provide various types of failover mechanisms to increase the fault-tolerance of an application server cluster.” EX-1005 ¶33. By having a plurality of application servers, a back-up server would fulfill the role of a first server in the event the first server failed or was taken offline, providing the benefit of redundancy. EX-1005 ¶¶33, 54, 57. Like load balancing and the use of application server clusters, the POSITA was well aware of the benefits of failover systems, including in NMS. §IV.F.4 (citing EX-1012 ¶¶17, 31 79; EX-1006 5:19-22); EX-1003 ¶93.

Moreover, both *Secer* and *Dinker* described establishing associations as part of failover procedures. For example, *Secer* described associating a new gateway with the network elements that were previously managed by a failed gateway. EX-1004 10:24-30, 10:45-52 (reassigning network element management to a new gateway), 17:34-18:11 (reassigning based on load and geographic location between the original NE and replacement gateway), 18:12-28. *Dinker* described a “primary” application server that serviced both (1) client request originating outside of the

server cluster and (2) other application servers inside the cluster, to “manage and provide processing information” necessary to operate, such as session information, for client requests. EX-1005 ¶¶54-55. This information necessary for this role was backed-up in another application server. EX-1005 ¶57, e.g. Fig. 2 (showing shared/mirrored data among servers). When the “primary” server failed, the backup application server already contained the backed-up data and would be promoted to primary, allowing the server cluster to continue operations uninterrupted. EX-1005 ¶57. Such application server clustering was used so that in the case of “failure on one application server” “requests may be routed to and processed by other application servers in the cluster.” ¶30. This shifting of load was well known. §IV.F.4 (citing EX-1006 3:17-21 (discussing “[w]hen a system fails, the remaining available systems **take over the failed system’s load.**”); EX-1007 ¶47 (disclosing “**redirect[ing] certain messages ... from servers which have failed to the available servers**); EX-1012 ¶¶31, 34-35). Such teachings were also known to extend to other components like adapters, where adapters established a relationship with a secondary application server for the purpose of ensuring that the adapter redistributes the information about network element events. EX-1012 ¶¶31, 81; EX-1003 ¶94.

Sixth, a POSITA would have modified the teachings of *Secer* to include a load-balanced, fault-tolerant application server cluster as taught by *Dinker* because

such a modification required nothing more than merely combining known network components (*i.e.*, adding additional servers and/or server functionality to existing servers) according to known methods (*i.e.*, installing server software and/or server hardware). EX-1005 ¶31; EX-1003 ¶95.

Seventh, a POSITA would have a reasonable expectation of success because the combination involved nothing more than routine and ordinary skill, and such features were already implemented in the NMS context. EX-1003 ¶96. *Secer* stated that its failover mechanisms “may be utilized within any client/server environment and may be applied to devices other than gateways for managing network elements,” and a POSITA would have recognized that an NMS server was just such a client/server environment and/or other components to which failover would apply. EX-1004 18:12-15. Likewise, *Dinker* stated that its load-balanced, fault-tolerant application server cluster “may be utilized in **any of various types of systems**,” and a POSITA would have recognized an NMS, like that of *Secer*, as just such one type of system. EX-1005 ¶45. Indeed, the POSITA was well aware of load-balanced, fault-tolerant application server clusters, including in the context of NMS, as evidenced by the state of the art. EX-1005 ¶¶ 33; EX-1004 Abstract; §IV.F (citing EX-1011 FIG. 5 (illustrating a multi-server environment); EX-1006 7:19-20 (describing a cluster of available servers); EX-1010 15:35-40, 16:4-11 (describing a plurality of application server clusters); EX-1012 ¶¶31, 34-35

(describing a fault-tolerant, load-balanced NMS)). Thus, the use of server clusters was widely implemented and known to provide redundancy, additional capacity, and a variety of application services, including NMS. A POSITA also knew how to implement load-balanced, fault-tolerant application server clusters using nothing more than the routine skill. EX-1003 ¶¶96.

For the reasons described above, therefore, a POSITA would have been motivated to modify *Secer's* teachings of a network management system with *Dinker's* teachings of application server clustering, load balancing, and failover recovery. EX-1003 ¶¶97.

4. Detailed Application of *Secer* in Combination with *Dinker* to the Challenged Claims

Claim 1

[1pre] *A method, comprising:*

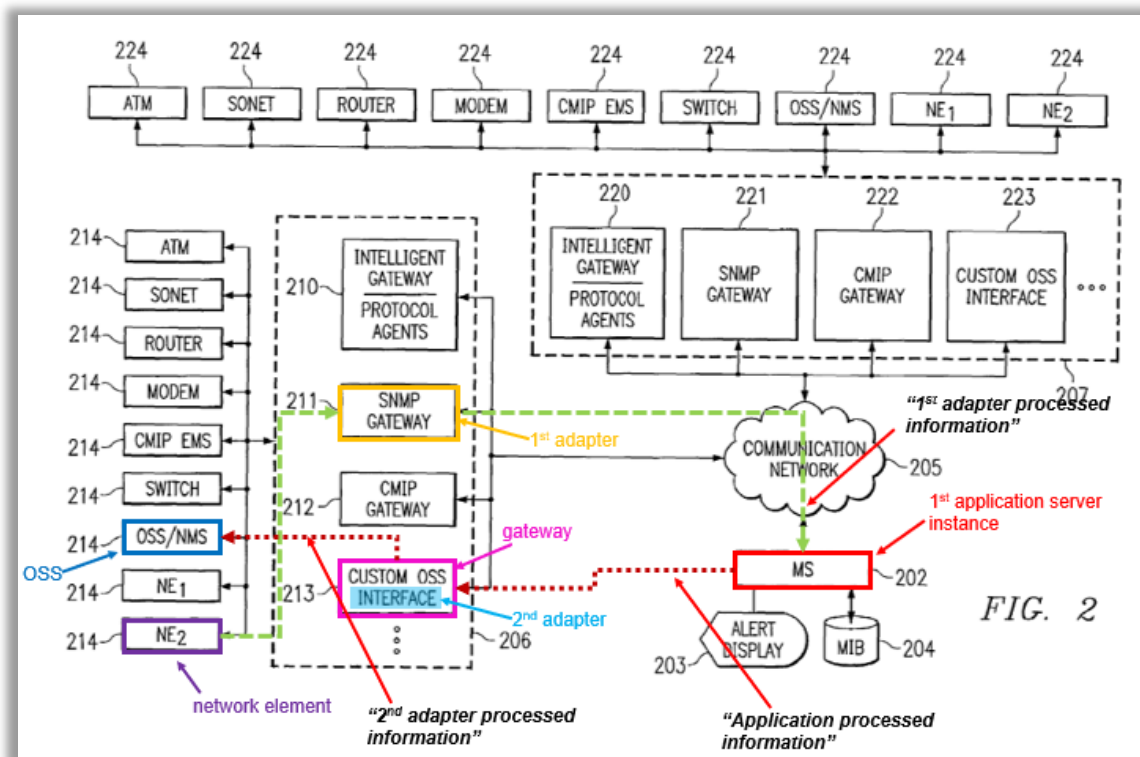
To the extent the preamble is limiting, *Secer* in view of *Dinker* rendered obvious the method comprising the limitations [1ai]-[1d] as discussed in detail below. EX-1003, ¶¶98-140.

[1ai] *receiving, at a first application server instance selected from a plurality of application server instances based on a load balancing process, first adapter processed information from a first adapter,*

Secer in view of *Dinker* rendered obvious this limitation. EX-1003 ¶¶99-112. *Secer* disclosed a central management system (MS) (*first application server instance*) and a gateway having adapter functionality (*first adapter*). MS received

processed network element data (*first adapter processed information*) from gateway (*receiving, at a first application service instance...first adapter processed information from a first adapter*). EX-1003 ¶¶99.

Secer disclosed its MS 202 (*first application server instance*) (red) and gateway having adapter functionality (*first adapter*) (gold) in, e.g., FIG. 2:



EX-1004 FIG. 2 (annotated); EX-1003 ¶¶100.

Secer disclosed that the network element data included information about the operation/performance of the network element(s), such as operational and health data including fault messages (traps), a network element's CPU utilization, network

element rebooting information, available storage capacity information, network element interface information, etc. EX-1004 2:12-35; 2:41-52; EX-1003 ¶101.

Secer disclosed that MS 202 (*first application server instance*) received the network element data that was first processed by a gateway (*first adapter*) according to a particular communication protocol (*first adapter processed information*). For example, *Secer* disclosed MS 202 (*first application server instance*) (red) comprised a server for network management that received data collected from the “*network elements 214*” (purple) at the gateway group 206. EX-1004 9:13-26, FIG. 2. *Secer*’s MS 202 is an application server of a network management system because it “manag[es] communication networks and network elements,” including receiving communications from its gateways (adapters) and directing the actions of gateways and network elements using “management behavior” objects. EX-1004 1:52-2:1 (“MSs,’ encompass ... NMSs”); 9:17-10:11, *see also* [1b], *infra*. The *network elements* managed by MS 202 included “routers, switches, computer equipment, etcetera” that communicated using “different protocols.” EX-1004 1:57-64; EX-1003 ¶102.

Next, *Secer* disclosed that gateways 210, 211, 212, 213 (*first adapter*) (gold) facilitated communication between the MS 202 and network elements 214. EX-1004 9:13-18. *Secer* explained that “[e]ach of the distributed gateways may, for example, be any suitable processor-based device operable to manage (e.g., receive

unsolicited messages and/or poll) its respective network elements.” EX-1004, 9:13-26. *Secer* further disclosed that its gateways perform typical monitoring of NEs to detect faults, including polling NEs to request information about the operation/performance of the NE (all network element data). EX-1004 2:39-52. For example, a gateway periodically polled a NE to determine whether the NE is operational. EX-1004 2:52-54. A failure of a NE to respond to a poll is indicative of a problem (e.g., failure) with the NE. EX-1004 2:52-58. Gateways also periodically polled NEs to determine the NE workload, available memory capacity, etc. EX-1004 2:49-61; EX-1003 ¶103.

