UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ServiceNow, Inc.
Petitioner

v.

BMC Software, Inc.
Patent Owner

U.S. Patent No. 8,646,093
Filing Date: December 9, 2009
Issue Date: February 4, 2014

TITLE: METHOD AND SYSTEM FOR CONFIGURATION MANAGEMENT DATABASE SOFTWARE LICENSE COMPLIANCE

DECLARATION OF TAL LAVIAN, PH.D.

Table of Contents

Page

Table of Contents
(continued)

Page

I, Tal Lavian, Ph.D., declare as follows:

1. I have been retained by counsel for ServiceNow, Inc. (Petitioner) in this case as an expert in the relevant art.

2. I have been asked to provide my opinions relating to claims 1, 5, 10-13, and 16 of U.S. Patent No. 8,646,093 to Myers et al. ("the '093 patent"), which I understand is owned by BMC Software, Inc.

3. I have previously submitted a declaration in connection with a Petition for *Inter Partes* Review of the '093 patent, IPR2015-01555 (filed July 3, 2015), in which I provided opinions regarding application of certain prior art references to claims 1, 5, 10-13, and 16 of the '093 patent. My opinions here regarding the level of ordinary skill in the art and the construction of certain terms from the '093 patent are the same as those in my earlier Declaration in IPR2015-01555.

I. BRIEF SUMMARY OF MY OPINIONS

4. Claims 1, 5, 10-13, and 16 generally describe a method and system for ensuring that an organization has deployed a software product in a manner that is consistent with its software license contract. In my opinion, these claims are not patentable because they are directed to an abstract idea.

5. In particular, the challenged claims are directed to the abstract idea of

ensuring that an enterprise is in compliance with its software license contracts. This abstract idea is composed of the abstract ideas of: (1) gathering information about how a software product has been installed in an enterprise, (2) locating the license contract for the product; and (3) comparing the product installations against the license contract to determine compliance or non-compliance. As I will show in **Part IV.B** and **Part VI.A.1** below, this abstract process is performed manually by human beings who create a list of where a software product is installed in an enterprise, and compare that list against the product's license contract.

6. As explained in detail in **Part VI.B**, claims 1, 5, 10-13, and 16 also fail to recite any limitations or technical requirements that could meaningfully transform them into something more than the abstract idea. The claims instead recite technologies that were conventional and routine to persons of ordinary skill in the art, such as the use of a standard Configuration Management Database (CMDB) to store information about software products and software contracts. The bases for my opinions are set forth below.

II. INTRODUCTION AND QUALIFICATIONS

A. Qualifications and Experience

7. I possess the knowledge, skills, experience, training and the education to form an expert opinion and testimony in this case. A detailed record of my professional qualifications, including a list of patents and academic and professional publications, is set forth in my curriculum vitae attached to this declaration as **Exhibit A**.

8. I have more than 25 years of experience in the networking, telecommunications, Internet, and software fields. I received a Ph.D. in Computer Science from the University of California at Berkeley in 2006 and obtained a Master's of Science ("M.Sc.") degree in Electrical Engineering from Tel Aviv University, Israel, in 1996. In 1987, I obtained a Bachelor of Science ("B.Sc.") in Mathematics and Computer Science, also from Tel Aviv University.

9. I am currently employed by the University of California at Berkeley and was appointed as a lecturer and Industry Fellow in the Center of Entrepreneurship and Technology ("CET") as part of UC Berkeley College of Engineering. I have been with the University of California at Berkeley since 2000 where I served as Berkeley Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, and Nortel's Scientist Liaison, where some positions and projects were

done concurrently, others sequentially.

10. I have more than 25 years of experience as a scientist, educator and technologist, and much of my experience relates to computer networking technologies. For eleven years from 1996 to 2007, I worked for Bay Networks and Nortel Networks. Bay Networks was in the business of making and selling computer network hardware and software. Nortel Networks acquired Bay Networks in 1998, and I continued to work at Nortel after the acquisition. Throughout my tenure at Bay and Nortel, I held positions including Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer, and led the development and research involving a number of networking technologies. I led the efforts of Java technologies at Bay network and Nortel Networks. In addition, during 1999-2001, I served as the President of the Silicon Valley Java User Group with over 800 active members from many companies in the Silicon Valley.

11. Prior to that, from 1994 to 1995, I worked as a software engineer and team leader for Aptel Communications, designing and developing mobile wireless devices and network software products. From 1990 to 1993, I worked as a software engineer and team leader at Scitex Ltd., where I developed system and network communications tools (mostly in C and C++).

12. I have extensive experience in the area of network communications and Internet technologies including design and implementation of computer-based systems for managing communications networks, including the ability to monitor and provision networks. While with Nortel Networks and Bay Networks (mentioned above), my work involved the research and development of these technologies. For example, I wrote software for Bay Networks and Nortel Networks Web based network management for Bay Networks switches. I developed Simple Network Management Protocol (SNMP) software for Bay Network switches and software interfaces for Bay Networks' Optivity Network Management System. I wrote software for Java based device management including software interface to the device management and network management for the Accelar routing switch family network management system.

13. I have extensive experience in network communications, including control and management of routing and switching architectures and protocols in layers 1-7 of the OSI model. Much of my work for Nortel Networks (mentioned above) involved the research and development of network communications technologies. For example, I wrote software for Bay Networks and Nortel Networks switches and routers, developed network technologies for the Accelar 8600 family of switches and routers, the OPTera 3500 SONET switches, the

OPTera 5000 DWDM family, and the Alteon L4-7 switching product family. In my lab, I installed, configured, managed, and tested many network communications equipment of competitors such as Cisco Systems, Juniper Networks, Extreme Networks, Lucent and Alcatel.

14. I am named as a co-inventor on more than 80 issued patents and I co-authored more than 25 scientific publications, journal articles, and peer-reviewed papers. Furthermore, I am a Senior Member of the Institute of Electrical and Electronics Engineers ("IEEE").

15. I currently serve as a Principal Scientist at my company Telecomm Net Consulting Inc., where I develop network communication technologies and provide research and consulting in advanced technologies, mainly in computer networking and Internet technologies. In addition, I serve as a Co- Founder and Chief Technology Officer (CTO) of VisuMenu, Inc., where I design and develop architecture of visual IVR technologies for smartphones and wireless mobile devices in the area of network communications. The system is based on cloud networking and cloud computing utilizing Amazon Web Services.

16. Additional details of my background are set forth in my curriculum vitae, attached as **Exhibit A** to this Declaration, which provides a more complete description of my educational background and work experience. I am being

compensated for the time I have spent on this matter. My compensation does not depend in any way upon the outcome of this proceeding. I hold no interest in the Petitioner (ServiceNow, Inc.) or the patent owner (BMC Software, Inc.).

B. Materials Considered

17. The analysis provided in this Declaration is based on my education and experience in the field of computer systems and service management tools, as well as the documents I have considered including U.S. Patent No. 8,646,093 ("'093 patent") **[Ex. 1001]**. The '093 patent states on its face that it issued from an application filed on December 9, 2009. The '093 patent also claims priority to and incorporates by reference U.S. Provisional Application Ser. No. 61/165,505 filed on March 31, 2009, which I have reviewed. ('093, Ex. 1001, 1:9-12; *see also* Ex. 1007 (provisional application).) For purposes of my analysis, I have assumed March 2009 as the priority date for the '093 patent.

18. My Declaration cites the following documents for purposes of describing the relevant state of the art as of March 2009.

19. The prior art documents I rely upon in this Declaration are:

Exhibit	Description of Document
1003	*Best Practice for Software Asset Management*, published by the IT Infrastructure Library, 2003

Exhibit	Description of Document
1004	*Introduction to ITIL*, published by the IT Infrastructure Library, 2005
1005	*A Guide to Software Asset Management*, published by Microsoft Corporation, 2004
1006	Excerpts from Microsoft Computer Dictionary (5th ed. 2002)

III. PERSON OF ORDINARY SKILL IN THE ART

20. I understand that an assessment of claims of the ’093 patent should be undertaken from the perspective of a person of ordinary skill in the art as of the earliest claimed priority date, which I understand is March 31, 2009.

21. In my opinion, a person of ordinary skill in the art as of March 2009 would have possessed at least a bachelor’s degree in computer science (or equivalent degree or experience) with at least four years of practical experience or coursework in the design or development of systems for management of network-based systems and network management databases, such as configuration management databases (CMDBs). Such a person would also have general familiarity with service management tools, software license contracts, and techniques for ensuring compliance with those licenses. As I will explain below, all of these areas were well-developed and mature well before March 2009.

22. My opinions regarding the level of ordinary skill in the art are based on, among other things, my over 25 years of experience in the field of network communications, computer science, and engineering, my understanding of the basic qualifications that would be relevant to an engineer or scientist tasked with investigating methods and systems in the relevant area, and my familiarity with the backgrounds of colleagues and co-workers, both past and present.

23. Although my qualifications and experience exceed those of the hypothetical person having ordinary skill in the art defined above, my analysis and opinions regarding the ’093 patent have been based on the perspective of a person of ordinary skill in the art as of March 2009.

IV. STATE OF THE ART OF THE RELEVANT TECHNOLOGY AT THE TIME OF THE ALLEGED INVENTION

24. The ’093 patent generally discloses a method and system for monitoring compliance with software license contracts. In this section, I provide a brief background of the state of software license contract compliance technology prior to March 2009 pertinent to the ’093 patent.

A. Software and Software License Contracts

25. By March 2009, computer software had become a common fixture of everyday life, and integral to the functioning of most business enterprises. Computer software has long been made available to customers pursuant to a contract known generally as a “software license,” which governs the customer’s use of the software. A software license may specify, among other things, the number of users within an enterprise who are permitted to use the licensed software (including the number of users who may use the software concurrently), or how long the licensed software may be used before the license must be renewed. Software license contracts can also be one component of a larger service contract with the provider, and thus, can include a number of complex provisions.

