

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

VMWARE, INC.,
Petitioner,

v.

CIRBA IP, INC.,
Patent Owner.

Case: PGR2021-00098
Patent: 10,951,459

DECLARATION OF DR. EREZ ZADOK, PH.D.

TABLE OF CONTENTS

I.	Qualifications.....	2
II.	Materials Considered.....	12
III.	Understanding of Relevant Legal Principles.....	12
	A. Claim Construction Standard	12
	B. Written Description	13
IV.	Background.....	14
	A. Technical Background.....	14
	B. Overview of the '459 Patent.....	18
	1. System Parameters	20
	2. Compatibility Analyses.....	21
	3. Consolidation Analyses	23
	4. Visualization	26
	5. Summary of the '459 Patent Prosecution History	26
V.	Level of Skill in the Art.....	28
VI.	Claim Construction.....	29
VII.	The '459 Patent Specification Does Not Describe Claims 1-63 Such That a POSITA Would Understand the '459 patent Inventors Were in Possession of the Claimed Systems, Methods, or Computer Readable Medium	30
	A. “Already Placed”.....	30
	1. Analysis Types.....	32
	a) 1-to-1 Compatibility Analysis.....	32
	b) N-to-1 Compatibility Analysis.....	33

c)	N-by-N Compatibility Analysis.....	33
d)	Consolidation Analysis	34
2.	“Source System”	35
B.	“Issue Instructions”	37
C.	Incorporated Applications	41

1. My name is Erez Zadok. I have been retained by Petitioner VMware Inc. (“Petitioner” or “VMware”) to assist regarding U.S. Patent No. 10,951,459 (Ex. 1001, “the ’459 patent”). Specifically, I have been asked to consider the patentability of claims 1-63 of the ’459 patent (“the Challenged Claims”) in view of prior art and the understanding of a person of ordinary skill in the art (“POSITA”) as it relates to the ’459 patent. I understand that my opinions in this declaration are being submitted as part of the petition in PGR2021-00098. Additionally, I understand two other of my declarations that address the same claims of the ’459 patent will be submitted in connection with two IPR petitions (IPR2021-01210 and IPR2021-01211).

2. I have personal knowledge of the facts and opinions set forth in this declaration and believe them to be true. If called upon to do so, I would testify competently thereto. I have been warned that willful false statements and the like are punishable by fine or imprisonment, or both.

3. My consulting company, Zadoks Consulting, LLC, is being compensated for my time at my standard consulting rate. I am also being reimbursed for expenses that I incur during the course of this work. My compensation is not contingent upon the results of my study and analysis, the substance of my opinions, or the outcome of any proceeding involving the Challenged Claims. I have no

financial interest in the outcome of this matter or in any litigation involving the '459 patent.

I. Qualifications

4. I am a Professor in the Computer Science Department at Stony Brook University (part of the State University of New York (“SUNY”) system). I direct the File-systems and Storage Lab (FSL) at Stony Brook’s Computer Science Department. My research interests include file systems and storage systems, operating systems, information technology and system administration, security and information assurance, networking, energy efficiency, performance and benchmarking, virtualization, compilers, applied machine learning, and software engineering.

5. I studied at a professional high school in Israel, focusing on electrical engineering (“EE”), and graduated in 1982. I spent one more year at the high school’s college division, receiving a special Certified Technician’s degree in EE. I then went on to serve in the Israeli Defense Forces for three years (1983-1986). I received my Bachelor of Science degree in computer science (“CS”) in 1991, my Master’s degree in CS in 1994, and my Ph.D. in CS in 2001—all from Columbia University in New York.

6. When I began my undergraduate studies at Columbia University, I also started working as a student assistant in the various campus-wide computer labs,

eventually becoming an assistant to the head labs manager, who was managing all public computer labs on campus. During that time, I also became more involved with research within the CS Department at Columbia University, conducting research on operating systems, file and storage systems, distributed and networked systems, security, and other topics. I also assisted the CS department's computer administrators in managing the department's computers, which included storage, IT, networking, and cyber-security related duties.

7. In 1991, I joined Columbia University's CS department as a full-time systems administrator, studying towards my MS degree part-time. My MS thesis topic related to file system reliability, fault tolerance, replication, and failover in mobile networked storage systems using file virtualization. My main duties as a systems administrator involved installing, configuring, and managing many networked servers, proxies, and desktops running several operating systems, as well as network devices setup; this included many hardware upgrades, device upgrades, and BIOS firmware/chipset updates/upgrades. My duties also included ensuring reliable, secure, authenticated access to networked systems/storage and licensed software, as well as software updates, security and bug fixes. Examples of servers and their protocols included email (SMTP), file transfer (FTP), domain names (DNS), network file systems (NFS), network news systems (NNTP), and Web (HTTP).