Secer further disclosed that the gateways monitored NE having particular communication protocols, including as examples SNMP gateway 211, CMIP gateway 212, and custom OSS interface gateway 213, which monitor various network element types 214 having various protocols, such as ATM, SONET, routers, modems, CMIP EMSs, switches, OSS/NMSs, as well as various other NEs local to group 206. EX-1004 8:57-9:3. *Secer*'s distributed gateways included functionality to process information received from network elements 214 (network element data) to include filtering and translating from “one plurality of different protocols to another plurality of different protocols.” EX-1004 9:21-26, cl. 23. *Secer*'s disclosure of the functionality of its gateways is consistent with the '282 Patent's disclosure that adapters process information to facilitate communication

between the NMS server and NEs using different protocols. EX-1001 2:64-67.

Thus, Secer's gateways including adapter functionality and were *adapters*.⁴ EX-1003 ¶104.

Thus, *Secer* disclosed MS 202 (*first application server instance*) received processed network element data (*first adapter processed information*) from a gateway (*first adapter*). EX-1003 ¶105.

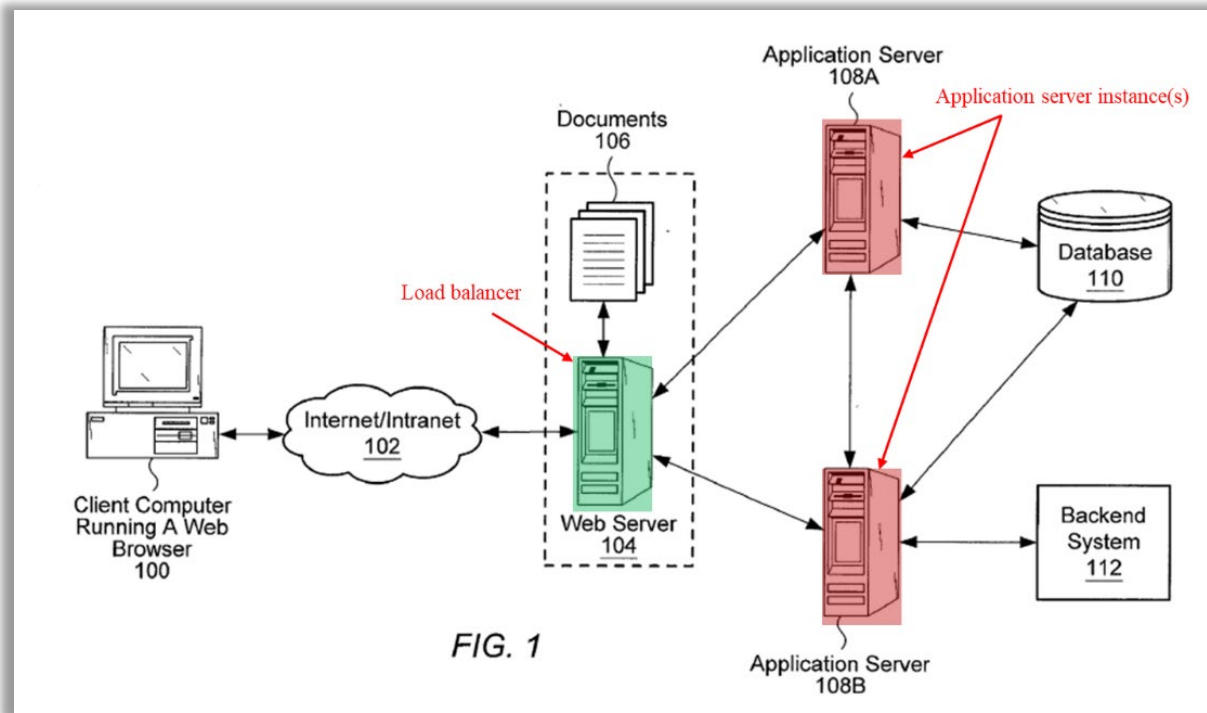
Although, *Secer* did not expressly disclose that its MS (*a first application server instance*) was “*selected from a plurality of application server instances based on a load balancing process.*” *Dinker* disclosed these features. EX-1005 ¶49; EX-1003 ¶106.

Dinker disclosed an application server cluster (*a plurality of application server instances*) and explained that the application server instances of the application server cluster included “[a]pplication code [that] may be replicated across multiple application servers in the cluster, enabling a given request to be processed by any of these multiple application servers” (*application server instances*). EX-1005 ¶¶23, 32; EX-1003 ¶107.

⁴ Although gateway 211 is annotated as the *first adapter* in FIG. 2, each of the other gateways are independent examples of a *first adapter* because they perform the same function as gateway 211. EX-1003 ¶104.

For example, *Dinker* illustrated a plurality of application server instances

(red):



EX-1005 FIG. 1 (annotated), FIGs. 2 and 3A (illustrating more servers and various data sharing); EX-1003 ¶108.

Dinker further disclosed the application server cluster included a broker (server) (e.g., web server) (green) that used “load balancing techniques” to select an application server (*a load balancing process*) to service the received requests. EX-1005 ¶49. *Dinker’s* broker (e.g., web server) is an example of a *load balancing component*. While the load balancer/balancing would thus conveniently be a software module of the gateway/adaptor following *Dinker’s* broker teaching, a POSITA understands that the module could also be conveniently co-located in

other components of *Secer*'s distributed NMS, including co-locating the function as a role of a "primary" application server instance of MS 202 following *Dinker*'s framework (EX-1005 ¶¶54, 54) and *Secer*'s teaching of load balancing coordination at the MS 202 (EX-1004 15:13-21) readily extendable to server load, or in a separate component such as monitor 401, 603 sitting between the gateways and server instances (EX-1004 11:37-12:6, FIGs. 4, 6). While obviousness does not require engineering an actual system, a POSITA would readily recognize that any of these options would be viable to implement the well-known techniques and benefits of load balancing. Likewise, load balancer 374 of the '282 Patent was a software module described simply as being "located on any device in NMS 300." EX-1001 9:8-11. *See also* [3d]; EX-1005 ¶49; EX-1003 ¶109.

A POSITA would be motivated to modify *Secer*'s teachings to include multiple instances of MS 202 (*application server instance*) and perform load-balancing among the MSs (*based on a load balancing process*) as taught by *Dinker* for the reasons described in the Motivation to Combine Section. §V.A.3; EX-1003 ¶110. For example, *Dinker* disclosed that multiple application servers facilitated application performance and scalability. EX-1005 ¶¶21, 31. *Dinker* also disclosed using "load balancing techniques" to select an available application server to process requests. EX-1005 ¶49; EX-1003 ¶110.

Moreover, it was already widely known to use a cluster of application server instances for network management and load balancing to distribute the load over the plurality of application server instances for the known benefits of scalability, redundancy, fault tolerance, etc. §V.A.3; §IV.F. In addition, the '282 Patent trivializes the load balancing aspects of the claims as evidenced by the lack of details provided with respect to the load balancing techniques, stating little more than that “**any** load balancing algorithm” may be used, and just be software located “on any device” in the NMS. EX-1001 8:63-64, 9:8-11, 9:54-58 (“The load balancer allows the management work load to be distributed among all server instances in **any desired fashion.**”). Thus, a POSITA would readily recognize the applicability of *Dinker*'s teachings to *Secer* and have a high degree of expected success in combining their teachings to achieve well-understood, predictable benefits. §V.A.3; EX-1003 ¶111.

Thus, *Secer* in view of *Dinker* disclosed this limitation. EX-1003 ¶112.

[1aii] wherein the first adapter processed information comprises event information received by the first adapter from a network element and processed by the first adapter based on a first communication protocol;

Secer disclosed this limitation. EX-1003 ¶¶113-115. As discussed in [1ai], *Secer*'s gateways including adapter functionality (*first adapter*) received the network element data, such as a trap messages, events, or poll information, sent by NEs (*event information received by the first adapter from a network element*) and

processed the network element data using a particular communication protocol (e.g., SNMP, CMIP, etc.) (*processed by the first adapter based on a first communication protocol*) to form the processed network element data (*first adapter processed information*). See [1ai]; EX-1003 ¶113.

Secer described that network elements sent network element data (trap messages, events, or poll information) to a gateway (*event information received by the first adapter from a network element*), and the gateway processed the network element data according to various communication protocols used by the NEs (*processed by the first adapter based on a first communication protocol*). EX-1004 3:52-62; 8:57-9:26. For example, as shown in FIG. 2, the SNMP gateway 211 processed the network element data (*event information*), from network element 214 in accordance with the SNMP protocol (*first communication protocol*). EX-1004 FIG. 2. *Secer* disclosed other communication protocols such as CMIP and referenced other protocols used by, e.g., ATM, SONET, routers, modems, CMIP EMSs, switches, and OSSs/NMSs network elements. EX-1004 9:62-10:3. The SNMP gateway (or other protocol-specific gateway) processed network element data (*first adapter processed information*) was then relayed to MS 202 for further processing. EX-1004 9:3-18; EX-1003 ¶114.

Thus, *Secer* disclosed this limitation. EX-1003 ¶115.

[1b] processing, by the first application server instance, the first adapter processed information based on an event management service to produce application processed information;

Secer disclosed this limitation. EX-1003 ¶¶116-119. As discussed in [1ai], *Secer* disclosed that MS 202 (*first application server instance*) received the processed network element data (*first adapter processed information*). See [1ai]. The processed network element data (including filtered and/or translated events such as trap messages and poll information) was used by MS 202’s “management process” (*event management service*) to identify and generate a “management behavior object” (*to produce application processed information*). EX-1004 10:21-24; EX-1003 ¶116.