26. It is important for a number of reasons for an enterprise to comply with the terms of software license contracts. Violating a software license contract could not only expose the enterprise to liability for breach of contract, but in some

cases, civil or criminal liability for copyright infringement. Although violations of software license contracts are often unintentional, those violations may still result in significant consequences. Larger enterprises may have hundreds of software products governed by different software license contracts, each presenting a large number of potentially complex licensing terms.

B. Managing Compliance with Software License Contracts

27. The software industry responded to the concerns expressed above by developing processes to monitor and manage compliance with software license contracts. Many of these processes were described in *Best Practice for Software Asset Management* (2003), a well-known publication in the field published by the IT Infrastructure Library ("ITIL"). (Ex. 1003 ("Best Practice").) ITIL is a division of the Office of Government Commerce of the United Kingdom government. (*Id.* at 1 (under "The IT Infrastructure Library").) The publications disseminated by ITIL, such as Best Practice, provide industry standards and practices for managing IT assets. As its name implies, the Best Practice publication defines a set of preferred processes for managing software assets. (Best Practice, Ex. 1003, at p. xi (stating that ITIL "is the most widely accepted approach to IT Service Management in the world.").)

28. ITIL's standards are often considered authoritative by many persons

of ordinary skill in the art. The specification of the '093 patent, in fact, specifically cites and discusses Best Practice. ('093, 1:51-2:8.) In fact, the entire Background section of the '093 patent is devoted to discussing and discussing ITIL-defined processes for enterprise and software asset management. ('093, 1:18-2:8.)

29. The overall process for ensuring license compliance is conceptually simple and little more than common sense: an organization must (1) obtain an inventory of the software products actually installed or deployed in the organization, (2) gather the licensing contracts governing those software products; and then (3) compare the installation of the software products against the governing license contracts to identify instances, if any, of non-compliance with the software license contracts.

30. Best Practice generally describes steps (1) and (2) of this process as "verification and audit," and provides detailed guidance on how to carry out those steps. (Ex. 1003, at p. 51, § 5.4.1.) "The objective of the verification and audit process is to ensure that the SAM [Software Asset Management] records accurately reflect what is actually held, and that appropriate corrective actions are undertaken when discrepancies are identified." (*Id.*) This involves obtaining an inventory of the software actually in use, and identifying the software license contracts and verifying that they are operative and authentic. (*Id.*) "Verifying the

integrity of records of licenses held is a manually intensive activity." (*Id.*)

31. Best Practice describes step (3) of the process above as "license compliance." (*Id.*, p. 52, § 5.4.2.) "Licensing compliance processes are responsible for ensuring that the use of all software within the organisation remains within all legal and contractual terms and conditions." (*Id.*) This includes identification of any "exception conditions and non-conformances relating to the use of unlicensed software," and the "correction and prevention of licensing shortfalls and the identification and highlighting of software overuse and redundancy situations." (*Id.*; *see also id.*, p. 14 ("Perform regular reconciliations of (a) what is actually installed against (b) what is recorded against (c) licences owned (for licensed software), and resolve any identified exceptions promptly.").)

32. This process can be done – and in fact is <u>commonly done – by human beings using pen and paper</u>. For example, Microsoft Corporation published a set of guidelines for verifying software contract compliance in 2004, *A Guide to Software Asset Management*. (Ex. 1005, at 017 (back page showing 2004 document date).) The first step of ensuring license compliance, according to Microsoft, is to perform an inventory of installed software. "You can perform a manual inventory simply by going to each PC and viewing the Add or Remove Programs screen. This will tell you exactly what programs are running on that

particular PC." (Ex. 1005, at p. 12.) The user can alternatively use a software inventory tool to automate the process of taking a software inventory. (*Id.*)

33. For the second step, Microsoft advises the user to "[l]ocate the licensing documentation for each software program your organisation currently holds," and "[o]nce you have collected all of your company's license documentation, record the information in a report." (*Id.* at p. 14.) "Then compare this report with the software inventory report you prepared in SAM Step 1." (*Id.*) Microsoft even suggests that the user record his or her findings by creating a table listing the software products, total number of installations, and total licenses owned, to calculate the excess or deficiency. (*Id.* at p. 14.) "If your company is over-licensed, software assets are going to waste." (*Id.* at p. 15.) But "[i]f your company is under-licensed, now is the time to acquire additional licenses through an authorised Microsoft software reseller." (*Id.*)

C. Using a Configuration Management Database (CMDB) to Manage Compliance with Software Licenses

34. As the number and variety of software licenses increased, industry recognized a need for a more automated and computer-assisted approach to record information about software products and their corresponding and software license contracts. One solution was to use a database known as a **"Configuration Management Database**" or "**CMDB**," to keep track of those licenses. Generally

speaking, "CMDB" is an industry-standard term referring to a database that stores information about the Information Technology ("IT") assets used by an enterprise, such as servers, workstations, software programs, documentation, and other computing resources. The CMDB contains a series of records, known as "**configuration items**" or "**CIs**," for storing information about the IT assets.

35. A CMDB is, at a conceptual level, little more than a database to keep track of components in an enterprise. As explained in *Introduction to ITIL*, published in 2005, that database need not even be computer-based:

> **Configuration Management Database (CMDB)**
> All CI's are included in the Configuration Management Database (CMDB). The CMDB keeps track of all IT components, their versions and status and the relationships between them. <u>In its most basic form, a CMDB could consist of paper forms or a set of spreadsheets</u>.

(Introduction to ITIL, Ex. 1004, § 6.1.1, at p. 58 (underlining added).)

36. The Background of the '093 patent acknowledges that CMDBs and CIs are not an invention of the patent and were, at the time of filing, "emerging as a prominent technology for Enterprise Management software." ('093, 1:24-26, 1:18-42.) "The CMDB serves as a point of integration between various IT management processes," including software asset management, which is a "core component of an overall asset management policy." ('093, 1:39-40; 1:49-51.) The

patent acknowledges that "[o]ne kind of CI that may be managed in a CMDB is a software asset." ('093, 1:43-44.)

37. Many of the techniques for managing compliance with software contracts using CMDBs were described in Best Practice. As noted, the Background section of the '093 patent specifically cites Best Practice and discusses ITIL CMDBs and ITIL-defined processes for management of software assets. ('093, 1:18-2:8.) For example, Best Practice provides an exemplary CMDB/CI schema for storing information about the organization's software assets and their corresponding software license contracts. (Ex. 1003, at pp. 119-23, Appendix D.)

V. THE '093 PATENT'S TECHNIQUE FOR SOFTWARE LICENSE COMPLIANCE

A. The Specification

38. The '093 patent generally describes a method and a system to "monitor and verify software license compliance in an enterprise." ('093, Ex. 1001, 2:66-67.) The patent generally describes a technique for managing software license compliance by (a) modeling the deployment of software products in an enterprise, and (b) comparing that deployment against the software license contracts for those products. ('093, Ex. 1001, Abstract.)

39. Figure 2 provides a general overview of one embodiment of the computer-implemented method and system described in the '093 patent:

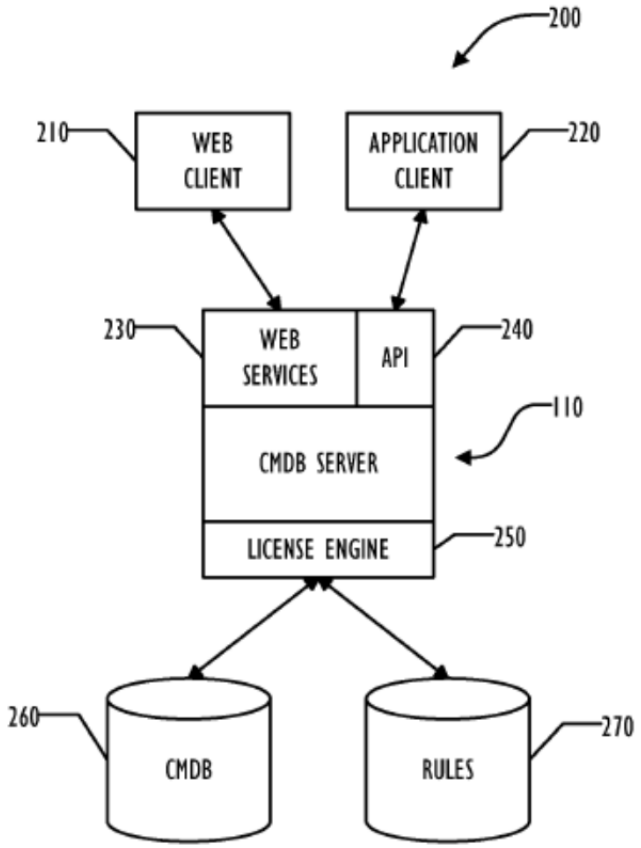


FIG. 2

(’093, Fig. 2.)

40. Figure 2 includes management system **200** that has a Configuration Management Database (“CMDB”) **260**. As explained in the Background and noted above, the CMDBs were well-known and “a prominent technology for Enterprise Management Software.” (’093, 1:18-26.) A CMDB “contains data about managed resources known as Configuration Items (CIs).” (’093, 1:29-30.).

41. The patent uses these configuration items (CIs) to store information

about software products and their associated software contracts. "Information about the software contracts is stored as CIs in the CMDB datastore **260** [of Figure 2] using one or more of the clients **210**/**220**." ('093, 5:1-3.)