8. In 1994, I left my systems administrator position to pursue my doctoral studies at Columbia University. My Ph.D. thesis topic was on versatile file system development using stackable (virtualized) file systems, with examples in the fields of security and encryption, efficiency, reliability, and failover. I continued to work part-time as a systems administrator at the CS department, and eventually I was asked to serve as manager to the entire information technology (“IT”) staff. From 1991 to 2001, I was a member of the faculty-level Facilities Committee that oversaw all IT operations at the CS department.

9. As part of my Ph.D. studies at Columbia, I collaborated on projects to develop advanced AI-like techniques to detect previously unknown viruses (a.k.a. “zero-day malware”), using data mining and rule-based detection. This work led to several highly cited papers (over 1,300 citations for one of the papers alone), and two patents. I also became a Teaching Assistant (TA) for a first-ever Computer Security course given at Columbia University’s CS department with Dr. Matt Blaze as instructor.

10. From 1990 to 1998, I consulted for SOS Corporation and HydraWEB Technologies, as a systems administrator and programmer, managing data storage use and backup/restore duties, as well as information assurance and cyber-security (e.g., malware protection, software licensing). From 1994 to 2000, I led projects at HydraWEB Technologies, and then became the Director of Software Development-

overseeing the development of several products and appliances such as stateful firewalls and HTTP load-balancers, utilizing network-virtualization and high-availability techniques. Since 2009, I have consulted for Packet General Networks, a startup specializing in secure, virtualized, network storage and applications' data security in the cloud.

11. In 2001, I joined the faculty of Stony Brook University, a position I have held since that time. In 2002, I joined the Operations Committee, which oversees the IT operations of the CS department at Stony Brook University. From 2006 to 2010, I was the Director of IT Operations of the CS department; my day-to-day duties included setting policies regarding computing, hiring and training new staff, assisting any staff with topics of my specialty, defining requirements for new software/hardware, and purchasing. From 2010 to 2015, I have served as the Co-Chair to the Operations Committee. From 2016 to 2019, I oversaw the IT Operations as the Chair of the Operations Committee. A significant component of these duties included defining and helping implement policies for data management, so as to ensure the security of users and their data, and data reliability and availability, while minimizing the inconvenience and performance impact to users. I personally helped setup and maintain an initial virtual host infrastructure in the department. Since late 2019, I've been a member of the department's Executive Committee that also oversees all IT operations.

12. In 2017, I became the department's Graduate Academic Adviser, advising all Masters students (over 400 annually on average) and many other graduate students on an assortment of academic matters.

13. Since 2001, I personally configured and managed my own research lab's network. This includes setting up and configuring multiple storage systems (e.g., NFS, CIFS/SMB, NAS), virtual and physical environments, applications such as database and mail servers, user access control (e.g., NIS, LDAP), backups and restores, snapshot policies, and more. I've personally installed, configured, changed, replaced parts, and upgraded components in numerous devices from laptops to servers, both physical and virtual.

14. Since 1995, I have taught courses on operating systems, storage and file systems, advanced systems programming in Unix/C, systems administration, data structures, data/software security, and more. My courses often use storage, file systems, distributed systems, and system/network security as key teaching principles and practical examples for assignments and projects. I have taught these concepts and techniques to my students, both to my direct advisees as well as in my courses. For example, in my graduate Operating Systems course, I often cover Linux's kernel mechanisms to protect users, applications, and data files, as well as distributed storage systems (e.g., NFS). And in the System Administration undergraduate

course, I covered many topics such as networking, storage, backups, and configuring complex applications such as mail, web, and database servers.

15. My research often investigates computer systems from many angles: security, efficiency, energy use, scalability, reliability, portability, survivability, usability, ease-of-use, versatility, flexibility, and more. My research gives special attention to balancing five often-conflicting aspects of computer systems: performance, reliability, energy use, security, and ease-of-use. Since joining Stony Brook University in 2001, my group in the File-systems and Storage Lab (FSL) has developed many file systems and operating system extensions; examples include a highly-secure cryptographic file system, a portable copy-on-write (COW) versioning file system, a tracing file system useful to detect intrusions, a replaying file system useful for forensics, a snapshotting and sandboxing file system, a namespace unification file system (that uses stackable, file-based COW), an anti-virus file system, an integrity-checking file system, a load balancing and replication/mirroring file system, network file system extensions for security and performance, distributed secure cloud-based storage systems, transactional key-value stores and file systems, OS level embedded databases, a compiler to convert user-level C code to in-kernel efficient yet safe code, GCC plugins, stackable file system templates, and a Web-based backup system. Many of these projects used

one form of virtualization or another (storage, network, host, etc.). I continue to maintain and release newer versions of some of these file systems and software.

16. I have published over 120 refereed publications (in ACM, IEEE, USENIX, and more). To date, my publications have been cited more than 8,700 times (as per Google Scholar as of June 21, 2021). My papers cover a wide range of related technologies such as file systems, storage systems, transactional systems, security, performance benchmarking and optimization, energy efficiency, system administration, and more. I also published a book titled “Linux NFS and Automounter Administration” (Sybex, 2001), covering systems administration topics related to network storage and data security.