Secer disclosed that the processed network element data (*first adapter processed information*) is sent from the gateway (*first adapter*) to MS 202 (*first application server instance*) for further processing. EX-1004 9:13-26. For example, *Secer* disclosed that the processed network element data was generated in response to messages or “events,” examples of which include trap messages and polling information, from NEs. EX-1004 3:52-55; 7:43-46, 9:17-22, 10:62-66. These messages and events related to both fault management (e.g., management of unsolicited messages) and performance management (e.g., polling of NEs). EX-1004 10:21-24. *Secer* disclosed that MS 202 used a “management process” to access “network element and/or gateway” specific “management behavior

objects.” EX-1004 9:34-9:62. Such objects “define[d]...management behavior responsive to particular trap messages or...polling” and specified “one or more distributed gateways which need to execute the defined management behavior.”

EX-1004 9:62-10:4. MS 202 would then push the management behavior object to the appropriate gateway(s). EX-1004 10:4-8. Thus, *Secer*’s MS 202 “management process” took processed network element data, identified the management behavior object, and “pushed” it to appropriate gateway(s) (*processing, by the first application server instance, the first adapter processed information based on an event management service to produce application processed information*). EX-1003 ¶117.

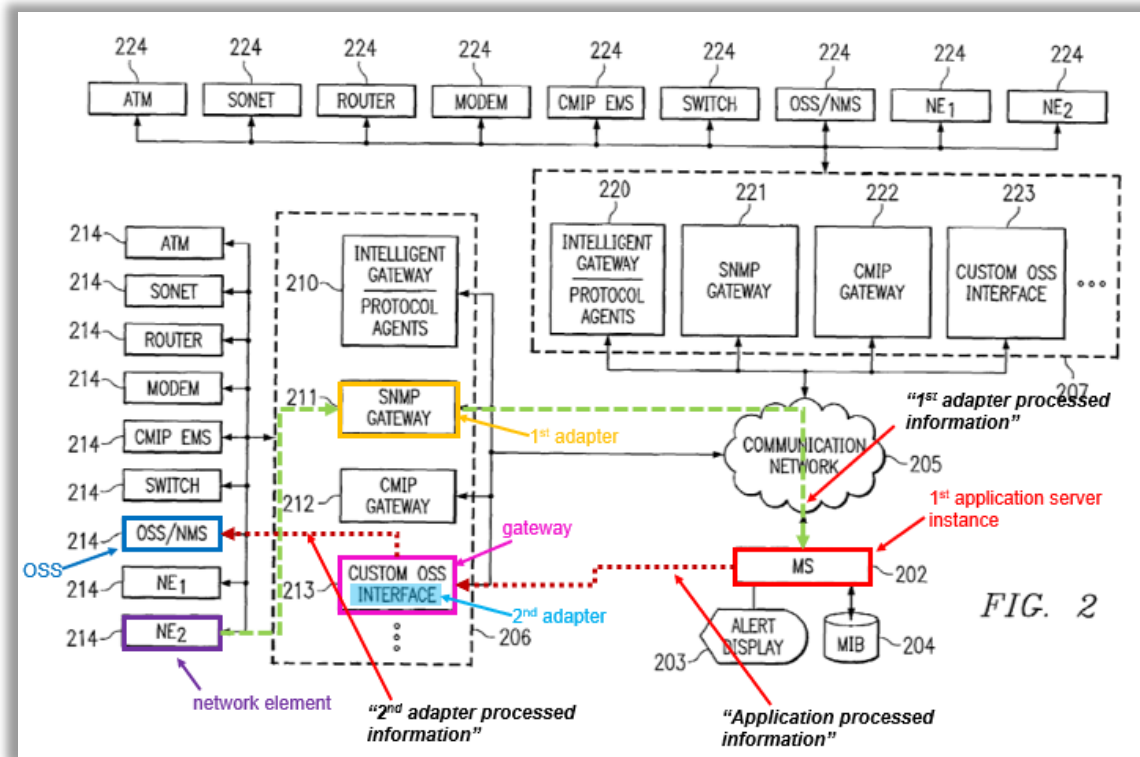
Secer’s disclosure is consistent with the ’282 Patent’s disclosure. For example, the ’282 Patent discloses an event log manager. EX-1001 5:24-29, 5:55-63. The ’282 Patent states that its event management service “collects and records **various events** from the network,” including “SNMP trap” and “periodic performance monitoring,” and “alarm[s].” EX-1001 55-63. These are the same sort of “events” handled by MS 202’s “management process.” Therefore, *Secer* disclosed an “*event management service*.” EX-1003 ¶118.

Thus, *Secer* disclosed this limitation. EX-1003 ¶119.

[1ci] *sending, by the first application server instance, the application processed information to a gateway device,*

Secer disclosed this limitation. EX-1003 ¶¶120-124. *Secer* described that once MS 202 (*first application server instance*) identified the management behavior object (*application processed information*), MS 202 “push[ed]” the management behavior object to the appropriate gateway (*a gateway device*) to which the management behavior relates (*sending, by the first application server instance, the application processed information to a gateway device*). EX-1003 ¶120.

For example, *Secer* disclosed gateways, such as custom OSS interface gateway 213, (*gateway device*) (pink) coupled between, OSS 214 (blue) and MS 202 (*first application server instance*) (red):



EX-1004 FIG. 2 (annotated); EX-1003 ¶121.

Secer described that the management behavior object (*application processed information*) is “push[ed]” from MS 202 (*sending, by the first application server instance*) (red) to gateway 213⁵ (*gateway device*). EX-1004 10:4-11. A POSITA understood that Secer disclosed sending data to the gateway 213 as defined by the management behavior object. As discussed above, the OSS oversaw NEs, and any

⁵ Although gateway 213 is annotated as the *gateway device*, another example is gateway 223, which had similar functionality and meet the requirements of the claimed *gateway device*. EX-1003 ¶122.

information related to NEs (e.g., performance characteristics, faults, configuration, etc.) would have been understood as having been propagated to the OSS as part of its management role. EX-1004 1:52-2:1 (“MSs,’ encompass...NMSs”); 9:17-10:11; §IV.F.1. Thus, A POSITA would have understood that *Secer’s* management behavior object would specify the OSS gateway 213 to ensure the propagation of this information. EX-1003 ¶122.

As discussed above, *Secer’s* gateways⁶ processed and relayed communication between central MS 202 and OSS 214, and included functionality to convert communications of different protocols. EX-1004 8:60-9:13. *Secer’s* disclosed gateways including both *gateway* functionality and *adapter* functionality consistent with the ’282 Patent’s gateway and adapters. For example, the ’282 Patent explains that its gateway “**processes the event and distributes the information** appropriately to each NB adapter” and “**relays management information** and operations between server 232 and an external OSS 208.” EX-1001 3:25-27, 4:4-5. The ’282 Patent also describes that its adapters “process information to facilitate communication between NMS and OSS **using different types of protocols.**” EX-1001 2:53-61; EX-1003 ¶123.

⁶ *Secer’s* gateways are both the claimed adapter and gateway device. EX-1003 ¶123.

Thus, *Secer* disclosed this limitation. EX-1003 ¶124.

[1cii] wherein the gateway device is one of a plurality of gateway devices respectively associated with the plurality of application server instances and is configured to transfer the application processed information to a second adapter of a plurality of second adapters configured to process the application processed information based on a second communication protocol to produce second adapter processed information and transfer the second adapter processed information to an operation support system device; and

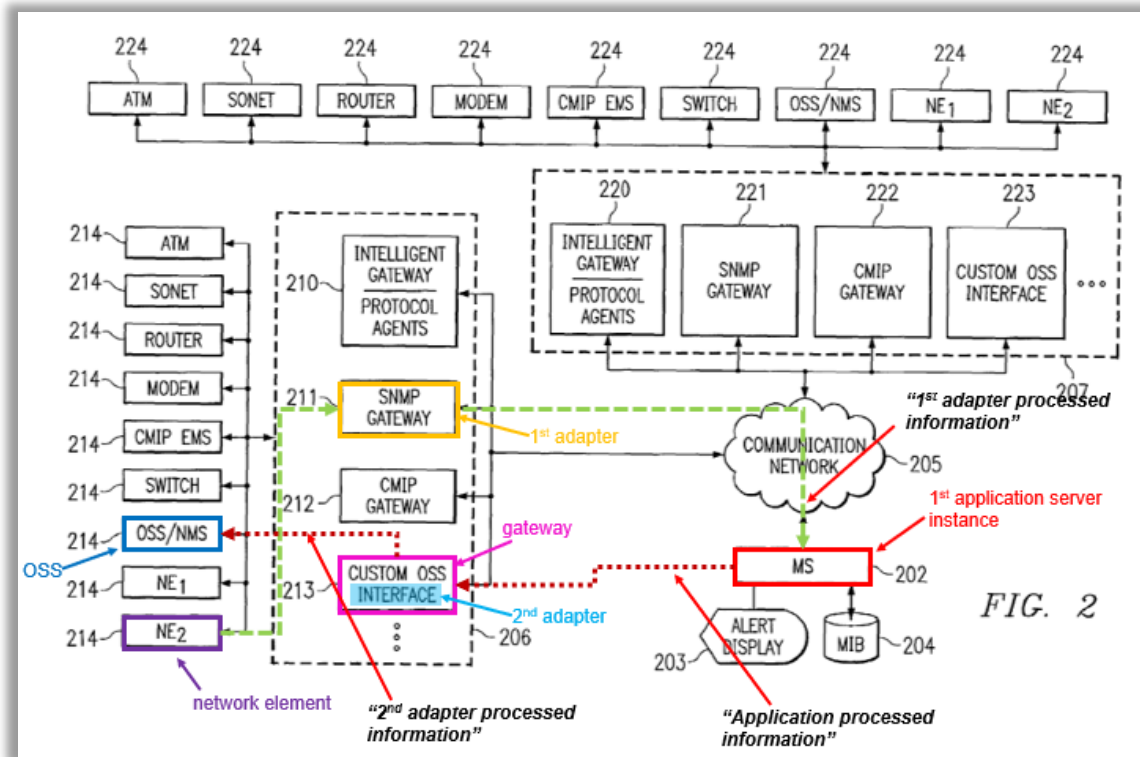
Secer in view of *Dinker* rendered obvious this limitation. EX-1003 ¶¶125-129.