42. The system in Figure 2 also includes a license datastore **270**. Although shown as a separate database in Figure 2 above, license datastore **270** "may be integrated with the CMDB datastore **260**." ('093, 4:11-13.) "The license datastore **270** provides storage for [sic] to model software contracts, including rules against which the CIs are evaluated for software license compliance and other information necessary for processing those rules." ('093, 4:13-17.) The '093 patent identifies at least two types of information used to evaluate compliance with a software license contract: (1) information about the **software license contract**, and (2) a **license certificate** corresponding to the license.

43. First, configuration information about a **software license contract** may be stored in the CMDB, including the term of the license, its current status (draft, executed, expired, etc.), the company, and other information associated with the contract. ('093, 5:10-56 (Table 1).) The license database may also store a number of pre-defined "license types" that can be essentially used as templates in identifying the characteristics of a software license contract. ('093, 6:1, 6:33-35.) Exemplary "license types" include "enterprise," "site," and "per instance" software

licenses (among others), each having certain pre-defined characteristics. ('093, 6:40-55 (Table 2).) The user can also create new license types, if needed. ('093, Fig 4 (step 420).) As shown in **Part V.B** below, this information relating to the software license contract is part of the "**first model**" recited in the claims.

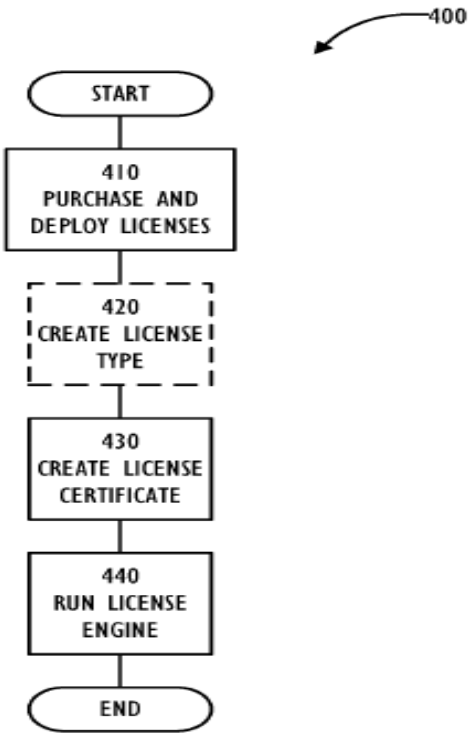
44. Second, the '093 patent describes a "**license certificate**" that is linked to its corresponding software license contract. ('093, 3:1-2.) "A license certificate indicates the right to deploy software in the environment managed by the CMDB server **110**." ('093, 8:59-63.) A "license certificate" may comprise a variety of information relating to the right to use the software, including the license category (client, server, mainframe), effective date and expiration date, among other information. ('093, 9:1-20 (Table 3).) The system may ask the user to enter additional information such as "how many licenses were purchased and how many copies per device are allowed under each license." ('093, 9:35-36.) "Other questions may be asked depending typically on the license type. The additional information supplied in response to those questions may be included in the license certificate as it is stored in the license datastore **270**." ('093, 9:36-46.) This license certificate information is part of the "**second model**" in the claims.

45. Once these two categories of information – **software license contract** and **license certificate** information – are compiled, the system uses them to

evaluate the status of the licenses:

> Returning to FIG. 4, after the license certificates are created, then in block **440**, the license engine **250** is run. The license engine evaluates the status of the software licenses modeled in the CMDB **260** against the license certificates created in block **430**.

('093, 10:28-32; *see also id.*, 4:63-65 ("FIG. **4** is a flowchart illustrating a technique **400** for performing software license compliance monitoring and verification . . .").)



('093, Fig. 4.)

46. One exemplary process for comparing the licenses against the certificate is shown in Figure 5. ('093, Fig. 5.) That process involves, among

other steps, evaluation of "compliance rules" in order "to determine whether each of the software CIs complies with the terms of the software contract. In block **560**, if any CI is not in compliance, then any desired exception processing may be performed." ('093, 10:49-53.) This exception processing "in one embodiment may be simply producing an error message or report indicating the exception." ('093, 10:55-56.) "Where the organization is not in compliance, the license engine identifies the non-compliance and provides information that may allow the contract or asset manager to address the problems and bring the organization into compliance." ('093, 13:20-24.) "For example, in one embodiment, a noncompliance exception indicating that more instances of a particular software are deployed than are licensed may automatically trigger a request to purchase sufficient additional license to bring the enterprise **100** back into compliance." ('093, 10:65-11:3.)

B. The Challenged Claims of the '093 Patent

47. The two independent claims addressed in this Declaration—i.e., claims 1 and 16—purport to recite a method and a system, respectively, for managing software license contracts. The first independent claim addressed in this Declaration is claim 1, which recites:

1. A computer-implemented method, comprising:

[a] modeling deployment of a software product and a software

license contract for the software product;

[b] storing a first model of the modeled deployment of the software product in a configuration management database (CMDB) by storing information related to the software product as a first configuration item in the CMDB and by storing information related to the software license contract as a second configuration item in the CMDB;

[c] storing a second model of the modeled software license contract for the software product in a license database by generating a license certificate corresponding to the software license contract and storing the license certificate in the license database; and

[d] evaluating the deployment of the software product for compliance with the software license contract, comprising:

[d][1] connecting and comparing the first model and the second model by comparing the first configuration item with the license certificate and connecting the license certificate with the second configuration item responsive to comparing the first configuration item with the license certificate; and

[d][2] generating an exception indication if the act of comparing the first model and the second model indicates non-compliance with the software license contract.

('093, 13:44-14:3 (Claim 1).) I added bracketed notations (e.g., "[a]," "[b]," etc.) for the purpose of identifying these limitations in my Declaration. Claims 5, 10, 11, 12, and 13 depend from independent claim 1 listed above. I address each claim

in more detail in **Part VI** below. The second independent claim addressed in this Declaration is claim 16, which generally recites a system claim for performing the method of independent claim 1 discussed above.

C. Claim Construction

48. I have been informed by counsel that invalidity analysis is a two-step process. In the first step, the scope and meaning of a claim is determined by construing the terms of that claim. In the second step, the claim as interpreted is compared to the prior art. Therefore, before I address the application of the prior art to the claims of the ’093 patent in **Part VI** below, I provide constructions for certain terms in those claims.

49. I have also been informed by counsel that a claim in an unexpired patent subject to CBM review must be given its “broadest reasonable construction in light of the specification of the patent in which it appears,” which is different from the manner in which the scope of a claim is determined in litigation. I apply the “broadest reasonable construction” standard in my analysis below.

1. “license certificate”

50. The term “license certificate” is recited multiple times in independent claim 1. In my opinion, the broadest reasonable construction of “license certificate” to one of ordinary skill in the art is “**information relating to the right**

to deploy software." I derive this definition from the following passage of the specification:

> After any new license types are created to handle the terms of the new software contracts terms, license certificates may be created in block **430**, to link software contracts to CIs. A license certificate indicates the right to deploy software in the environment managed by the CMDB server **110**. In one embodiment, a license certificate comprises the information listed in Table 3 below.

('093, 8:59-65 (underlining added).) Table 3 in the specification provides an exemplary license certificate that includes a number of fields or attributes including a summary description of the certificate, the expiration date of the license, and other fields. ('093, 9:1-20 (Table 3, reproduced below).)

TABLE 3

License Certificates	
Field name	Description
Company	This information comes from the software contract.
Software Contract ID	
Certificate ID	The certificate ID identifies the license certificate in listings and reports. It does not have to be unique.
Summary	This field provides additional space to describe the certificate.
Status	Initially set to Draft.
License Category Type	Select from Client, Server, or Mainframe.
License Type	The appropriate license type. The license type determines the connection questions and the compliance questions.
Cost Center	This information comes from the software contract, but can be changed.
Effective Date	The date that the license becomes effective.
Expiration Date	The date that the license expires. If the license does not expire, this field may be left blank.

51. Based on my analysis of the specification, I believe the broadest reasonable construction of "license certificate" to one of ordinary skill in the art is "**information relating to the right to deploy software**," which is derived from the underlined sentence quoted above. ('093, 8:61-63 ("A license certificate indicates the right to deploy software in the environment managed by the CMDB server **110**.") (underlining added).) I note that my construction removes a portion of that sentence, "in the environment managed by the CMDB server **110**," because claim 1 does not recite a server, let alone a "CMDB server." Although the claims do recite a CMDB, they do not require any CMDB server.

2. "model" and "modeling"

52. The terms "model" and "modeling" are used throughout the claims addressed in this Declaration. Claim 1, for example, recites the step of "modeling deployment of a software product and a software license contract for the software product," and storage of a "first model" and a "second model" containing information about a software product and its software license contract. The broadest reasonable construction of "**model**" is "**an organized collection of information about an object**," and in the verb form "**modeling**," that simply refers to "**creating a model**."

53. The specification does not define "model" but provides examples of how a software license contract can be modeled. ('093, 2:13-17.) For example, the specification explains that "the CMDB server **110** may model the software product packages or components installed on each of the computer systems **120**, as well as the software contracts under which that software is licensed." ('093, 3:42-45 (underlining added).) Table 1 of the '093 patent provides an illustration of such a model. ('093, 5:3-5.) It includes a number of fields for a particular software license including the ID, Summary, Term, Status, Expiration Date, and other fields. ('093, 5:10-57.) This example shows that creating a "model" of an object, such as a software license contract, involves storage of an organized collection of

information about that object. Although the term "model" in other contexts has been used to describe a mathematical or graphical representation of an object, the specification does not require any such representation.

54. In my opinion, therefore, the broadest reasonable construction of "**model**" is "**an organized collection of information about an object**," and the verb "**modeling**" means "**creating a model**."