17. Some of my research has led to public software releases that have been used worldwide. I have publicly maintained the Amd Berkeley Automounter in a package called “am-utils” since 1992; this software helps administrators manage the multitude of file system mounts on dozens of different Unix systems, especially helping to automate access to multiple NFS/NAS storage volumes. Since 1997, I have maintained and released several stackable (virtualized) file system software projects for Linux, FreeBSD, and/or Solaris, in a package called FiST. One of my stackable file system encryption projects, called Cryptfs, became the basis for IBM’s public release of eCryptfs, now part of Linux. Packet General Networks, for whom I have provided consulting services since 2009, licensed another encryption file

system called Ncryptfs. Another popular file system released in 2003, called Unionfs, offers virtual namespace unification, transparent shadow copying (a.k.a. copy-on-write or COW), file system snapshotting (e.g., useful for forensics and disaster recovery), and the ability to save disk space by sharing a read-only copy of data among several computers, among other features.

18. My research and teaching make extensive use of data security features. For example, each time I taught the graduate operating system course, the first homework assignment includes the creation of a new system call that performs new or added functionality, often for encrypting a file or verifying its integrity; many of my other assignments cover topics of user/process access control, anti-virus filtering, and more. Since 2001, over 1,000 graduate students were exposed to these simple principles directly through my teaching and research at Stony Brook University.

19. Moreover, in an undergraduate course titled “Advanced Systems Programming in Unix/C,” I cover many topics of system security and vulnerabilities, such as the structure of UNIX processes, and memory segments such as the heap and stack. Often, the first assignment for this course is to develop a tool to encrypt/decrypt files using advanced ciphers, use digital signatures to certify the cipher keys used, and reliably recover files in case of failures. Since 2001, several hundred undergraduate students were exposed to these principles directly through my teaching and research at Stony Brook University.

20. In another undergraduate course, System Administration, I taught network configuration, security, and storage configuration and reliability. In a special topics course on Storage Systems, I covered many topics such as data deduplication, RAID, transactional storage, storage hardware including modern Flash based ones, virtual storage, backup/restore, snapshots and continuous data protection (CDP), NAS and SAN, and NFS.

21. Overall , in addition to the aforementioned experience, my technical experience relevant to this patent at the time of the alleged invention included the following: configuring and running hypervisors on Linux, Windows, and Mac OS X; experimenting with and/or using products from VMware (ESX, GSX, Fusion, Workstation), Sun Microsystems (VirtualBox), Linux (open-source Xen and KVM), and others (e.g., Parallels); installed and ran numerous virtual machines (VMs) on these hypervisors, setup storage backends, configured and executed VM migration, manually load-balanced and consolidated VMs and hosts, evaluated hardware capabilities of hosts and VMs, and optimized my VM clusters. I've also studied, researched, taught and published on topics of operating system and storage system optimizations using complex algorithms.

22. My research has been supported by many federal and state grants as well as industry awards, including an NSF CAREER award, two IBM Faculty awards, two NetApp Faculty awards, a Western Digital award, a Facebook award,

several Dell-EMC awards, and several equipment gifts. I was the winner of the 2004 Computer Science Department bi-annual Graduate Teaching Award, the winner of the 2006 Computer Science Department bi-annual Research Excellence Award, and a recipient of the 2008 SUNY Chancellor's Excellence in Teaching award (an award that can be given only once a lifetime).

23. My service record to the community includes being the co-chair for the USENIX Annual Technical Conference in 2020 (ATC'20); being the co-chair for USENIX File and Storage Technologies (FAST) in 2015 and being on the FAST Steering Committee since 2015; joining the ACM HotStorage Steering Committee in 2021; and being the co-chair in 2012 and on the Steering Committee of the ACM SYSTOR conference. I have also been an Associate Editor to the ACM Transactions on Storage (TOS) journal since 2009.

24. I am a named inventor on four patents, two titled "Systems and Methods for Detection of New Malicious Executables" (U.S. Patent No. 7,487,544, issued February 3, 2009; and U.S. Patent No. 7,979,907, issued July 12, 2011); and two more titled "Multi-Tier Caching," (U.S. Patent No. 9,355,109, issued May 31, 2016; and U.S. Patent 9,959,279, issued May 1, 2018).

25. I have been disclosed as a testifying expert in 13 cases (including *inter partes* review (IPR) proceedings) in the past four years. I have been deposed 11 times and testified in trial twice. A complete copy of my curriculum vitae, which

includes a list of my publications and contains further details on my education, experience, publications, patents, and other qualifications to render an expert opinion, is attached as Exhibit 1007.