Secer disclosed custom OSS interface gateway 213 among other gateways (e.g., gateways 210-213, 220-223) (each a *gateway device* and an *adapter*) in communication with MS 202, a load-balanced, fault-tolerant server cluster as modified by *Dinker* (*wherein the gateway device is one of a plurality of gateway devices respectively associated with the plurality of application server instances*) (see [1ai]) and the gateways received management behavior objects (*application processed information*) from MS 202. *Secer*'s gateways included both gateway functionality to transfer the communication (*gateway device*) and adapter functionality (*second adapter*) to process the management behavior object (*configured to transfer the application processed information to a second adapter and configured to process the application processed information based on a second communication protocol to produce second adapter processed information,*

respectively). *Secer's* custom OSS interface gateway 213 communicated with OSS 214 according to the appropriate OSS protocol (*transfer the second adapter processed information to an operation support system device*). *Secer's* other gateways (e.g., gateways 223) (*a plurality of second adapters*) also communicated with an OSS (another example of *configured ... transfer the second adapter processed information to an operation support system device*). EX-1003 ¶126.

As illustrated, gateway group 206 (*a plurality of gateway devices*) included gateways 210-213 (each, individually, a *gateway device*) where custom OSS interface gateway 213 (*the gateway device*) (pink) received the management behavior object (*application processed information*) from the load-balanced, fault-tolerance MS 202 (*associated with the plurality of application server instances*):⁷

⁷ Each gateway device received management behavior objects from MS 202 and, therefore, were *respectively associated with the plurality of application server instances*. See also [1ai]; EX-1003 ¶127.



EX-1004 FIG. 2 (annotated). As described in limitation [1ci], gateway 213, as well as other gateways (including 223) included functionality to recognize and process (filter and translate) different protocols, which the '82 Patent refers to as an *adapter*. EX-1004 2:29-35; 9:3-16; cl. 23. Thus, gateways 213 and 223 were a *plurality of second adapters*. The gateways included, custom OSS interface gateway 213 (pink), in addition to 223, that included functionality (light blue) to recognize and process different protocols (*second adapter*).⁸ EX-1004 9:3-16; EX-1003 ¶127.

⁸ Although gateway 213 is annotated as including the *second adapter* in FIG. 2, this is only one example, and because each of the gateways 210-212 and 220-223

Secer disclosed that gateway 213 monitored OSS 224 having a particular communication protocol. EX-1004 9:3-16. Moreover, *Secer* disclosed that MS 202 (*application server instance*) communicated (e.g., “push[ed]”) the created management behavior (e.g., the object defining such management behavior) to the appropriate gateways and NEs to which the management behavior relates. EX-1004 10:4-12. In other words, the management behavior object (*application processed information*) for gateway 213 and OSS 214 would be communicated to gateway 213 (pink) for processing using, e.g., the OSS interface protocol (*to process the application processed information based on a second communication protocol to produce second adapter processed information*). EX-1004 10:4-12. Custom OSS interface gateway 213 would send the processed management behavior object (*second adapter processed information*) to OSS 214 (*transfer the second adapter processed information to an operation support system device*). EX-1004 10:4-12, 18:12-15, FIG. 2. A POSITA understood that the custom OSS interface gateway 213, is an intermediary and bridges the communication between the OSS 214 and MS 202, where the custom OSS interface gateway 213 forwarded

have similar functionality; any of these gateways would satisfy the claimed *second adapter* for similar reasons. EX-1003 ¶127.

the management behavior object information (failure, performance data, etc.) to the OSS 214 because the OSS manages the network. [1ci]; §IV.F.1; EX-1003 ¶128.

Therefore, *Secer* in view of *Dinker* rendered obvious this limitation. EX-1003 ¶129.

[1d] in response to determining that the first application server instance has become disabled, facilitating establishing an association between the first adapter and a second application server instance of the plurality of application server instances and between the gateway device and the second application server instance.

Secer in view of *Dinker* rendered obvious this limitation. EX-1003 ¶¶130-140. *Secer* detailed techniques and benefits of using distributed gateways, and how to detect and recover from a failure of a gateway. EX-1004 7:64-8:6, 10:12-16, 10:31-40, 13:37-17:8, FIGs. 3, 5, 6. In response to such failure, recovery was achieved by reassigning management (*association[s]*) of NEs to an available gateway. EX-1004 10:45-52. *Secer* stored *associations* between gateways and managed NEs in management information base (MIBs), which was part of or coupled to MS 202, containing management behavior objects. EX-1004 9:34-44, FIG 2. These “objects may have an attribute specifying the relationship of such objects to the network elements and/or gateways.” EX-1004 9:53-55. Thus, when a new gateway became responsible for managing a NE, such objects would be “modified” with the new gateway so that the “central MS may determine to which gateways and/or network elements the object relates and implement the

management behavior defined by such object for the related network elements and/or gateways.” EX-1004 9:58-62. EX-1003 ¶130.

Secer specifically taught how MS 202 would coordinate recovery of the distributed NMS in the event of a failure of a gateway. EX-1005 13:37-17:8, FIGs. 3, 5, 6. *Secer* taught managing the relationships in “gateway management description information” stored local (internal or external) to MS 202. EX-1004 14:47-53, FIG. 6. “Gateway management description information... include[d] a list of managed devices to which each gateway is assigned management responsibility.” EX-1004 14:53-56. Such information also included an “available gateway list” and “gateway load.” EX-1004 14:64-15:21. Once failover was detected (e.g. by monitor 603) and an appropriate gateway identified, “recovery information” was “provided to such ‘substitute’ gateways to enable them to recover management of the network elements.” EX-1004 15:22-60. “Thus, a substitute gateway [] is assigned management responsibility for a device” in place of “a failed gateway” in a timely and efficient manner. EX-1004 15:62-63, 10:49-53; EX-1003 ¶131.

While *Secer* disclosed recovering from a failed gateway, including establishing the necessary associations for a second gateway, it did not expressly disclose doing so for failures of an application server instance in a server cluster. Yet, *Secer* expressly disclosed that its failover teachings should be applied to other

components including “devices other than gateways for managing network elements” and those in “client/server relationships.” EX-1004 18:12-15; EX-1003 ¶132.

In the same vein, *Dinker* disclosed using distributed application server instances, detecting a failure of a first application server instance and, in response, facilitating establishment of associations with a second application server instance, all consistent with server clustering. EX-1005 ¶¶32, 33, 57, 30; §§IV.F.1, IV.F.4. *Dinker* taught application server clustering so that in the case of “failure on one application server” “requests may be routed to and processed by other application servers in the cluster.” ¶30. Likewise, *Dinker* observed that “in computer systems” it was desirable to “failover mechanisms to increase the fault-tolerance of an application server cluster”, including a “backup operational mode” where secondary components took over for primary components when they fail. EX-1005 ¶33; EX-1003 ¶133.

In such a cluster, *Dinker* taught a framework in which at least one server should be designated a “primary” application server to “manage and provide processing information” necessary for the other servers in the cluster to operate, such as session information for client requests. EX-1005 ¶¶54-55. The information necessary for this role was mirrored as back-up data by another application server designated as a “backup” for the extra roles of the “primary”

server. EX-1005 ¶57, e.g. Fig. 2 (showing shared/mirrored data among servers).

When the “primary” server failed, the backup server already contained the backed-up data and would be promoted so as to support the server cluster in the role of primary server, providing the extra management and data necessary for the remaining servers in the cluster to continue operations uninterrupted. EX-1005 ¶57. *Dinker* taught that all the servers in the cluster should send each other “heartbeat” messages such that a failure of a server instance would be quickly detected. EX-1005 ¶85. *Dinker* explained that, “[i]n response to determining that a cluster failure occurred, one or more backup application server computers may be promoted to a primary server role” (*in response to determining that the first application server instance has become disabled, facilitating establishing an association ...*). EX-1005, ¶¶35, 57; EX-1003 ¶134.

As already discussed, it would have been obvious to implement *Secer*’s MS 202 as a server cluster following *Dinker*. See §§V.A.4.[1ai]; V.A.3. Likewise, it would have been obvious to extend *Secer*’s failover teachings for gateways to also include failover between servers of the MS 202 server cluster following *Dinker*’s failover teachings for application servers. EX-1003 ¶135.

While obviousness does not entail engineering an actual system (and the ’282 Patent provides no information at all), a POSITA would readily recognize that *Secer*’s coordination of the reassignment of management roles, and shared state

information, to recover from a failed gateway (*see, e.g.*, discussion including *Secer's* MIBs and FIG. 6 above) would readily be beneficially extended to coordinate (*facilitate establishing associations*) the recovery of a failed MS 202 server instance as a responsibility of a designated “primary” application server instance in the MS 202 cluster, with such MIB and state information mirrored to the designated back-up server instance in the framework taught by *Dinker*. EX-1005 ¶¶ 54-54, FIGs. 1, 2. Thus, the “primary” server instance would coordinate any failover of the other server instances using the same techniques *Secer* already detailed for gateway failures. If the “primary” server instance itself failed, the designated back-up server would readily be promoted to primary, following *Dinker's* teachings for clustered servers, and thus carry out the reassignment of the gateways/adapters and network elements to itself or another available server in the cluster. *Secer* as modified by *Dinker* therefore disclosed the primary server (*first application server instance*) utilized a back-up server (*second application server instance*) to continue providing service. EX-1003 ¶136.