55. The following table identifies the above claim constructions:

Claim Term	Broadest Reasonable Construction
"license certificate"	"information relating to the right to deploy software"
"model"	"an organized collection of information about an object"
"modeling"	"creating a model"

VI. OPINIONS REGARDING PATENT-ELIGIBLE SUBJECT MATTER

56. I have been asked to provide my opinion as to whether claims 1, 5, 10-13, and 16 of the '093 patent are (1) directed to an abstract idea, and (2) if so, whether the claims provide meaningful additional elements that transform them into something more than the abstract idea itself. In my opinion, the answer to the first question (1) is "yes," but the answer to the second question (2) is "no."

57. I am informed by counsel that patent claims that cover an "abstract idea" are not eligible for patent protection. I am informed by counsel that to determine if an alleged invention of a claim is an ineligible abstract idea, two

inquiries must be performed. First, it must be determined if the claim is directed to an abstract idea. This requires that the concept embodied in the claim be characterized or defined.

58. Second, if it is determined that the claim is directed to an abstract idea, it must be determined if the elements of the claim as a whole contain an inventive concept or meaningful limitations sufficient to transform the claimed abstract idea into a patent-eligible invention. I am further informed that claim limitations that merely recite well-understood, routine, and conventional activities, such as the use of generic computer components or technology, do not add sufficiently meaningful limitations to distinguish the claim from the abstract idea itself. With these principles in mind, I turn to claims 1, 5, 10-13, and 16.

A. Are Claims 1, 5, 10-13, and 16 Directed to an Abstract Idea?

1. Claim 1

59. Claim 1 is directed to the abstract idea of ensuring that an enterprise is in compliance with its software license contracts. This broader abstract idea is composed of the following three abstract ideas:

(1) gathering information about how a software product has been installed in an enterprise,

(2) locating the license contract for that product; and

(3) comparing the product installations against the license contract to determine compliance or non-compliance with the license.

This idea derives directly from the language and structure of claim 1:

1. A computer-implemented method, comprising:

[a] modeling deployment of a software product and a software license contract for the software product;

[b] storing a first model of the modeled deployment of the software product in a configuration management database (CMDB) by storing information related to the software product as a first configuration item in the CMDB and by storing information related to the software license contract as a second configuration item in the CMDB;

[c] storing a second model of the modeled software license contract for the software product in a license database by generating a license certificate corresponding to the software license contract and storing the license certificate in the license database; and

[d] evaluating the deployment of the software product for compliance with the software license contract, comprising:

[d][1] connecting and comparing the first model and the second model by comparing the first configuration item with the license certificate and connecting the license certificate with the second configuration item responsive to comparing the first configuration item with the license certificate; and

[d][2] generating an exception indication if the act of comparing the first model and the second model indicates non-compliance with the software license contract.

('093, Ex. 1001, 13:44-14:3 (Claim 1).)

60. The first step **[a]** in claim 1, "**modeling deployment of a software product and a software license contract for the software product**," is simply a restatement of the first and second parts of the abstract idea identified above – gathering information about how a software product has been installed (or "deployed") in an enterprise, and locating the license contract for that product. In fact, the idea behind the modeling step of claim 1**[a]** <u>could be performed by a human being using pen-and-paper</u>. For example, a system administrator could walk from computer-to-computer and write down an inventory of where the software product is installed, and then gather the governing software license contract. As I explained in **Part IV.B** above, the Microsoft Corporation publication, *A Guide to Software Asset Management*, describes such a technique. (Ex. 1005, at pp. 12-15.)

61. Microsoft provides a further example in which the administrator can generate a written deployment model by creating a table (below) listing the installed software, the number of installations, the number licenses owned, etc. (*Id.* at p. 14.)

Software Installed			Total Licenses Owned	License Excess of Deficiency
Software Program	Version	Total Installations		
Microsoft® Office	XP	14	12	-2
Microsoft® Office	2000	12	12	0
Microsoft® Windows	XP®	12	10	-2
Microsoft® Windows®	2003	37	40	+3

(Ex. 1005, at p. 14.)

62. The second step **[b]** of claim 1 is directed at little more than the storage of the deployment information from step 1**[a]**. Step **[b]** requires that this deployment information be stored in configuration items (CIs) in a Configuration Management Database (CMDB), but as I will show in the analysis in **Part VI.B** below ("Do Claims 1, 5, 10-13, and 16 Provide Meaningful Limitations?"), the storage of such information in CIs of a CMDB was a well-known and routine practice before the '093 patent was filed.

63. In fact, the ITIL literature confirms that a CMDB as recited in claim 1**[b]** is itself <u>an abstraction that does not necessarily require a technical implementation.</u> As explained in *Introduction to ITIL*, published in 2005:

> **Configuration Management Database (CMDB)**
> All CI's are included in the Configuration Management Database (CMDB). The CMDB keeps track of all IT components, their versions and status, and the relationships between them. <u>In its most basic form, a CMDB could consist **of paper forms** or a set of</u>

<u>spreadsheets</u>.

(Introduction to ITIL, Ex. 1004, § 6.1.1, at p. 58 (emphasis added).) Like the preceding step **[a]**, the storage of information in CIs of a CMDB as recited in step **[b]** merely reflects the abstract idea of gathering and storing information about the installation of a software product and its corresponding license contract. This is not affected by the fact that the information is stored in CIs of a CMDB.

64. This abstract idea is further expressed in step 1**[c]**, which recites the creation of a "second model" that includes "a license certificate corresponding to the software license contract and storing the license certificate in the license database." The term "license certificate," as explained in **Part V.C.1** above, simply refers to "information pertaining to the right to deploy software," and the claim language does not prescribe any particular way in which the certificate must be represented.

65. Because the license certificate information corresponds to the software license contract, step 1**[c]** is simply another limitation expressing the abstract idea of gathering and storing information about software license contracts.

66. Turning to step 1**[d]**, that step and its sub-steps speak to the other part of the abstract idea that I identified above – using the information gathered in the preceding steps to ensure that the software product has been installed (deployed) in

a manner that does not violate its software license contract. This abstract idea is recited in claim 1**[d]** itself, "evaluating the deployment of the software product for compliance with the software license contract."

67. The sub-steps of claim 1**[d]** recite the unremarkable requirement that the evaluation takes place by comparing information about the software product (the "first configuration item" in the "first model") with the license certificate (the "second model"), and generating "an exception indication if comparing the first model and the second model indicates non-compliance with the software license contract." ('093, 13:59-14:3 (claim 1**[d]**).) <u>This step could be performed by a human being using nothing more than pen and paper</u>, as illustrated in *A Guide to Software Asset Management* published by Microsoft, which advises the user to create a table or spreadsheet to determine if the enterprise has licensed too many, or too few, copies of the software product:

Software Installed			Total Licenses Owned	License Excess of Deficiency
Software Program	Version	Total Installations		
Microsoft® Office	XP	14	12	-2
Microsoft® Office	2000	12	12	0
Microsoft® Windows	XP®	12	10	-2
Microsoft® Windows®	2003	37	40	+3

(Ex. 1005, at p. 14.)

68. The evaluation and comparison process described in claim 1**[d]** <u>could</u>

be carried out by a human being using such a table. Using the "**Microsoft Office**" product listed above as an example, the user has compared the "**Total Installations**" number (the "first configuration item" in the "first model") against the "**Total licenses Owned**" number (license certificate), which revealed that the organization has two unlicensed installations of Microsoft Office. The user recorded this as a "-2" in the "**License Excess of [sic; or] Deficiency**" column.

69. Finally, claim 1**[d]** recites the step of "generating an exception if the act of comparing the first model and the second model indicates non-compliance with the software license contract." This similarly adds nothing concrete to claim 1. It is simply directed to the abstract idea of calling attention to the fact that the comparison uncovered non-compliance with the software license contract. The Microsoft publication similarly discloses that "[i]f your company is under-licensed, now is the time to acquire additional licenses through an authorised Microsoft software reseller." (*Id.* at p. 15.) This last step of claim 1 is simply part-and-parcel of the abstract idea of ensuring that the deployment of a software product does not violate its software license contract.

2. Claim 5

70. Claim 5 depends from claim 1 and adds the requirement that "the act

of evaluating is performed on demand." This further limitation does not change my opinion that claim 5 is directed to an abstract idea. This claim specifies when the claimed evaluation may occur ("on demand"), but does not specify how the evaluation in claim 1 is carried out. The claimed step of "evaluating" from claim 1 remains an abstract idea regardless of whether it is performed on demand.

3. Claim 10

71. Claim 10 depends from claim 1 and recites that the claimed act of evaluating compliance further comprises "indicating a suggested action for achieving compliance if the act of comparing the first model and the second model indicates non-compliance with the software license contract." This further limitation adds nothing to the abstract nature of the idea to which it is directed. There is nothing concrete or non-abstract about offering a suggestion to remedy the non-compliance with the software license contract.

72. For example, the specification explains that one such "suggested action" is to purchase additional software licenses in order bring the organization into compliance with the software license contract. ('093, 10:65-11:3.) This is precisely the type of action that <u>can be performed manually by a human being</u>, as expressly suggested by the Microsoft publication. (*See* Ex. 1005, at p. 15 ("If your

company is under-licensed, now is the time to acquire additional licenses through an authorised Microsoft software reseller.”).)

4. Claims 11-13 (“License Type” Claims)

73. Claims 11-13 depend from claim 1 and recite the additional features of receiving a selection of a “license type.” None of these claims change the nature of the overall abstract idea ensuring that an enterprise is in compliance with its software license contracts.

74. Claim 11 recites the step of “receiving a selection of a license type corresponding to the software license contract; and receiving license contract data corresponding to the selected license type.” This step does not, in my opinion, alter the fact that the claim is directed to an abstract idea.

75. The ability to select a license type, and receive license contract data corresponding to that selection, is simply a refinement to the abstract idea of collecting information about the software license contract. <u>There is nothing about this step that requires any particular technical implementation</u>.