II. Materials Considered

26. In performing my analysis and forming the opinions below, I considered the '459 patent, its prosecution history, the applications that it incorporates by reference, and the materials listed as exhibits to the petition in (this) PGR2021-00098.

III. Understanding of Relevant Legal Principles

27. I am not a lawyer, and I will not provide any legal opinions. Rather, I have been asked to provide my technical opinions based on how a person of ordinary skill in the art would have understood the claims of the '459 patent, in light of its disclosure and prosecution history, as of the patent's priority date. Although I am not a lawyer, I have been advised that certain legal standards are to be applied by technical experts in forming opinions regarding the meaning and validity of patent claims.

A. Claim Construction Standard

28. I understand that claim terms are given their ordinary and customary meaning, as would be understood by a person of ordinary skill in the relevant art in the context of the patent's entire disclosure and prosecution history. A claim term,

however, will not receive its ordinary meaning if the patentee acted as his own lexicographer and clearly set forth a definition of the claim term in the specification. In that case, the claim term will receive the definition set forth in the patent.

29. The face of the '459 patent claims priority to a series of applications, the earliest of which was filed April 21, 2006. As I explain below, none of the applications in this priority chain provide written description support to the challenged claims. I understand that this means the '459 patent's priority date is its November 19, 2019 filing date. Even if, however, an earlier priority date—up to and including the earliest April 21, 2006 date listed on the face of the '459 patent—were considered for purposes of determining the level of ordinary skill in the art and interpreting the '459 patent, my opinions below would not change.

B. Written Description

30. I have been informed that 35 U.S.C. § 112 requires patents to contain an adequate written description of the claimed invention. I understand that the purpose of the written description requirement is to demonstrate that the inventor was in possession of the invention at the time the patent application was filed, even though subsequently the claims may have been changed or new claims may have been added. I understand that this requirement is met if, at the time of filing the patent application, a person of ordinary skill in the art reading that application would have recognized that it described the invention as claimed. I understand that the

application does not need to specifically disclose a claim limitation as long as a person of ordinary skill in the art would understand that the missing requirement is necessarily implied in the application as originally filed. I understand that the written description inquiry must take into account the entirety of what is disclosed within the specification. I understand that it is insufficient that undisclosed subject matter would have been obvious to a person of ordinary skill in the art.

IV. Background

A. Technical Background

31. To carry out their day-to-day operations, businesses and other organizations typically require an information technology (“IT”) infrastructure to provide computing and data processing services. IT infrastructures frequently include one or more “data centers,” where computing and networking equipment is concentrated for the purpose of collecting, storing, processing, or permitting access to data, software, or other computing resources. (Ex. 1013 (Wood) at p. 229 (“Data centers—server farms that run networked applications—have become popular in a variety of domains such as web hosting, enterprise systems, and e-commerce sites.”).)

32. In the early days of computing, such a data center might have contained a single, powerful “mainframe” computer with enough resources to store large amounts of data and provide services to the organization’s users. But as computing

equipment got smaller and cheaper, organizations began to replace these large monolithic mainframe computers with compute “clusters” made up of many (hundreds or even thousands of) smaller, cheaper computer servers, networked together. (Ex. 1014 (Baker) at p. 1 (“there has been a dramatic shift from mainframe or ‘host-centric’ computing to a distributed ‘client-server’ approach” due to “the realisation that clusters of high-performance workstations can be realistically used for a variety of applications either to replace mainframes, vector supercomputers and parallel computers or to better manage already installed collections of workstations.”).)

33. Like the monolithic mainframe had, the cluster would provide users with the perception of single computing resource. (Ex. 1020 (PlateSpin) at p. xxiv (“*Clustering* allows several physical machines to collectively host one or more virtual servers.”).) Users could connect to that resource over a network to access data and computing services as before. (Ex. 1020 (PlateSpin) at p. xxiv (“With clustering, clients won’t connect to a physical computer but instead connect to a logical virtual server running on top of one or more physical computers.”).)

34. As data centers grew and changed over time, one problem that arose is “server sprawl.” (Ex. 1018 (Khanna) at Abstract, p. 373 (“As businesses have grown, so has the need to deploy I/T applications rapidly to support the expanding business processes. Often, this growth was achieved in an unplanned way,” which

“has led to what is often referred to as ‘server and storage sprawl’, i.e., many underutilized servers, with heterogeneous storage elements.”).)

35. Over time, as data centers grew, they frequently came to include “an increasingly complex mixture of server platforms, hardware, operating systems, and applications.” (Ex. 1020 (PlateSpin) at p. 505.) But as new servers and software were added, it quickly became difficult to track them all and to ensure that they remained sufficiently utilized. (Ex. 1020 (PlateSpin) at p. 513 (“Today’s data center is riddled with production servers that are underutilized and that therefore represents a large amount of potential savings”).) For example, legacy technologies persisted long after they were no longer needed, and underutilized servers imposed significant unnecessary cost in terms of power, temperature conditioning, floor space, licensing, and other costs of maintenance and ownership. (Ex. 1020 (PlateSpin) at p. 505 (“Data centers have accumulated and assimilated a large variety of new technologies that over time have become ‘legacy’ technologies that never go away” and they “collectively increase data center costs because of power consumption, temperature conditioning, and floor space.”).)