As to the actual detection of a server instance failure, *Secer* already disclosed that the MS 202 can detect gateway failures, and that that functionality can also be housed in “gateway monitors” (*see* monitors 401, 603 in FIGs. 4, 6) that communicate failures back to MS 202. EX-1004 0:31-52. It would have been obvious to implement heartbeat messages between the server instances of an MS 202

cluster as taught by *Dinker* (EX-1005 ¶85), or to adapt the monitors 401, 603 of *Secer* to add that functionality, for the self-evident benefit of quickly detecting the failure of a server in the MS 202 cluster. EX-1003 ¶137.

A POSITA understood that with *Secer* modified by the teachings of *Dinker* to include a second *application server instance*, when the *first application server instance* fails, the system would establish the associations (*facilitating establishing an association*) necessary to continue operation, including between the new server instance and both gateway 211 (*first adaptor*) (gold) and gateway 213 (*the gateway device*) as discussed above. *See also* §V.A.3; [1ai]. EX-1003 ¶138. It would have been readily apparent to a POSITA to *facilitate establishing the association* needed to maintain continuity of communication in the event of a failure of a *first server instance* of *Secer*'s MS 202 following the server clustering and failover teachings of *Dinker*. EX-1003 ¶¶138.

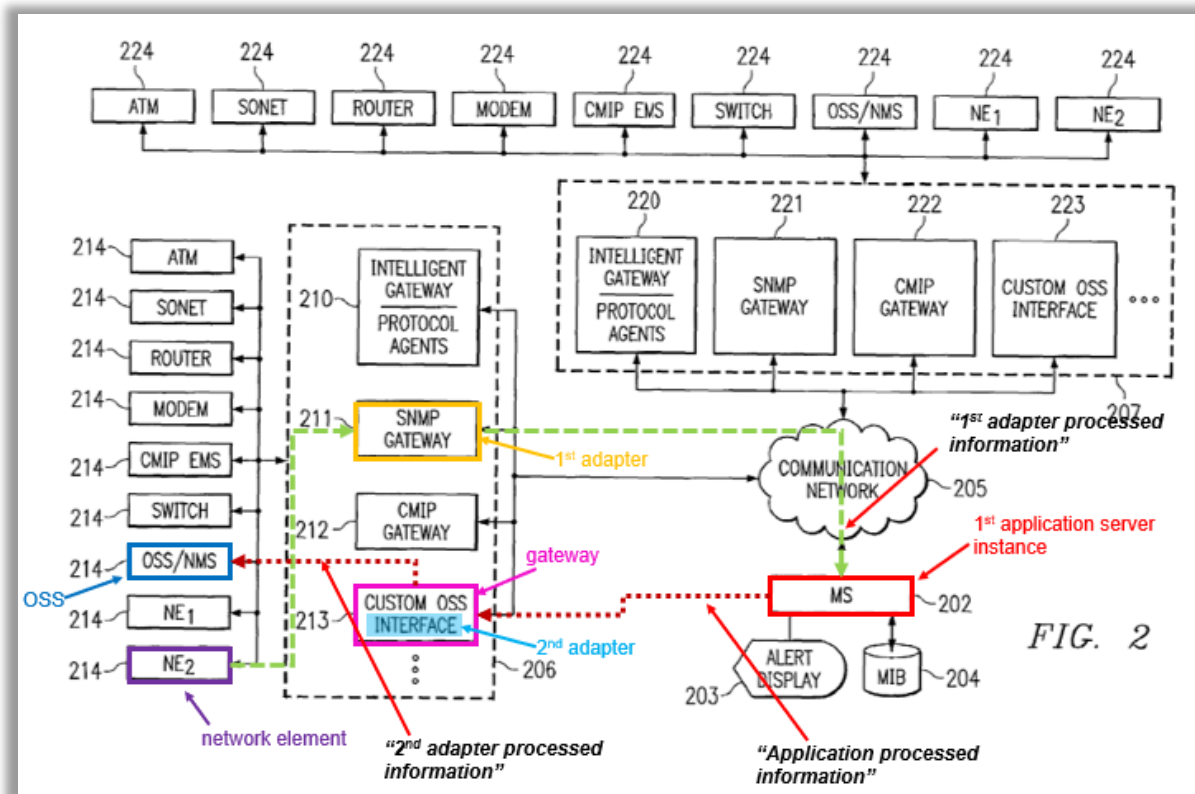
A POSITA would have been motivated for the reasons above, as well as those described in the Motivation to Combine. §V.A.3; *see also* VI.A.4.[1ai]; EX-1003 ¶139.

Thus, *Secer* in view of *Dinker* rendered obvious this limitation. EX-1003 ¶140.

Claim 2

[2] The method of claim 1, wherein the first application server instance is configured to execute on a separate physical machine from the first adapter.

Secer disclosed this limitation. EX-1003 ¶¶141-143. Secer’s disclosed “distributed” gateways, e.g., SNMP gateway 211 including adapter functionality (first adapter) separated from MS 202 (first application server instance configured to execute on a sperate physical machine):



EX-1004 FIG. 2 (annotated); EX-1003 ¶¶141.

Secer further explained that MS 202 can exist within an “client/server environment” where it was routine for the server to be a separate physical machine. EX-1004 18:12-15, FIG. 1, 6:33-38, 6:54-57, 6:64-7:1; EX-1003 ¶¶142. Secer also

disclosed that the gateways (*first adapter*) were devices geographically distributed from MS 202 (*first application server instance*). EX-1004 9:13-18, 13:37-42.

Moreover, *Secer* disclosed communication network 205 (e.g., the Internet) between the MS and gateways. Thus, at least the two devices (*machines*, i.e., MS 202 and gateway 211) are physically separate. EX-1003 ¶142.

Thus, *Secer* disclosed this limitation. EX-1003 ¶143.

Claim 3

[3pre] *A system, comprising:*

Secer in view of *Dinker* rendered obvious *a system* comprising limitations [3a]-[3g] discussed below. EX-1003 ¶¶144-151.

[3a] *a first application server instance configured to receive first adapter processed information from a first adapter[,]*⁹ *process the first adapter processed information based on an event management service to yield application processed information,*¹⁰ *and*

Secer disclosed this limitation for the reasons discussed in limitations [1ai] and [1b]. EX-1003 ¶145.

⁹ Punctuation (“comma (,)”) added for clarity.

¹⁰ “Application processed information” is interpreted as “application server processed information” for the purposes of this petition. *See* [3b], [3e].

[3b] send the application server processed information to a gateway device,

Secer disclosed this limitation for the reasons discussed in limitation [1ci].

EX-1003 ¶146.

[3c] wherein the first adapter processed information comprises event information from a network element that has been processed by the first adapter based on a first communication protocol; and

Secer disclosed this limitation for the reasons discussed in limitation [1aii].

EX-1003 ¶147.

[3d] a load balancing component configured to select the first application server instance from a plurality of application server instances based on a load balancing process;

Secer in view of *Dinker* rendered obvious this limitation for the reasons discussed in limitation [1ai]. EX-1003 ¶148.

[3e] wherein the gateway device is one of a plurality of gateway devices respectively associated with the plurality of application server instances and is configured to transfer the application server processed information to a second adapter of a plurality of second adapters configured to process the application server processed information based on a second communication protocol to yield second adapter processed information, and

Secer in view of *Dinker* rendered obvious this limitation for the reasons discussed in limitation [1cii]. EX-1003 ¶149.

[3f] send the second adapter processed information to an operation support system device, and

Secer disclosed this limitation for the reasons discussed in limitation [1cii].

EX-1003 ¶150.

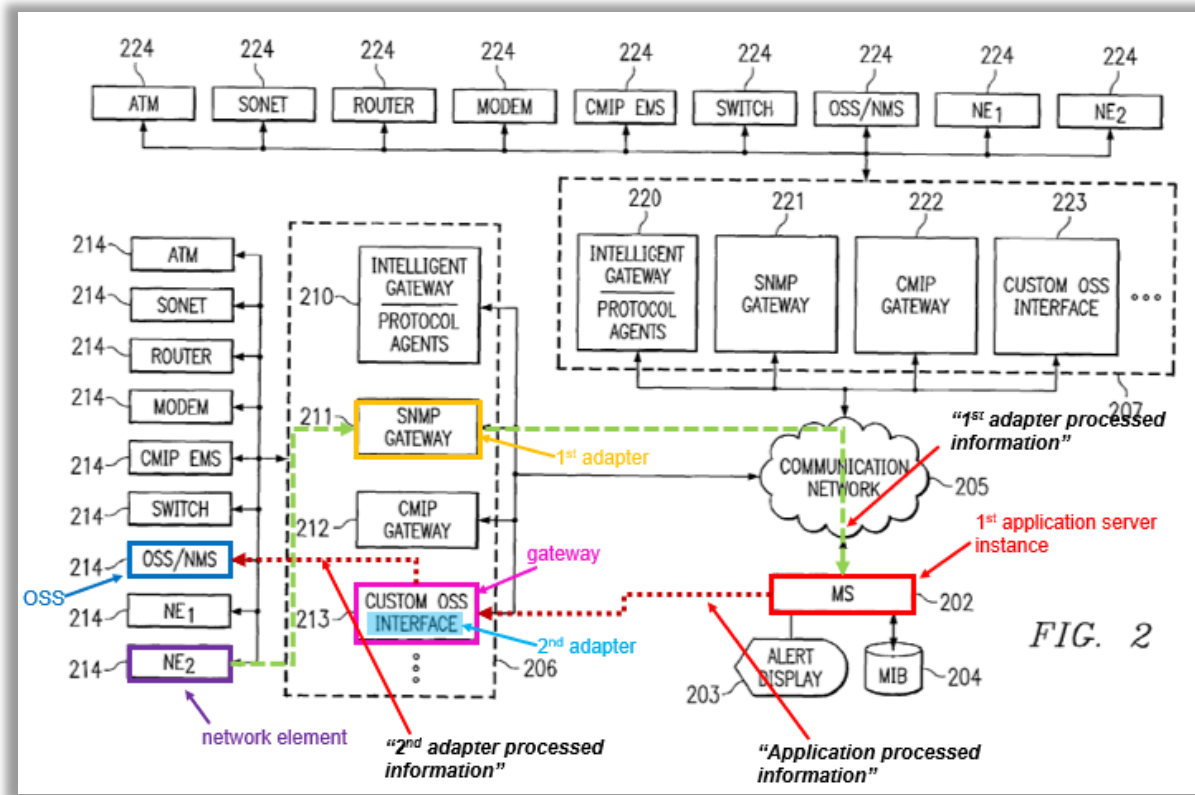
[3g] wherein, the first adapter and the gateway device are further configured to, in response to disablement of the first application server instance, establish an association with a second application server instance of the plurality of application server instances.