76. Moreover, it was well-known to persons of ordinary skill in the art that there are many different types of software license contracts. Best Practice, for example, provides four pages and two entire sections to describing the many

different types of licenses that may exist, including software license contracts based on broad categories including duration of use (e.g. perpetual vs temporary), measure of usage (e.g. per copy, concurrent usage, enterprise-wide), and numerous other categories. (Best Practice, Ex. 1003, at pp. 106-09, §§ B.2, B.3.)

77. Claim 12 depends from claim 11 and further recites that "the act of generating the second model further comprises: providing a plurality of pre-determined license types." This claim adds very little to claim 11 for purposes of my analysis. Predetermined license types were also well-known to persons of ordinary skill in the art, including "Perpetual," "Temporary," "Per Copy," license contracts. (Best Practice, Ex. 1003, at pp. 106-09, §§ B.2, B.3.)

78. Claim 13 depends from claim 11 and recites that "the act of generating the second model further comprises: allowing a user to define a custom license type." Like claim 11 and 12, this is simply a further refinement on the ability to collect information about the software license contract, which is simply part of the abstract idea.

5. Claim 16

79. Claim 16 is a system claim that recites a server computer that performs the steps of claim 1:

> 16. A system, comprising:
>
> a server computer, comprising:
>
> a processor;
>
> a configuration database, coupled to the processor;
>
> a license database, coupled to the processor; and
>
> a program store, coupled to the processor, on which is stored instructions for the processor, <u>wherein the instructions cause the processor to perform the method of claim 1</u>.

80. Claim 16 is directed to the same abstract idea as claim 1. As I will show below, the recitation of a "server computer," "processor," a "configuration database," a "license database," and a "program store," are not meaningful limitations or technical requirements. They are simply generic computer components that were well-known and routine to skilled artisans.

B. Do Claims 1, 5, 10-13, and 16 Provide Meaningful Limitations?

81. In my opinion, the limitations of claims 1, 5, 10-13, and 16 as a whole do not contain any inventive concept or meaningful limitation sufficient to transform them into something more than the abstract idea ensuring that an

enterprise is in compliance with its software license contracts.

1. Claim 1

82. Beyond the abstract idea to which it is directed, the limitations recited in claim 1 recite activities that were well-understood, routine, and conventional to a person of ordinary skill in the art.

83. Many of the claim steps, in fact, do not inherently require a computer to perform. Step 1**[a]**, for example, recites "modeling deployment of a software product and a software license contract for the software product." <u>This step could be carried out manually by a human being</u>, as explained above, by manually taking an inventory of installations of the software product and collecting its license. (See Ex. 1005, at pp. 12-15.) Nevertheless, to the extent the claim requires that this particular step be computer-implemented, the specification makes clear that the alleged invention can be carried out using generic server hardware, such as an Intel-based computer with standard components such as a keyboard, mouse, display, network interface, and storage device. ('093, Fig. 3, 4:21-59.)

84. Step 1**[b]** similarly recites no meaningful limitations that distinguish the claim from the abstract idea. As I explained above, it is not clear if storage of the deployment data in CIs of a CMDB, as recited in step 1**[b]**, even requires a computer. As explained in *Introduction to ITIL*, published in 2005:

> **Configuration Management Database (CMDB)**
>
> All CI's are included in the Configuration Management Database (CMDB). The CMDB keeps track of all IT components, their versions and status and the relationships between them. In its most basic form, a CMDB could consist of **paper forms** or a set of **spreadsheets**.

(Introduction to ITIL, Ex. 1004, § 6.1.1, at p. 58 (emphasis added).)

85. But even the claim required that the storage in step 1**[b]** be computer-implemented, the idea of storage of such information in a CMDB was routine, generic, and conventional by the time the application for the '093 patent was filed. In fact, the Background of the '093 patent expressly acknowledges that CMDBs are in the prior art, and describes them as "emerging as a prominent technology for Enterprise Management software." ('093, 1:18-42, 1:24-26.) The patent further acknowledges that "[o]ne kind of CI that may be managed in a CMDB is a software asset." ('093, 1:43-44.)

86. In fact, the specification goes out of its way to emphasize that it is not providing any particular technical limitation with respect to storage of information in a CMDB. (''093, 3:46-49 ("[T]he techniques for discovering and storing the information modeled by CIs for hardware and software components in the CMDB server **110** is outside the scope of the present invention.") (underlining added).) The storage step in claim 1**[b]** is simply a reference to existing CMDB techniques,

not a description of any particular technical implementation of a CMDB.

87. Best Practice further confirms that step 1**[b]** was at best nothing more than routine and generic technology. Best Practice explains that "each component contained within the CMDB is referred to as a Configuration Item (CI). Each CI record within the CMDB contains all of the attributes and information relating to a component necessary for managing it, whether software, hardware, contracts, etc." (Best Practice, Ex. 1003, at pp. 81-82 (underlining added).) **Appendix D** of Best Practice provides an example of a CMDB with CIs for storing information about the deployment of software products and their corresponding licenses, mirroring the requirements of claim 1**[b]**. (*Id.* at pp. 119-25.)

88. In particular, **Table D.2** in Best Practice discloses an "Installed software inventory" CI with more than a dozen attributes about the software products installed in the enterprise, including "Product," "Version," "Installation status," and "Licence used," the latter attribute linking to the license that governs use of the software. (*Id.* at 123.) The second table shown in **Table D.1** of Appendix D provides an exemplary a "Software licence inventory" CI that includes more than 40 attributes relating to software licenses, including "Licence status and counts" (*id.* at 119), "Licensing basis" such as "per PC," "per device," "per user," etc. (*id.* at 122), "Transferability" information about the license (*id.*),

"Source references" to specific the documentation of the licensing terms (*id.*), and many other attributes. These tables from Best Practice, published several years before the '093 patent filing date, confirm that the use of a CIs in a CMDB to store information about software products and their corresponding software license contracts was a generic, routine and conventional activity to skilled artisans.

89. Nor do I see anything in claim 1**[c]** that goes significantly beyond the abstract idea. This limitation recites generation of another piece of information, a "second model" with a "license certificate" corresponding to the software license contract, stored in a "license database." The claim does not require any particular programming or technical implementation to accomplish this step. One of ordinary skill in the art would have understood that this step could be carried out using conventional computer programming techniques.

90. The patent also makes clear that the claimed "license database" in claim 1**[c]**, which is referred to as a "license datastore" in the specification, can be implemented using conventional database techniques such as the standard structured query language (SQL). ('093, 4:17-20 ("The CMDB datastore **260** and the license datastore **270** may be implemented as a collection of flat files, a structured query language database, or in any other way desired.").) The term "structured query language" (SQL) refers to a well-known and standard technique

for accessing and updating information in databases. (*See* Microsoft Computer Dictionary (5th ed. 2002), Ex. 1006, at p. 501 ("**structured query language** *n.* A database sublanguage used in querying, updating, and managing relational databases—the de facto standard for database products.") (underlining added; bold and italics in original).) Claim 1**[c]** simply relies on routine and generic database technology, and requires no special programming or particular implementation.

91. Claim 1**[d]** similarly does not recite anything that meaningfully distinguishes the claim from the abstract idea. The step of "evaluating the deployment of the software product for compliance with the software license contract," is simply an abstract idea as noted above. This portion of the claim does not place any restrictions on how the claimed evaluation takes place.

92. Substep 1**[d][1]** continues by reciting the step of "connecting and comparing the first model and the second model by comparing the first configuration item with the license certificate . . ." This limitation merely recites step of comparing information about the software product ("first configuration item") against information about its software license contract ("license certificate"). The claim imposes no limitations on how the comparison takes place. This is simply a restatement of the abstract idea described above of comparing the software installation against the license contract to determine compliance.

93. This substep continues by reciting "connecting the license certificate with the second configuration item responsive to comparing the first configuration item with the license certificate." The claim again provides no restriction on how the claimed "connecting" takes place; the claim under its broadest reasonable construction would appear to encompass any way in which the license certificate and the second CI are joined or linked.

94. Finally, claim 1**[d][2]** recites the step of "generating an exception if the act of comparing the first model and the second model indicates non-compliance with the software contract." As noted above, this is simply part of the abstract idea of ensuring that the enterprise is in compliance with the license contract. The specification makes clear that something as simple as "producing an error message or report indicating the exception" may suffice. ('093, 10:54-56.) The claim does not specify a particular way the exception must be generated.

2. Dependent Claims 5, and 10-13

95. As I explained in my discussion in **Part VI.A** above, these dependent claims add minor features to the method of claim 1. Claim 5 requires that the act of evaluation be performed "on demand," claim 10 adds the ability to indicate a "suggested action" to achieve compliance, and claims 11-13 pertain to the ability

to specify a "license type" for the software license contract. None of these claims requires any particular programming or technical implementation, and are simply trivial details for implementing the abstract idea of ensuring that an enterprise is in compliance with its software license contracts.

3. Independent Claim 16

96. Claim 16 is a system claim that simply surrounds the steps of claim 1 with generic and conventional computer equipment:

> 16. A system, comprising:
>
> a server computer, comprising:
>
> a processor;
>
> a configuration database, coupled to the processor;
>
> a license database, coupled to the processor; and
>
> a program store, coupled to the processor, on which is stored instructions for the processor, <u>wherein the instructions cause the processor to perform the method of claim 1</u>.

97. None of the components recited in claim 16 provides a <u>meaningful distinction from the abstract idea</u>. The specification confirms that each of those components is generic. For example, the "processor" could be "any programmable controller device," including standard Intel-based microprocessors commonly

found in servers and personal computers. ('093, 4:44-50.)