36. In light of the above issues, before the ’459 patent’s earliest priority date, the advantages of server consolidation were well-known. For example, a patent assigned to IBM and filed in June 2005 recognized the importance of consolidation analysis and the cost benefits of consolidation:

It was also known that in some cases where the customer is currently using two or more of its own servers, ***these can be consolidated into one dedicated server*** in an “on demand” or “utility” model. For example, if the customer was only using a small percentage of the capacity of its current servers, these may be consolidated into one vendor server of similar or equal power in an “on demand” or “utility” model. ***The cost savings result*** from a reduction in (a) number of operating systems, (b) virtual memory, (c) real memory, (d) swap disk space, (e) system management software licenses, (f) customer application software licenses, (g) systems administration and support, (h) floor space, and/or (i) electricity and cooling costs.

(Ex. 1015 (Taylor) at 2:1-13 (emphases added); *see also* 11:43-15:5 referring to Figs. 4(A)-(E) describing a program that “determines whether two customer servers can be consolidated into one vendor server.”) Similarly, there were known techniques for “costing and planning the consolidation of multiple source server computers or other source computer hardware devices to fewer target server computers or other target hardware devices.” (Ex. 1016 (Power) at 3:22-26.) And others were describing how to identify “different possible configurations [of communications systems] that consolidate some or all of two or more of servers in the system.” (Ex. 1017 (Van Hoose) at Abstract.)

37. It also was well-known that one way to combat server sprawl is by using virtualization to perform “server consolidation.” (Ex. 1019 (Menascé) at p. 1 (“Virtualization may be used for server consolidation”); Ex. 1020 (PlateSpin) at p. 505 (“server virtualization alleviates the complexity of the data center to some extent

through consolidation”).) Server consolidation refers to the process of converting some physical machines into VMs and consolidating those VMs onto remaining physical machines, which act as hosts. (See Ex. 1018 (Khanna) at pp. 373-74 (illustrating five steps of “a simple algorithm for server consolidation” using virtualization).)

B. Overview of the ’459 Patent

38. I have reviewed the overview of the ’459 patent from the corresponding petition and incorporate it below.

39. The ’459 patent presents techniques for calculating “consolidation solutions” that can allegedly improve the efficiency of a computing environment with multiple computer systems. (’459 patent at 2:5-13, 2:66-3:10.) Specifically, these consolidation solutions are recommendations on where to move existing applications and data from certain systems (called “source systems”) to other systems (called “target systems”) in order to reduce the number of systems in the overall environment. (’459 patent at 5:16-24, 5:55-64, 6:35-51, 8:17-25.)

40. The ’459 patent explains that a “system” can be a “physical system,” a “virtual system,” or a “hypothetical system” (i.e., a system that does not currently exist in the environment but is useful for analyzing hypothetical situations). (’459 patent at 5:65-6:10.) According to the ’459 patent, managing an environment with many of these systems presents challenges in “optimizing efficiency” and

“avoid[ing] redundancies and/or under-utilized hardware.” (’459 patent at 1:47-52.) For example, a sub-optimal environment may include “additional hardware” with “separate maintenance considerations” that require costly “incidental attention[.]” (’459 patent at 1:63-67.) Too many systems means that “[h]eat production and power consumption can also be a concern.” (’459 patent at 1:67-2:1.)

41. The ’459 patent discloses analytical techniques that purportedly address these problems by calculating “roadmaps” for efficiently reducing the number of systems. (’459 patent at 5:16-24.) The roadmaps identify “source” systems, “from which applications and/or data are to be moved,” and “target” systems, “to which applications and/or data are to be moved.” (’459 patent at 5:57-60, 6:46-51.) Thus, for example, “an underutilized environment having two systems [] can be consolidated to a target system (one of the systems) by moving applications and/or data from the source system (the other of the systems) to the target system.” (’459 patent at 5:60-64.)