Secer in view of *Dinker* rendered obvious this limitation for the reasons discussed in limitations [1ci-1cii] and 1[d]. EX-1003 ¶151.

Claim 4

[4] The method of claim 1, further comprising converting, by the second adapter, a protocol specific message associated with the first application server instance to a formatted message associated with the second communication protocol.

Secer disclosed this limitation for the same reasons discussed in limitations [1ci-1cii]. EX-1003 ¶¶152-153. *Secer* disclosed a custom OSS interface gateway 213 that included adapter functionality (*second adapter*) processed communication (e.g., management behavior object) (red dashed arrow) between OSS 214 and MS 202 (*converting ... a protocol specific message associated with the first application server instance to a formatted message associated with the second communication protocol*):



EX-1004 FIG. 2 (annotated). Custom OSS interface gateway 213 (*second adapter*) received communications (management behavior objects from MS 202 (*first application server instance*)) that operated according to its own protocol and a protocol used by communication network 205. EX-1004 8:62-9:3; FIG. 2.

Custom OSS interface gateway 213 (*second adapter*), included functionality (light blue) to recognize and process (convert) different protocols.¹¹ EX-1004 8:62 9:16;

¹¹ Although gateway 213 is annotated as including the *second adapter* in FIG. 2, this is only one example. *E.g.*, Gateway 223 would also be a second adapter. EX-1003 ¶152.

see also, [1cii] (discussing the operation of *Secer*'s gateway (*second adapter*)).

Secer disclosed the custom OSS interface gateway 213 converted the received communication from the MS to protocol for communication with OSS 214. EX-1004 FIG. 2; EX-1003 ¶152.

Thus, *Secer* disclosed this limitation. EX-1003 ¶153.

Claim 5

[5] *The method of claim 1, further comprising selecting the first application server instance based on a determination that the first application server instance has a lowest processing load of the plurality of application servers instances.*

Secer in view of *Dinker* rendered obvious this limitation. EX-1003 ¶¶154-156. As discussed in [1ai], *Secer* in view of *Dinker* disclosed selecting the *application server instance based on load balancing process*. See [1ai]; §V.A.3; EX-1003 ¶154.

Secer also expressly disclosed selecting a gateway based on a lowest processing load because it had the “minimum load,” e.g., the lowest CPU load. EX-1004 14:19-45, 15:16-21, 17:9-33 (“minimum load”), FIG. 7. As discussed above, *Secer* also stated, that its “detection and recovery techniques... may be applied to devices other than gateways.” EX-1004 18:12-15. A POSITA would have recognized that such devices would include *Dinker*'s load-balanced application server cluster. See [1ai]. Further, it was well known that load-balancing included “evening” the load, and the term “balance” connotes

distributing the load to servers with less load to bring about this “balance.”

§IV.F.3 (citing EX-1012 ¶81 (evenly distributing load); EX-1011 8:15-20, EX-1010 15:35-40). EX-1003 ¶155.

Thus, *Secer* in view of *Dinker* rendered obvious this limitation. EX-1003 ¶156.

Claim 6

[6] The method of claim 1, wherein the first communication protocol and the second communication protocol comprise one or more communication protocols associated with at least one of extensible markup language, simple network management protocol, common object request broker architecture, or transaction language 1.

Secer disclosed this limitation. EX-1003 ¶¶157-159. *Secer* stated, “[b]ecause different types of network elements may communicate in different protocols, management systems may utilize different processes for managing different types of network elements...[a] Simple Network Management Protocol (SNMP) gateway process may be implemented for managing SNMP devices” (*wherein the first communication protocol and the second communication protocol comprise one or more communication protocols associated with at least one of ... simple network management protocol*). EX-1004 1:64-2:1; 2:4-6; 2:26-32; 8:62-9:16; FIGS. 2. *Secer* disclosed the gateways using the SNMP protocol (*first communication protocol*) and an OSS protocol (*second communication protocol*). EX-1003 ¶157.

A POSITA understood that such OSS protocols included, or would have been obvious to include, SNMP because SNMP was known as one of the most common communication protocols in NMSs. §IV.F.1 (citing, e.g., EX-1009 3:49-54); EX-1003 ¶158 (citing EX-1012 ¶75). Likewise, a POSITA would have understood that such OSS protocols included, or it would have been obvious to include “*Common Object Request Broker Architecture*” (CORBA), because that was also a well-known protocol for communicating between components in an OSS. §IV.F.1 (citing, e.g., EX-1008 4:63-67); EX-1003 ¶158.

Thus, *Secer* disclosed this limitation. EX-1003 ¶159.

Claim 7

[7] *The method of claim 1, further comprising at least one of collecting, recording, or publishing the event information in accordance with the event management service for access by the operation support system device.*

Secer disclosed this limitation. EX-1003 ¶¶160-161. *Secer* disclosed the processed network element data (including filtered and/or translated events such as trap messages and poll information (*event information*)) was used by MS 202’s “management process” (*event management service*) to identify and generate a “management behavior object.” EX-1004 9:17-26; 9:23-26; 10:21-24. MS 202 collected and stored this event information. EX-1004 9:34-44. *Secer*’s FIG. 2 showed OSS 214 coupled to MS 202 that collected the *event information*. EX-1004 FIG. 2. A POSITA understood that OSS 214 has access to the stored *event*

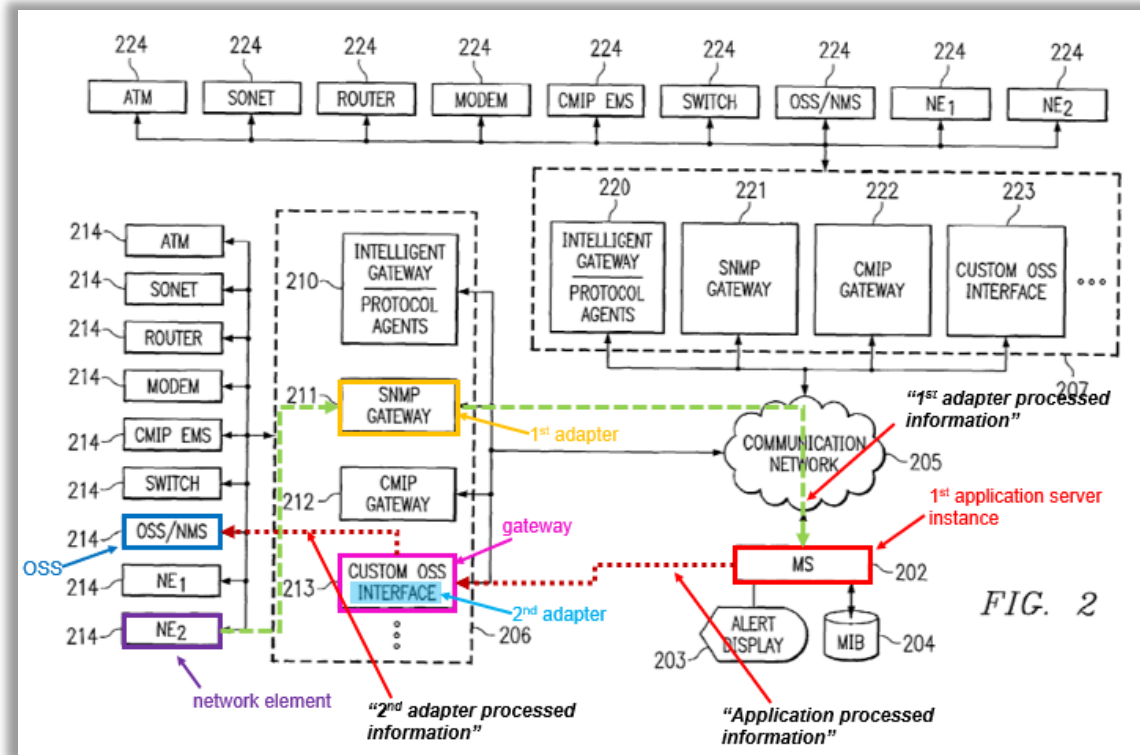
information of *Secer* because it is coupled to MS 202 via the custom OSS interface gateway 213 where the OSS is tasked with managing the network and NEs. *See* [1ci-1cii]. Moreover, the NE event information associated with a management behavior object would be communicated to the OSS (*publish[ed]*) to inform the OSS of the conditions of the network it managed. *See* [1c]-[1cii]; §IV.F.1; EX-1003 ¶160.

Secer disclosed this limitation. EX-1003 ¶161.