98. Nor is there anything non-generic about the claimed configuration database and license database. The patent makes clear that these databases (referred to as "datastores") may be implemented in any way including standard database technologies such as the structured query language (SQL). ('093, 4:17-20 ("The CMDB datastore **260** and the license datastore **270** may be implemented as a collection of flat files, a structured query language database, or in any other way desired.").) The term "structured query language" (SQL) refers to a well-known and standard technique for accessing and updating information in databases. (*See* Microsoft Computer Dictionary (5th ed. 2002), Ex. 1006, at p. 501 ("**structured query language** *n.* A database sublanguage used in querying, updating, and managing relational databases—the de facto standard for database products.") (bold and italics in original).) The claimed "configuration database" and "license database" are therefore generic.

99. With respect to the claimed "program store," it is similarly generic. The specification explains that the program store can simply be a standard hard disk drive. ('093, 4:25-27 ("A program storage device (PSD) **380** (sometimes referred to as a hard disc) is included with the system unit **310**."), Fig. 3.)

100. Accordingly, the "**server**," **processor**," "**configuration database**," "**license database**," and "**program store**" limitations do not reflect any unique or inventive technology and therefore add no inventive concept that would distinguish the claims from the abstract idea. These limitations simply require that the method steps of claim 1 be performed by generic computer equipment. The claims do not recite any specific programming or technical implementation for carrying out the steps of claim 1. Therefore, in my opinion, the claims **do not add any inventive concept sufficient to meaningfully distinguish them from the abstract idea**.

VII. CONCLUSION

101. In my opinion, for all of the reasons above, claims 1, 5, 10-13, and 16 of the '093 patent are directed to an abstract idea and are unpatentable. I hereby declare under penalty of perjury that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under 28 U.S.C. § 1001.

Dated: August 9th, 2015

Respectfully submitted,

Tal Lavian, Ph.D.

Tal Lavian 8/9/2015

EXHIBIT A

Tal Lavian, Ph.D.

http://telecommnet.com
http://cs.berkeley.edu/~tlavian
tlavian@telecommnet.com

1640 Mariani Dr.
Sunnyvale, CA 94087
(408)-209-9112

Research and Consulting: Telecommunications, Network Communications, and Mobile Wireless technologies

- Scientist, educator, and technologist with over 25 years of experience
- Co-author on over 25 scientific publications, journal articles, and peer-reviewed papers
- Named inventor on over 80 issued and filed patents
- Industry fellow and lecturer at UC Berkeley Engineering – Center for Entrepreneurship and Technology (CET)

EDUCATION

- **Ph.D.**, Computer Science specializing in networking and communications, UC Berkeley
- **M.Sc.**, Electrical Engineering, Tel Aviv University
- **B.Sc.**, Mathematics and Computer Science, Tel Aviv University

EXPERTISE

Network communications, telecommunications, Internet protocols and mobile wireless:

- **Communication networks:** Internet Protocols; TCP/IP suite; TCP; UDP; IP; VoIP; Ethernet; network protocols; network software applications; Data Link, Network, and Transport Layers (L2, L3, L4)
- **Internet Software:** Internet software applications; distributed computing; cloud computing; Web applications; FTP; HTTP; Java; C; C++; client server; file transfer; multicast; streaming media
- **Routing/switching:** LAN; WAN; VPN; routing protocols; RIP; BGP; MPLS;OSPF; IS-IS;DNS; QoS; switching; packet switching; network infrastructure; network communication architectures
- **Mobile Wireless:** Wireless LAN; 802.11; cellular systems; mobile devices; smartphone technologies

LITIGATION SUPPORT SERVICES

- Expert witness in numerous USPTO PTAB – Inter Partes Review (IPR) and CBM cases
- Expert witness in Federal courts and the ITC (over 30 cases)
- Expert reports, depositions, and courtroom testimonies
- Skilled articulation of technical material for both technical and non-technical audiences
- Product and technology analysis, patent portfolios, claim charts, patentability research
- Litigation support and technology education in patent disputes
- Past cases involved Cisco, Juniper, HP, Ericsson, Microsoft, Google, Samsung and Apple

ACCOMPLISHMENTS

- Selected as Principal Investigator for three US Department of Defense (DARPA) projects
- Led research project on networking computation for the US Air Force Research Lab (AFRL)
- Led and developed the first network resource scheduling service for grid computing
- Led wireless research project for an undisclosed US federal agency
- Managed and engineered the first demonstrated transatlantic dynamic allocation of 10Gbs Lambdas as a grid service
- Spearheaded the development of the first demonstrated wire-speed active network on commercial hardware
- Invented over 80 patents; over 50 prosecuted *pro se* in front of the USPTO
- Created and chaired Nortel Networks' EDN Patent Committee
- Current IEEE Senior Member

PROFESSIONAL EXPERIENCE

University of California, Berkeley, Berkeley, CA 2000-Present

Berkeley Industry Fellow, Lecturer, Visiting Scientist, Ph.D. Candidate, Nortel's Scientist Liaison

Some positions and projects were concurrent, others sequential

- Serves as an Industry Fellow and Lecturer at the Center for Entrepreneurship and Technology (CET).
- Studied network services, telecommunication systems and software, communications infrastructure, and data centers
- Developed long-term technology for the enterprise market, integrating communication and computing technologies
- Conducted research projects in data centers (RAD Labs), telecommunication infrastructure (SAHARA), and wireless systems (ICEBERG)
- Acted as scientific liaison between Nortel Research Lab and UC Berkeley, providing tangible value in advanced technologies
- Earned a Ph.D. in Computer Science with a specialization in communications and networking

Telecomm Net Consulting, Inc. (Innovations-IP) Sunnyvale, CA 2006-Present

Principal Scientist

- Consulting in the areas of network communications, telecommunications, Internet protocols, and smartphone mobile wireless devices

- Providing architecture and system consultation for software projects relating to computer networks, mobile wireless devices, Internet web technologies
- Acting as an expert witness in network communications patent infringement lawsuits

VisuMenu, Inc. – Sunnyvale, CA 2010-Present

Co- Founder and Chief Technology Officer (CTO)

- Design and develop architecture of visual IVR technologies for smartphones and wireless mobile devices in the area of network communications
- Design crawler/spider system for IVR / PBX using Asterisk, SIP and VoIP
- Deploy the system as cloud networking and cloud computing utilizing Amazon Web Services (EC2, S3, VPC, DNS, and RDS)

Ixia, Santa Clara, CA 2008-2008

Communications Consultant

- Researched and developed advanced network communications testing technologies:
  • IxNetwork/IxN2X — tests IP routing and switching devices and broadband access equipment. Provides traffic generation and emulation for the full range of protocols: routing, MPLS, layer 2/3 VPNs, Carrier Ethernet, broadband access, and data center bridging.
  • IxLoad — quickly and accurately models high-volume video, data, and voice subscribers and servers to test real-world performance of multiservice delivery and security platforms.
  • IxCatapult — emulates a broad range of wireless access and core protocols to test wireless components and systems. When combined with IxLoad, provides an end-to-end solution for testing wireless service quality.
  • IxVeriWave — employs a client-centric model to test Wi-Fi and wireless LAN networks by generating repeatable large-scale, real-world test scenarios that are virtually impossible to create by any other means.
  • Test Automation — provides simple, comprehensive lab automation to help test engineering teams create, organize, catalog, and schedule execution of tests.

Nortel Networks, Santa Clara, CA 1996 - 2007

Originally employed by Bay Networks, which was acquired by Nortel Networks

Principal Scientist, Principal Architect, Principal Engineer, Senior Software Engineer

- Held scientific and research roles at Nortel Labs, Bay Architecture Labs, and in the office of the CTO

Principal Investigator for US Department of Defense (DARPA) Projects

- Conceived, proposed, and completed three research projects: Active Networks, DWDM-RAM, and a networking computation project for Air Force Research Lab (AFRL)
- Led a wireless research project for an undisclosed US federal agency

Academic and Industrial Researcher

- Analyzed new technologies to reduce risks associated with R&D investment
- Spearheaded research collaboration with leading universities and professors at UC Berkeley, Northwestern University, University of Amsterdam, and University of Technology, Sydney
- Evaluated competitive products relative to Nortel's products and technology
- Proactively identified prospective business ideas, which led to new networking products
- Predicted technological trends through researching the technological horizon and academic sphere
- Developed software for switches, routers and network communications devices
- Developed systems and architectures for switches, routers, and network management
- Researched and developed the following projects:
  - Data-Center Communications: network and server orchestration 2006-2007
  - DRAC: SOA-facilitated L1/L2/L3 network dynamic controller 2003-2007
  - Omega: classified wireless project for undisclosed US Federal Agency 2006
  - Open Platform: project for the US Air Force Research Laboratory (AFRL) 2005
  - Network Resource Orchestration for Web Services Workflows 2004-2005
  - Proxy Study between Web/Grids Services and Network Services 2004
  - Streaming Content Replication: real-time A/V media multicast at edge 2003-2004
  - DWDM-RAM: US DARPA-funded program on agile optical transport 2003-2004
  - Packet Capturing and Forwarding Service on IP and Ethernet traffic 2002-2003
  - CO2: content-aware agile networking 2001-2003
  - Active Networks: US DARPA-funded research program 1999-2002
  - ORE: programmable network service platform 1998-2002
  - JVM Platform: Java on network devices 1998-2001
  - Web-Based Device Management: network device management 1996-1997

Technology Innovator and Patent Leader

- Created and chaired Nortel Networks' EDN Patent Committee
- Facilitated continuous stream of innovative ideas and their conversion into intellectual property rights
- Developed intellectual property assets through invention and analysis of existing technology portfolios

Aptel Communications, Netanya, Israel 1994-1995

Software Engineer, Team Leader

Start-up company focused on mobile wireless CDMA spread spectrum PCN/PCS

- Developed a mobile wireless device using an unlicensed band [Direct Sequence Spread Spectrum (DSSS)]
- Designed and managed a personal communication network (PCN) and personal communication system (PCS), the precursors of short text messages (SMS)
- Designed and developed network communications software products (mainly in C/C++)
- Brought a two-way paging product from concept to development