42. The ’459 patent discloses an “analysis program” that performs techniques for generating consolidation roadmaps. (’459 patent at 5:16-24.) The program starts by gathering technical, business, and workload parameters for each system in the collection of systems to be consolidated. (’459 patent at 5:16-24.) Next, the program uses the parameters to compute how compatible the systems are with each other. (’459 patent at 5:16-24.) Finally, the program produces

consolidation roadmaps using the compatibility analyses. ('459 patent at Abstract, 5:16-24.) These steps are illustrated in Figure 1 and explained in further detail below:

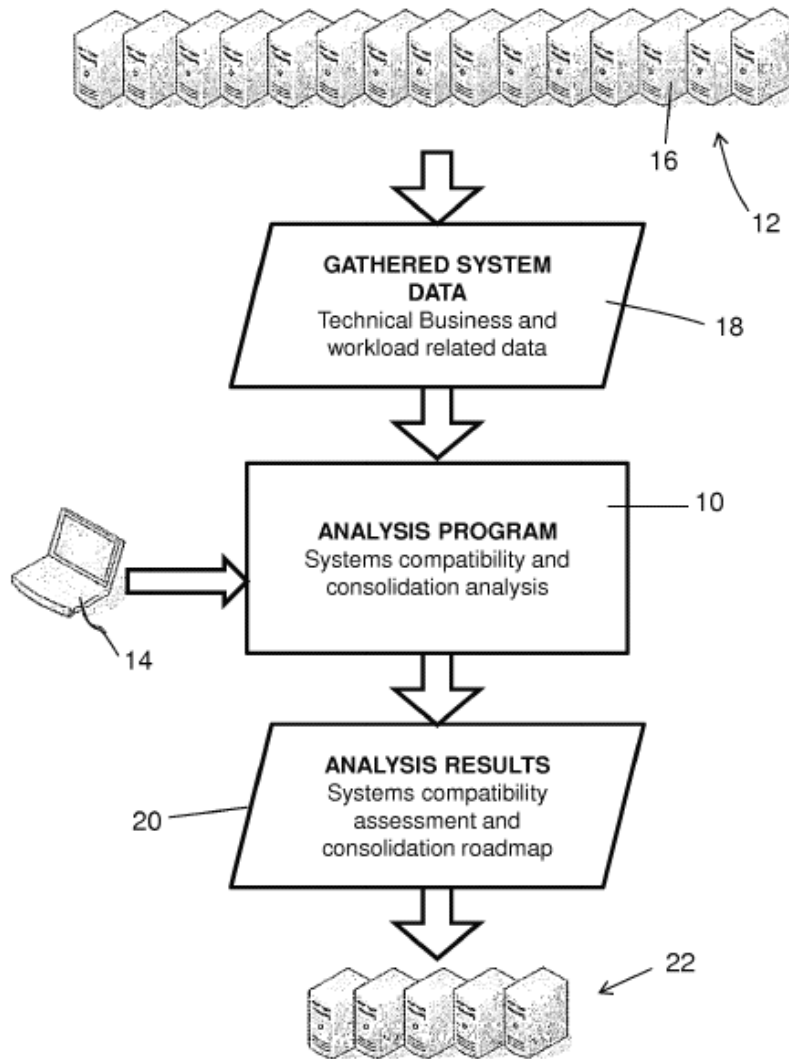


Figure 1

1. System Parameters

43. In the first step, the analysis program “gathers data 18 pertaining to a collection of systems to be consolidated,” including a “distinct data set” for each

system. ('459 patent at 5:18-20, 5:29-30.) The analysis program may collect “technical” parameters such as a system’s “operating system, OS version, patches, application settings, hardware devices, etc.” ('459 patent at 5:40-42.) The program may also collect “business” parameters, such as “physical location, organization department, data segregation requirements, owner, service level agreements, maintenance windows, hardware lease agreements, software licensing agreements, etc.” ('459 patent at 5:43-48.) Finally, the program may collect “workload” parameters, including “resource utilization and capacity metrics related to the system processor, memory, disk storage, disk I/O throughput and network bandwidth utilization.” ('459 patent at 5:49-53.)

2. Compatibility Analyses

44. Using this data, the analysis program calculates systems’ compatibilities with each other. To do this, the program 10 applies “rule sets” to the systems’ technical and business parameters, as well as “workload stacking algorithms” to their workload parameters. ('459 patent at 6:66-7:7.) A rule set might, for example, consider whether systems run different operating systems, or are in different time zones, with different weights and penalties for each parameter when computing a compatibility score based on the rule set. ('459 patent at 14:7-16; *see also* Fig. 16.) Similarly, a workload stacking algorithm might consider factors such as CPU usage and the effect that moving one system’s applications and data to

another system would have on the latter's CPU utilization. ('459 patent at 23:13-23, 23:45-55, 24:62-25:4.) Thus, for a set of two or more systems, the analysis program calculates a separate compatibility score for each rule set and each specified workload type, and then combines the scores into an "overall compatibility score" for the set of systems. ('459 patent at 10:10-14, 29:46-54.)

45. The '459 patent specification describes three types of compatibility analyses that are referenced in the claims. The first, 1-to-1, analyzes the compatibility of each possible source-target pair in the set of systems to be consolidated. ('459 patent at 7:19-21.) In other words, every system is evaluated as a source against every other system as a target, and also as a target against every other system as a source, such that, for a set of N systems, the analysis generates an N-by-N matrix of compatibility scores. (*See, e.g.*, '459 patent at 7:19-21, 26:53-56 ("[T]he 1-to-1 compatibility analyses of N system[s] computes N×N compatibility scores by individually considering each system 16 as a consolidation source and as a target."), Fig. 37.)