Claim 8

[8] *The method of claim 1 wherein the plurality of gateway devices comprise a plurality of physically or logically separated gateway devices.*

Secer disclosed this limitation for the reasons discussed with respect to limitation [1cii]. EX-1003 ¶¶162-165. *Secer* showed the distributed gateways are physically separated gateway devices (including gateway 213 and 223) (*plurality of gateway devices comprise a plurality of physically...separated gateway devices*).



EX-1004 FIG. 2 (annotated); EX-1003 ¶162.

Secer stated, “Each of the distributed gateways may, for example, be any suitable processor-based device...” EX-1004 9:13-15. *Secer* also disclosed that gateways may be in different geographic locations. EX-1004 8:57-60. Because *each gateway* is a separate device in a geographic location separate from other gateways, *Secer* disclosed the gateways are *physically separated gateway devices*. EX-1003 ¶163.

Moreover, *Secer* depicted the gateways are *logically separated gateway devices* in FIG. 2. EX-1004 FIG. 2; EX-1003 ¶164.

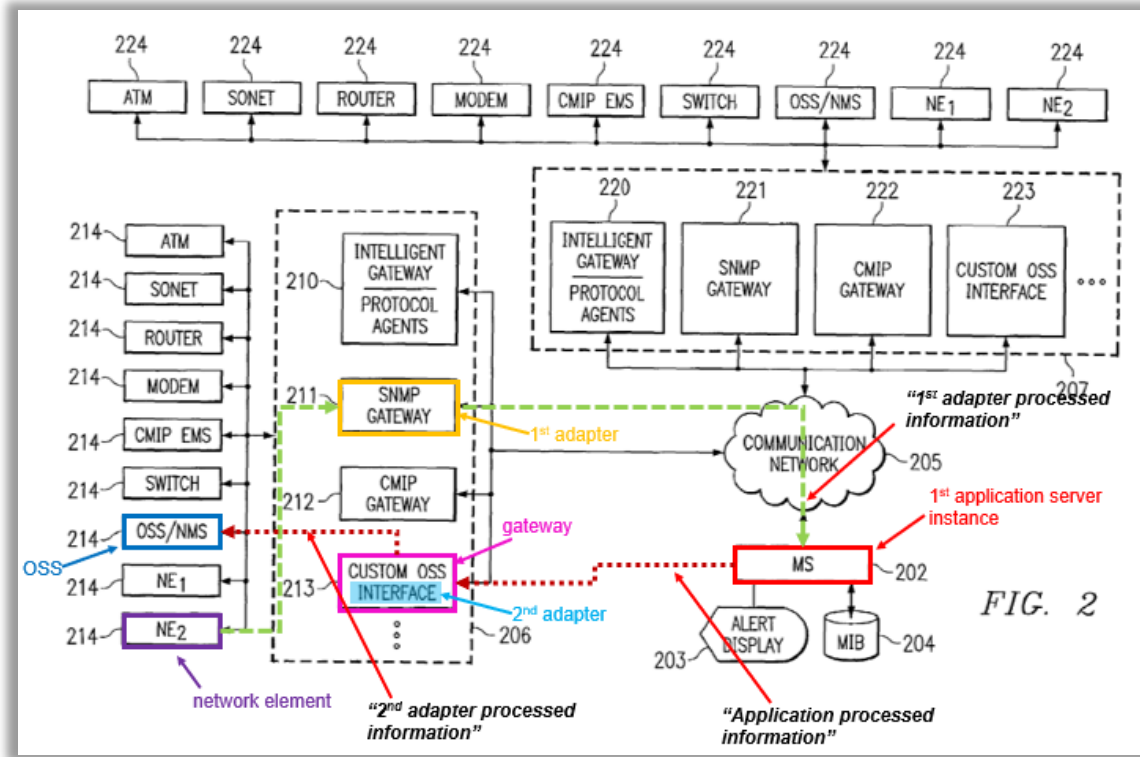
Thus, *Secer* disclosed this limitation. EX-1003 ¶165.

Claim 9

[9] *The method of claim 1, wherein the gateway device is a physically and logically separate machine from the first application server instance.*

Secer disclosed this limitation. EX-1003 ¶¶166-170. *Secer's* FIG. 2 disclosed gateways 213 (*gateway device is a physically and logically separate machine*) are coupled to MS 202 over the communication network and separate machines from MS 202 (*from the first application server instance*). EX-1004 FIG. 1, 6:33-38, 6:54-57, 6:64-7:1, 18:12-15 (“client/server environment”); EX-1003 ¶166.

For example, the gateways (*gateway device*) (e.g., gateways 210-213) of FIG. 2 are connected to communication network 205 and MS 202 (*first application server instance*) is connected to communication network 205.



EX-1004 FIG. 2 (annotated); EX-1003 ¶167.

Secer further stated, “Each of the distributed gateways may, for example, be any suitable processor-based device.” EX-1004 9:13-15. *Secer* also stated “[t]he gateways are distributed from MS 202.” EX-1004 10:12; EX-1003 ¶168.

Secer further disclosed that a gateway process ran on a gateway device and stated, the “gateway process (e.g., software executing on the gateway) and the gateway hardware (e.g., the processor-based device on which the gateway process is implemented).” EX-1004 11:5-7. A gateway process that ran on a gateway “device” would be logically separate from MS 202 (*first application server*

instance) because it was located in an independently separate “device.” EX-1003 ¶169.

Thus, *Secer* disclosed this limitation. EX-1003 ¶170.

Claim 10

[10] *The method of claim 1, further comprising converting, by the first adapter, a message associated with the first communication protocol to a protocol specific message associated with the first application server instance.*

Secer disclosed this limitation. EX-1003 ¶¶171-172. *Secer* depicted a gateway having adapter functionality (*first adapter*) that communicated with a NE in a first communication protocol (e.g., SNMP, CMIP, etc.) (*a message associated with the first communication protocol*) and further communicated with MS 202 (*converting... to a protocol specific message associated with the first application server instance*) over a communication network 205. EX-1004 8:62-9:17. Communication network 205 operated using different protocols (e.g., IP) from that used by the NEs (*first communication protocol*) and required conversion. EX-1004 8:43-56. Further, the MS 202 also operated using a different protocol, as evidenced by the gateway translations (*converting*). EX-1004 7:16-19 (gateways “are responsible for protocol translations, for such protocols as SNMP and CMIP”). EX-1003 ¶171.

Thus, *Secer* disclosed this limitation. EX-1003 ¶172.

Claim 11

[11] *The method of claim 1, further comprising processing, by the gateway device, a function associated with the application server processed information.*

Secer disclosed this limitation. EX-1003 ¶¶173-175. *Secer* disclosed a custom OSS interface gateway 213 (*gateway device*) that received and processed a management behavior object that defined a management behavior (e.g., need to respond to particular trap messages or perform defined polling activities) (*processing... a function associated with the application server processed information*) for sending to the OSS. EX-1003 ¶173.

Secer disclosed that the “management behavior” described the gateway function behavior that is responsive to a particular trap message or management behavior of the gateway for polling network elements. EX-1004 9:49-10:4. Based on the “management behavior” (*application server processed information*) the gateways (*gateway device*) operated in accordance with the “management behavior” to manage/recover management of the network elements. EX-1004 9:62-10:4; EX-1003 ¶174.

Thus, *Secer* disclosed this limitation. EX-1003 ¶175.

Claim 12

[12] *The method of claim 1, wherein the event information relates to a configuration of the network element.*

Secer disclosed this limitation. EX-1003 ¶¶176-177. *Secer* disclosed gateways and MS 202 collected trap messages, events, and poll information (*event information*) to gather information about various operational characteristics of NEs (*...relates to a configuration of the network element*). EX-1004 2:21-25, 9:3-5, 14:35-42, 9:13-18. *Secer* also disclosed that the MS was “implemented...for managing... network elements.” EX-1004 1:55-60. The management of NEs included the configuration of the NEs. §IV.F.1; EX-1003 ¶176.

Thus, *Secer* disclosed this limitation. EX-1003 ¶177.

Claim 13

[13] *The method of claim 1, further comprising executing the first adapter and the second adapter on one or more virtual machines.*

Secer disclosed this limitation. EX-1003 ¶¶178-180. *Secer* disclosed that distributed gateways (*first adapter and second adapter*) (see [1ai], [1cii]) are remote from and connected to MS 202. EX-1004 8:57-60 (stating “gateway group 206 may be implemented at one geographic location of managed network and group 207 may be implemented at another geographic location.”). *Secer* also disclosed the gateways included “gateway process[es] (e.g., software executing on the gateway)” and, as shown in FIG. 1, multiple gateway processes run on the MS 20 in *virtual machines*. EX-1004 11:5-8, EX-1003 ¶178.

To the extent *Secer* does not expressly disclose that its gateways are implemented in virtual machines, a POSITA understood that it was common to logically partition software code across physical machines, and thus it was merely a design choice to implement the adapter functionality software virtually. EX-1003 ¶179 (citing EX-1005 ¶32). Thus, a POSITA would have found it obvious to implement multiple gateways on a single machine like those disclosed in *Secer*'s FIG. 1. EX-1003 ¶179.

Thus, *Secer* disclosed and/or rendered this limitation obvious. EX-1003 ¶180.

Claim 14

[14] *The method of claim 1, wherein the first adapter and the second adapter are configured to communicate with the first application server instance over a remote method invocation interface.*

Secer disclosed this limitation. EX-1003 ¶¶181-183. *Secer* disclosed geographically distributed gateways (*first adapter, second adapter*) that are coupled to and communicated with MS 202 (*configured to communicate with the first application server instance*) over a network (*remote method invocation interface*) as discussed with reference to claim [1]. EX-1004 9:17-18, 9:22-26, EX-1004 8:57-60; EX-1003 ¶181.