Scitex Ltd.,Herzeliya, Israel 1990-1993

Software Engineer, Team Leader

Software and hardware company acquired by Hewlett Packard (HP)

- Developed system and network communications (mainly in C/C++)
- Invented Parallel SIMD Architecture
- Participated in the Technology Innovation group

Shalev, Ramat-HaSharon, Israel 1987-1990

Start-up company

Software Engineer

- Developed real-time software and algorithms (mainly in C/C++ and Pascal)

PROFESSIONAL ASSOCIATIONS

- IEEE Senior Member
- IEEE CNSV co-chair Intellectual Property SIG (2013)
- President Next Step Toastmasters (an advanced TM club in the Silicon Valley) (2013)
- Technical Co-Chair, IEEE Hot Interconnects 2005 at Stanford University
- Member, IEEE Communications Society (COMMSOC)
- Member, IEEE Computer Society
- Member, IEEE Systems, Man, and Cybernetics Society
- Member, IEEE-USA Intellectual Property Committee
- Member, ACM, ACM Special Interest Group on Data Communication (SIGCOM)
- Member, ACM Special Interest Group on Hypertext, Hypermedia and Web (SIGWEB)
- Member, IEEE Consultants' Network (CNSV)
- Global Member, Internet Society (ISOC)
- President Java Users Group – Silicon Valley Mountain View, CA,1999-2000
- Toastmasters International

ADVISORY BOARDS

- Quixey (present) – search engine for wireless mobile apps
- Mytopia – mobile social games
- iLeverage – Israeli Innovations

PROFESSIONAL AWARDS

- Top Talent Award – Nortel
- Top Inventors Award – Nortel EDN
- Certified IEEE-WCET - Wireless Communications Engineering Technologies
- Toastmasters International - Competent Communicator (twice)
- Toastmasters International - Advanced Communicator Bronze

Patents and Publications

(Not an exhaustive list)

Patents Issued:

- **US 8,688,796** Rating system for determining whether to accept or reject objection raised by user in social network
- **US 8,572,303** Portable universal communication device
- **US 8,553,859** Device and method for providing enhanced telephony
- **US 8,548,131** Systems and methods for communicating with an interactive voice response system
- **US 8,537,989** Device and method for providing enhanced telephony
- **US 8,341,257** Grid proxy architecture for network resources
- **US8,161,139** Method and apparatus for intelligent management of a network element
- **US 8,146,090** Time-value curves to provide dynamic QoS for time sensitive file transfer
- **US 8,078,708** Grid proxy architecture for network resources
- **US 7,944,827** Content-aware dynamic network resource allocation
- **US7,860,999** Distributed computation in network devices
- **US 7,734,748** Method and apparatus for intelligent management of a network element
- **US 7,710,871** Dynamic assignment of traffic classes to a priority queue in a packet forwarding device
- **US 7,580,349** Content-aware dynamic network resource allocation
- **US 7,433,941** Method and apparatus for accessing network information on a network device
- **US 7,359,993** Method and apparatus for interfacing external resources with a network element
- **US 7,313,608** Method and apparatus for using documents written in a markup language to access and configure network elements
- **US 7,260,621** Object-oriented network management interface

- **US 7,237,012** Method and apparatus for classifying Java remote method invocation transport traffic
- **US 7,127,526** Method and apparatus for dynamically loading and managing software services on a network device
- **US7,047,536** Method and apparatus for classifying remote procedure call transport traffic
- **US7,039,724** Programmable command-line interface API for managing operation of a network device
- **US6,976,054** Method and system for accessing low-level resources in a network device
- **US6,970,943** Routing architecture including a compute plane configured for high-speed processing of packets to provide application layer support
- **US6,950,932** Security association mediator for Java-enabled devices
- **US6,850,989** Method and apparatus for automatically configuring a network switch
- **US6,845,397** Interface method and system for accessing inner layers of a network protocol
- **US6,842,781** Download and processing of a network management application on a network device
- **US6,772,205** Executing applications on a target network device using a proxy network device
- **US6,564,325** Method of and apparatus for providing multi-level security access to system
- **US6,175,868** Method and apparatus for automatically configuring a network switch
- **US6,170,015** Network apparatus with Java co-processor
- **US 8,619,793** Dynamic assignment of traffic classes to a priority queue in a packet forwarding device
- **US 8687,777** Systems and methods for visual presentation and selection of IVR menu
- **US 8,681,951** Systems and methods for visual presentation and selection of IVR menu

- **US 8,625,756** Systems and methods for visual presentation and selection of IVR menu
- **US 8,594,280** Systems and methods for visual presentation and selection of IVR menu
- **US 8,548,135** Systems and methods for visual presentation and selection of IVR menu
- **US 8,406,388** Systems and methods for visual presentation and selection of IVR menu
- **US 8,345,835** Systems and methods for visual presentation and selection of IVR menu
- **US 8,223,931** Systems and methods for visual presentation and selection of IVR menu
- **US 8,160,215** Systems and methods for visual presentation and selection of IVR menu
- **US 8,155,280** Systems and methods for visual presentation and selection of IVR menu
- **US 8,054,952** Systems and methods for visual presentation and selection of IVR menu
- **US 8,000,454** Systems and methods for visual presentation and selection of IVR menu
- **EP 1,905,211** Technique for authenticating network users
- **EP 1,142,213** Dynamic assignment of traffic classes to a priority queue in a packet forwarding device
- **EP 1,671,460** Method and apparatus for scheduling resources on a switched underlay network
- **CA 2,358,525** Dynamic assignment of traffic classes to a priority queue in a packet forwarding device

Patent Applications Published and Pending:
(Not an exhaustive list)

- **US 20140105025** Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device
- **US 20140105012** Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device
- **US 20140012991** Grid Proxy Architecture for Network Resources
- **US 20130080898** Systems and Methods for Electronic Communications
- **US 20130022191** Systems and Methods for Visual Presentation and Selection of IVR Menu
- **US 20130022183** Systems and Methods for Visual Presentation and Selection of IVR Menu
- **US 20130022181** Systems and Methods for Visual Presentation and Selection of IVR Menu
- **US 20120180059** Time-Value Curves to Provide Dynamic QOS for Time Sensitive File Transfers
- **US 20120063574** Systems and Methods for Visual Presentation and Selection of IVR Menu
- **US 20110225330** Portable Universal Communication Device
- **US 20100220616** Optimizing Network Connections
- **US 20100217854** Method and Apparatus for Intelligent Management of a Network Element
- **US 20100146492** Translation of Programming Code
- **US 20100146112** Efficient Communication Techniques
- **US 20100146111** Efficient Communication in a Network
- **US 20090313613** Methods and Apparatus for Automatic Translation of a Computer Program Language Code
- **US 20090313004** Platform-Independent Application Development Framework
- **US 20090279562** Content-aware dynamic network resource allocation
- **US 20080040630** Time-Value Curves to Provide Dynamic QoS for Time Sensitive File Transfers
- **US 20070169171** Technique for authenticating network users
- **US 20060123481** Method and apparatus for network immunization
- **US 20060075042** Extensible Resource Messaging Between User Applications and Network Elements in a Communication Network

- **US 20050083960** Method and Apparatus for Transporting Parcels of Data Using Network Elements with Network Element Storage
- **US 20050076339** Method and Apparatus for Automated Negotiation for Resources on a Switched Underlay Network
- **US 20050076336** Method and Apparatus for Scheduling Resources on a Switched Underlay Network
- **US 20050076173** Method And Apparatus for Preconditioning Data to Be Transferred on a Switched Underlay Network
- **US 20050076099** Method and Apparatus for Live Streaming Media Replication in a Communication Network
- **US 20050074529** Method and apparatus for transporting visualization information on a switched underlay network
- **US 20040076161** Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device
- **US 20020021701** Dynamic Assignment of Traffic Classes to a Priority Queue in a Packet Forwarding Device
- **WO 2007/008976** Technique for Authenticating Network Users
- **WO 2006/063052** Method and apparatus for network immunization
- **WO2000/0054460** Method and apparatus for accessing network information on a network device

Publications

(Not an exhaustive list)