46. The patent refers to the other two types of compatibility analysis as "multi-dimensional compatibility analysis." ('459 patent at 7:26-31.) These analysis types focus on "transfer sets," meaning sets of "one or more transfers that involve a common target, wherein the set specifies one or more source entities, the target and a transfer type." ('459 patent at 6:42-45.)

47. While an N-to-1 compatibility analysis evaluates the compatibility of each source system in a transfer set with the target system, an N-by-N compatibility analysis evaluates the compatibilities of source systems with another. ('459 patent at 30:36-44.) For example, in an exemplary embodiment discussed in the '459 patent's specification, a transfer set includes source systems s1, s2, and s3 and target system s16, and the exemplary N-to-1 compatibility analysis evaluates s1 against s16, s2 against s16, and s3 against s16. ('459 patent at 31:40-46.)

48. In the '459 patent's exemplary N-by-N analysis, on the other hand, the system evaluates s1 against s2, s2 against s1, s2 against s3, s3 against s2, s1 against s3, and s3 against s1. ('459 patent at 32:40-43.)

49. N-to-1 and N-by-N compatibility analyses may be combined into a single multi-dimensional analysis, depending on the applicable rule sets and workload types. ('459 patent at 30:46-49.) As with 1-to-1 compatibility analyses, multi-dimensional compatibility analyses compute a compatibility score for each rule set and workload type as well as an overall compatibility score, but do so for transfer sets instead of source-target pairs. ('459 patent at 29:59-62.)

3. Consolidation Analyses

50. In the last step of the '459 patent's disclosed techniques, the analysis program uses the scores generated through compatibility analyses to identify a

consolidation roadmap 20 that “maximizes the number of transfers” while meeting any pre-defined constraints. (’459 patent at 33:60-64.)

51. The consolidation analysis first filters out combinations of incompatible systems from the universe of possible transfers. (’459 patent at 35:32-48.) It then “compiles a collection of candidate transfer sets” from the remaining pool of potential sources and targets through the use of a heuristic “auto fit algorithm” that generates candidate transfer sets based on “auto fit parameters” that govern how transfer sets can be created (e.g., minimum allowable score, minimum transfers permitted, maximum transfers permitted, etc.). (’459 patent at 35:14-48.) Next, the analysis compares the candidate transfer sets and “chooses the ‘best’ transfer set,” according to pre-defined criteria, such as the number of transfers in the set or the set’s overall compatibility score. (’459 patent at 35:57-63.) This “best” transfer set then becomes part of an “intermediate consolidation solution,” and the systems involved “are removed from the list of available sources and targets.” (’459 patent at 35:65-36:2.)

52. This process repeats until “all the sources or targets are consumed or ruled out,” thereby creating a complete “consolidation solution.” (’459 patent at 36:4-6.) The user may then choose to create additional consolidation solutions, utilizing different “auto fit parameters” for generating different candidate transfer sets to add to a consolidation solution. (’459 patent at 36:6-9.) The process can then

select a “best consolidation solution” from among the different consolidation solutions. (’459 patent at 36:10-16.)

53. Ultimately, the ’459 patent’s techniques produce a set of analytical results that include the consolidation solution and its associated compatibility scores, as shown in Figure 29, annotated below. This allows the computing environment “to operate with few systems” (’459 patent at 8:24-25) and “can save a significant amount of cost on a yearly basis” (’459 patent at 2:1-4).