Because the gateways were remote from MS, they required an interface to send network element data (*event information*) and receive the responsive

management behavior objects (*application process information*) over the communication network with the MS. Thus, *Secer* disclosed a *remote method invocation interface*. EX-1003 ¶182.

Thus, *Secer* disclosed this limitation. EX-1003 ¶183.

Claim 15

[15] *The system of claim 3, wherein the event information is associated with a configuration of the network element.*

Secer disclosed this limitation for the reasons discussed in claim 12. See [12]; EX-1003 ¶184.

Claim 16

[16] *The system of claim 3, wherein the first application server instance comprises a performance manager component configured to collect performance data from a plurality of network elements respectively associated with a plurality of first adapters including the first adapter.*

Secer disclosed this limitation. EX-1003 ¶¶185-187. *Secer* disclosed “performance management” by MS (*performance manager component*) that is based on the “collection of data from the network elements,” including performance information (*collect performance data from a plurality of network elements respectively associated with a plurality of first adapters including the first adapter*). EX-1003 ¶¶185.

For example, the network element data included “information regarding the performance of network element” and “operational characteristics of such network

element(s)” (*performance data*) collected by the gateways (*first adapters*) and then transmitted to the MS 202 (*first application server instance*). EX-1004 2:36-61, 9:13-18; 9:22-26; 10:21-24; 1:57-60, 2:36-61 (performance of network elements included “workload,” available memory capacity, etc.”). *Secer’s* performance management occurred in MS 202 because the data from the plurality of NEs is collected and processed in MS 202 to generate and “push” management behavior objects to the appropriate gateway and network elements. EX-1004 9:17-18, 9:66-10:8, 10:21-24; *see also* [1]. *Secer* disclosed multiple gateways (*first adapters*) managing multiple NEs (*plurality of network elements respectively associated with a plurality of first adapters*). EX-1004 FIG. 2; EX-1003 ¶186.

Thus, *Secer* disclosed this limitation. EX-1003 ¶187.

Claim 17

[17] *The system of claim 16, wherein the performance manager component is further configured to deliver at least a subset of the performance data to one or more of the plurality of network elements.*

Secer disclosed and/or rendered this limitation obvious. EX-1003 ¶¶188-190. *Secer* described the *performance manager component is further configured to deliver at least a subset of the performance data to one or more of the plurality of network elements* with reference to claim 16. *See* [16]. EX-1003 ¶188.

As explained above, *Secer* disclosed *performance data from a plurality of network elements*. *See* [16]. This performance data included, e.g., load on NEs

(traffic, CPU usage, workload, etc.) and other operational characteristic information indicating the NE performance. EX-1004 6:41-49; 2:36-61 (performance of network elements included “workload,” available memory capacity, etc.”). As also explained above, *Secer* disclosed MS distributing management behavior objects to gateways and NEs. *See* [16]. A POSITA understood that such management behavior objects would include data from the originating message in, e.g., selecting a new NE to take over from a failed element, to reconfigure an element in response to performance issues, and rerouting traffic in the network by reconfiguring NE. §IV.F.1; EX-1003 ¶189. Such data would include the original configuration parameters or portions thereof, e.g., network element associations, identifications of traffic routes, fault or trap messages (e.g., high CPU utilization, device reboot, interface down, low memory), performance metrics obtained through polling (e.g., workload levels, bandwidth utilization, available storage capacity), and device state information (e.g., operational status, protocol-specific health checks). EX-1003 ¶189.

Thus, *Secer* disclosed and/or rendered this limitation obvious. EX-1003 ¶190.

Claim 18

[18] The system of claim 3, wherein the first communication protocol and the second communication protocol comprise one or more communication protocols associated with at least one of extensible markup language, simple network management protocol, common object request broker architecture, or transaction language 1.

Secer disclosed this limitation for the reasons discussed in claim 6. *See* [6];

EX-1003 ¶191.

Claim 19

[19] The system of claim 3, wherein at least one of the first adapter or the second adapter are executed on one or more virtual machines.

Secer disclosed this limitation for the reasons discussed in claim 13. *See*

[13]; EX-1003 ¶192.

Claim 20

[20] The system of claim 3, wherein the first adapter and the second adapter are configured to communicate with the first application server instance over a remote method invocation interface.

Secer disclosed this limitation for the reasons discussed in claim 14. *See*

[14]; EX-1003 ¶193.

Claim 21

[21] The system of claim 3, wherein the load balancing component is configured to select the first application server instance based on a determination that the first application server instance has a lowest processing load of the plurality of application server instances.

Secer disclosed this limitation for the reasons discussed in claim 5. *See* [5];

EX-1003 ¶194.

Claim 22

[22] The system of claim 3, wherein the plurality of gateway devices comprises a plurality of physically or logically separated gateway devices.

Secer disclosed this limitation for the reasons discussed in claim 8. *See* [8];

EX-1003 ¶195.

VI. MANDATORY NOTICES – 37 C.F.R. §42.8

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

Petitioner certifies that the real party-in-interest in this Petition is Ciena Corporation.

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

1. Judicial Matters

To Petitioner’s knowledge, the ’282 Patent is involved in the following litigation and investigation:

Case Heading	Number	Tribunal	Date
<i>K. Mizra LLC v. Ciena Corporation</i>	NDGA 1:24-cv-05442-SDG	N.D. GA	Nov. 25, 2024

2. Administrative Matters

The '282 Patent is subject to a pending *ex parte* reexamination:

Reexamination No. 90/019,063 (filed May 15, 2025). As of this Petition's filing date and to the best knowledge of Petitioner, the '282 Patent has not been subject to any other administrative proceedings, including any *inter partes* reviews and/or reissues.

3. Related Patents

To the best knowledge of Petitioner, the following U.S. patents and patent applications related to the '282 Patent include U.S. Patent Nos. 9,680,713 and 9,282,010.

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

Lead Counsel	Back-Up Counsel
John M. Baird Reg. No. 57,585 jmbaird@duanemorris.com DUANE MORRIS LLP 901 New York Avenue NW Suite 700 East Washington, D.C. 20001 T: (202) 776-7800	Patrick D. McPherson Reg. No. 46,255 pdmcpherson@duanemorris.com DUANE MORRIS LLP 901 New York Avenue NW Suite 700 East Washington, D.C. 20001 T: (202) 776-7800

	Daniel D. Mitchell Reg. No. 75,226 dmitchell@duanemorris.com DUANE MORRIS LLP 1075 Peachtree Street NE Suite 1700 Atlanta, GA 30309 T: (404) 253-6900
	Paul H. Belnap Reg. No. 73,106 phbelnap@duanemorris.com DUANE MORRIS LLP 901 New York Avenue NW Suite 700 East Washington, D.C. 20001 T: (202) 776-7800
	Stephen J. Smith Reg. No. 80,519 SJSmith@duanemorris.com DUANE MORRIS LLP, 1075 Peachtree Street NE Suite 1700 Atlanta, GA 30309 P: 404-253-6973

In a concurrently filed Power of Attorney, Petitioner has granted Power of Attorney to these practitioners at Duane Morris LLP.

D. Service Information Under 37 C.F.R. §42.8(b)(4)

Service via hand-delivery may be made at the postal mailing address of either lead or back-up counsel. Petitioner consents to service by e-mail.

E. Payment of Fees – 37 C.F.R. §42.103

The required fee is being paid using the Patent Review Processing System.

VII. CONCLUSION

Petitioner requests the Board institute an IPR and cancel the Challenged Claims.

Respectfully submitted,

DUANE MORRIS LLP

Dated: August 31, 2025

/ John M. Baird /

John M. Baird
USPTO Reg. No. 57,585
DUANE MORRIS LLP
901 New York Avenue NW
Suite 700 East
Washington, D.C. 20001

LEAD COUNSEL FOR PETITIONER

CERTIFICATE OF COMPLIANCE WITH WORD COUNT

Pursuant to 37 C.F.R. §42.24 *et seq.*, the undersigned certifies that this document complies with the type-volume limitations. This document contains 13,870 words as calculated by the “Word Count” feature of Microsoft Word 365, the word processing program used to create it.

Pursuant to 37 C.F.R. §42.24(d), this word count excludes the table of contents, table of authorities, mandatory notices under §42.8, certificate of service, certificate of word count, and any claim listing.

Dated: August 31, 2025

/ John M. Baird /

John M. Baird

Reg. No. 57,585

Lead Counsel for Petitioner

CERTIFICATION OF SERVICE ON PATENT OWNER

Pursuant to 37 C.F.R. §§42.6(e), 42.8(b)(4) and 42.105, the undersigned certifies that on August 31, 2025, a complete and entire copy of this Petition for *Inter Partes* Review of U.S. Patent 8,782,282 and all supporting exhibits were served via Federal Express, postage prepaid, to the correspondence address of record for the '282 Patent:

Date of service August 31, 2025

Manner of service FEDERAL EXPRESS

Documents served Petition for *Inter Partes* Review Under 35 U.S.C. § 312 and 37 C.F.R. § 42.104 of U.S. 10,735,282; Petitioner's Exhibit List; All Exhibits; Petitioner's Power of Attorney.

Persons served BRAINSPARK
ASSOCIATES, LLC
2606 W Mesquite St
Chandler, AZ 85224
UNITED STATES

A courtesy copy of the foregoing was also served, via email, on the
following counsel for Patent Owner in the related litigation:

Alden Kwong Wei Lee
alden.lee@foliolaw.com

Cliff Win, Jr.
cliff.win@foliolaw.com

Scott Patrick Amy
s.amy@pkhip.com

Joseph Wendell Staley
j.staley@pkhip.com

Timothy Dewberry
timothy.dewberry@foliolaw.com

/ John M. Baird /

John M. Baird
Reg. No. 57,585

Lead Counsel for Petitioner