- "R&D Models for Advanced Development & Corporate Research" Understanding Six Models of Advanced R&D - Ikhlaq Sidhu, Tal Lavian, Victoria Howell - University of California, Berkeley. Accepted paper for 2015 ASEE Annual Conference and Exposition- June 2015
- "Communications Architecture in Support of Grid Computing",Tal Lavian,Scholar's Press 2013ISBN 978-3-639-51098-0.
- "Applications Drive Secure Lightpath Creation across Heterogeneous Domains, Feature Topic Optical Control Planes for Grid Networks: Opportunities, Challenges and the Vision." Gommans L.; Van Oudenaarde B.; Dijkstra F.; De Laat C.; Lavian T.; Monga I.; Taal A.; Travostino F.; Wan A.; IEEE *Communications Magazine*, vol. 44, no. 3, March 2006, pp. 100-106.
- *Lambda Data Grid: Communications Architecture in Support of Grid Computing*. Tal I. Lavian, Randy H. Katz; Doctoral Thesis, University of California at Berkeley. January 2006.
- "Information Switching Networks." Hoang D.B.; T. Lavian;*The 4th Workshop on the Internet, Telecommunications and Signal Processing, WITSP2005*, December 19-21, 2005, Sunshine Coast, Australia.
- "Impact of Grid Computing on Network Operators and HW Vendors." Allcock B.; Arnaud B.; Lavian T.; Papadopoulos P.B.; Hasan M.Z.; Kaplow W.; *IEEE Hot Interconnects at Stanford University 2005*, pp.89-90.
- *DWDM-RAM: AData Intensive Grid Service Architecture Enabled by Dynamic Optical Networks*. Lavian T.; Mambretti J.; Cutrell D.; Cohen H.J; Merrill S.; Durairaj R.; Daspit P.; Monga I.; Naiksatam S.; Figueira S.; Gutierrez D.; Hoang D.B., Travostino F.; CCGRID 2004, pp. 762-764.
- *DWDM-RAM: AnArchitecture for Data Intensive Service Enabled by Next Generation Dynamic Optical Networks*. Hoang D.B.; Cohen H.; Cutrell D.; Figueira S.; Lavian T.; Mambretti J.; Monga I.; Naiksatam S.; Travostino F.; Proceedings IEEE Globecom 2004, Workshop on High-Performance Global Grid Networks, Houston, 29 Nov. to 3 Dec. 2004, pp.400-409.
- *Implementation of a Quality of Service Feedback Control Loop on Programmable Routers*. Nguyen C.; Hoang D.B.; Zhao, I.L.; Lavian, T.; Proceedings, 12th IEEE International Conference on Networks 2004. (ICON 2004) Singapore, Volume 2, 16-19 Nov. 2004, pp.578-582.
- *A Platform for Large-Scale Grid Data Service on Dynamic High-Performance Networks*. Lavian T.; Hoang D.B.; Mambretti J.; Figueira S.; Naiksatam S.; Kaushil N.; Monga I.; Durairaj R.; Cutrell D.; Merrill S.; Cohen H.; Daspit P.; Travostino F; GridNets 2004, San Jose, CA., October 2004.
- *DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks*. Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Daspit, P.; Gutierrez D.; Hoang D. B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F; Proceedings, 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004, pp. 707-714.
- *DWDM-RAM: Enabling Grid Services with Dynamic Optical Networks*. Figueira S.; Naiksatam S.; Cohen H.; Cutrell D.; Gutierrez D.; Hoang D.B.; Lavian T.; Mambretti J.; Merrill S.; Travostino F.; 4th IEEE/ACM International Symposium on Cluster Computing and the Grid, Chicago, USA, April 2004.

- *An Extensible, Programmable, Commercial-Grade Platform for Internet Service Architecture*. Lavian T.; Hoang D.B.; Travostino F.; Wang P.Y.; Subramanian S.; Monga I.; IEEE Transactions on Systems, Man, and Cybernetics on Technologies Promoting Computational Intelligence, Openness and Programmability in Networks and Internet Services Volume 34, Issue 1, Feb. 2004, pp.58-68.
- *DWDM-RAM: An Architecture for Data Intensive Service Enabled by Next Generation Dynamic Optical Networks*. Lavian T.; Cutrell D.; Mambretti J.; Weinberger J.; Gutierrez D.; Naiksatam S.; Figueira S.; Hoang D. B.; Supercomputing Conference, SC2003 Igniting Innovation, Phoenix, November 2003.
- *Edge Device Multi-Unicasting for Video Streaming*. Lavian T.; Wang P.; Durairaj R.; Hoang D.; Travostino F.; Telecommunications, 2003. ICT 2003. 10th International Conference on Telecommunications, Tahiti, Volume 2, 23 Feb.-1 March, 2003 pp. 1441-1447.
- The SAHARA Model for Service Composition Across Multiple Providers. Raman B.; Agarwal S.; Chen Y.; Caesar M.; Cui W.; Lai K.; Lavian T.; Machiraju S.; Mao Z. M.; Porter G.; Roscoe T.; Subramanian L.; Suzuki T.; Zhuang S.; Joseph A. D.; Katz Y.H.; Stoica I.; Proceedings of the First International Conference on Pervasive Computing. ACM Pervasive 2002, pp. 1-14.
- *Enabling Active Flow Manipulation in Silicon-Based Network Forwarding Engines*. Lavian T.; Wang P.; Travostino F.; Subramanian S.; Duraraj R.; Hoang D.B.; Sethaput V.; Culler D.; Proceeding of the Active Networks Conference and Exposition, 2002.(DANCE) 29-30 May 2002, pp. 65-76.
- *Practical Active Network Services within Content-Aware Gateways*. Subramanian S.; Wang P.; Durairaj R.; Rasimas J.; Travostino F.; Lavian T.; Hoang D.B.; Proceeding of the DARPA Active Networks Conference and Exposition, 2002.(DANCE) 29-30 May 2002, pp. 344-354.
- *Active Networking on a Programmable Network Platform.* Wang P.Y.; Lavian T.; Duncan R.; Jaeger R.; Fourth IEEE Conference on Open Architectures and Network Programming (OPENARCH), Anchorage, April 2002.
- *Intelligent Network Services through Active Flow Manipulation*. Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; IEEE Intelligent Networks 2001 Workshop (IN2001), Boston, May 2001.
- Intelligent Network Services through Active Flow Manipulation. Lavian T.; Wang P.; Travostino F.; Subramanian S.; Hoang D.B.; Sethaput V.; Intelligent Network Workshop, 2001 IEEE 6-9 May 2001, pp.73 - 82.
- *Enabling Active Flow Manipulation in Silicon-based Network Forwarding Engine*. Lavian, T.; Wang, P.; Travostino, F.; Subramanian S.; Hoang D.B.; Sethaput V.; Culler D.; Journal of Communications and Networks, March 2001, pp.78-87.
- *Active Networking on a Programmable Networking Platform*. Lavian T.; Wang P.Y.; IEEE Open Architectures and Network Programming, 2001, pp. 95-103.
- *Enabling Active Networks Services on a Gigabit Routing Switch*. Wang P.; Jaeger R.; Duncan R.; Lavian T.; Travostino F.; 2nd Workshop on Active Middleware Services, 2000.

- *Dynamic Classification in Silicon-Based Forwarding Engine Environments*. Jaeger R.; Duncan R.; Travostino F.; Lavian T.; Hollingsworth J.; Selected Papers. 10th IEEE Workshop on Metropolitan Area and Local Networks, 1999. 21-24 Nov. 1999, pp.103-109.
- *Open Programmable Architecture for Java-Enabled Network Devices*. Lavian, T.; Jaeger, R. F.; Hollingsworth, J. K.; IEEE Hot Interconnects Stanford University, August 1999, pp. 265-277.
- *Open Java SNMP MIB API*. Rob Duncan, Tal Lavian, Roy Lee, Jason Zhou, Bay Architecture Lab Technical Report TR98-038, December 1998.
- *Java-Based Open Service Interface Architecture*. Lavian T.; Lau S.; BAL TR98-010 Bay Architecture Lab Technical Report, March 1998.
- *Parallel SIMD Architecture for Color Image Processing*. Lavian T. Tel – Aviv University, Tel – Aviv, Israel, November 1995.
- *Grid Network Services, Draft-ggf-ghpn-netservices-1.0*.George Clapp, Tiziana Ferrari, Doan B. Hoang, Gigi Karmous-Edwards, Tal Lavian, Mark J. Leese, Paul Mealor, Inder Monga, Volker Sander, Franco Travostino, Global Grid Forum(GGF).
- *Project DRAC: Creating an applications-aware network.*Travostino F.; Keates R.; Lavian T.; Monga I.; Schofield B.; Nortel Technical Journal, February 2005, pp. 23-26.
- *Optical Network Infrastructure for Grid, Draft-ggf-ghpn-opticalnets-1.*Dimitra Simeonidou, Reza Nejabati, Bill St. Arnaud, Micah Beck, Peter Clarke, Doan B. Hoang, David Hutchison, Gigi Karmous-Edwards, Tal Lavian, Jason Leigh, Joe Mambretti, Volker Sander, John Strand, Franco Travostino, Global Grid Forum(GGF) GHPN Standard GFD-I.036 August 2004.
- *Popeye - Using Fine-grained Network Access Control to Support Mobile Users and Protect Intranet Hosts.*Mike Chen, Barbara Hohlt, Tal Lavian, December 2000.

Presentations and Talks

(Not an exhaustive list)

- Lambda Data Grid: An Agile Optical Platform for Grid Computing and Data-intensive Applications.
- Web Services and OGSA
- WINER Workflow Integrated Network Resource Orchestration.
- Technology & Society.
- Abundant Bandwidth and how it affects us?
- Active Content Networking(ACN).
- DWDM-RAM:Enabling Grid Services with Dynamic Optical Networks .
- Application-engaged Dynamic Orchestration of Optical Network Resources .
- A Platform for Data Intensive Services Enabled by Next Generation Dynamic Optical Networks .
- Optical Networks.
- Grid Optical Network Service Architecture for Data Intensive Applications.
- Optical Networking & DWDM.
- OptiCal Inc.
- OptiCal & LUMOS Networks.
- Optical Networking Services.
- Business Models for Dynamically Provisioned Optical Networks.
- Business Model Concepts for Dynamically Provisioned Optical Networks.
- Optical Networks Infrastructure.
- Research Challenges in agile optical networks.
- Services and Applications' infrastructure for agile optical networks.
- Impact on Society.
- TeraGrid Communication and Computation.
- Unified Device Management via Java-enabled Network Devices.
- Active Network Node in Silicon-Based L3 Gigabit Routing Switch.
- Active Nets Technology Transfer through High-Performance Network Devices.
- Programmable Network Node: Applications.
- Open Innovation via Java-enabled Network Devices.
- Practical Considerations for Deploying a Java Active Networking Platform.
- Open Java-Based Intelligent Agent Architecture for Adaptive Networking Devices.
- Java SNMP Oplet.
- Open Distributed Networking Intelligence: A New Java Paradigm.
- Open Programmability.
- Active Networking On A Programmable Networking Platform.
- Open Networking through Programmability.
- Open Programmable Architecture for Java-enabled Network Devices.

- Integrating Active Networking and Commercial-Grade Routing Platforms.
- Programmable Network Devices.
- To be smart or not to be?