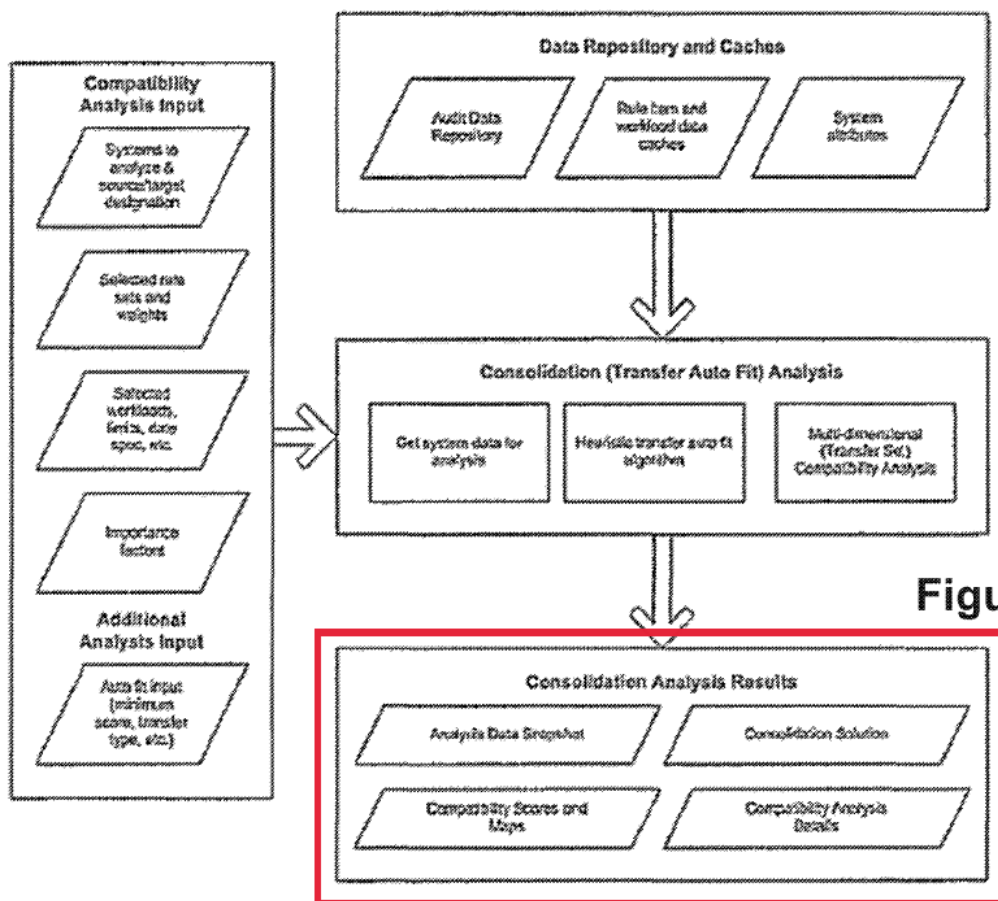


Figure 29

4. Visualization

54. Optionally, according to the '459 patent, its analytical techniques can include visualizing or reporting the results of the analyses at a user interface. ('459 patent at 26:52-28:62.) The '459 patent does not describe any further steps beyond performing the analysis and presenting the analysis results for the user's consideration.

5. Summary of the '459 Patent Prosecution History

55. As originally filed, the application for the '459 patent included only one claim directed to a “computer-implemented method for determining a consolidation solution[.]” (U.S. Patent Application No. 16/687,966 as originally filed (the “'966 Application”)) (Ex. 1002) at 63.) This claim reflected the analysis program described above and in the rest of the specification. The full claim is reproduced below:

A computer-implemented method for determining a consolidation solution for a plurality of computer systems comprising:

- a) obtaining a data set comprising a compatibility score for each pair of said plurality of computer systems, each said compatibility score having been computed using parameters of said plurality of computer systems to indicate a comparable compatibility of respective parameters of one of said plurality of computer systems with respect to another of said plurality of computer systems;
- b) using said compatibility scores to determine one or more candidate transfer sets from said plurality of computer systems and, each candidate transfer set indicating one or more source computer systems having compatible parameters that enable the source computer system(s) to be transferred to a target computer system;

- c) selecting a desired one of said one or more candidate transfer sets according to a predetermined selection criterion and adding said desired one of said one or more candidate transfer sets to a first candidate consolidation solution;
- d) repeating b) and c) by, at each iteration, removing computer systems included in said desired one of said one or more candidate transfer sets and using said compatibility scores corresponding to remaining computer systems to generate at least one additional desired candidate transfer set to be added to said first candidate consolidation solution; and
- e) providing said first candidate consolidation solution as a desired consolidation solution when no additional candidate transfer sets can be determined in b).

(’966 Application at 63.)

56. This lone claim, submitted in the original application to the Patent Office on November 19, 2019, focused on calculating a consolidation solution specifying sets of transfers of applications and data from source computer systems to target computer systems. It notably did *not* refer to a source system that was “placed,” let alone “already placed,” on a target system. Nor did it refer to “issuing instructions” to place a source system onto a target system according to a determined placement.

57. In an amendment on July 2, 2020, the Patent Owner canceled the single originally filed claim and added 57 new claims. (’966 application July 2, 2020 Preliminary Amendment (Ex. 1003) at 2-58.) These new claims included, for the first time, the limitation in the as-issued ’459 patent claims relating to “evaluat[ing] compatibility” between a source system and another source system that was “already

placed” on a target system by evaluating the “impact on resource utilization of the target system[.]” (*See, e.g.*, Ex. 1003 at 2.) The new claims also included, for the first time, limitations for “issu[ing] instructions to place the at least one source system on the at least one target system in accordance with” a determined placement. (*See, e.g.*, Ex. 1003 at 2.) The examiner allowed the claims on October 5, 2020. (’966 application October 4, 2020 Notice of Allowability (Ex. 1004) at 2.) But on December 10, 2020, the Patent Owner made further amendments that added an additional “already placed” limitation for evaluating compatibility between a source system and another source system. (’966 application December 10, 2020 Post-Allowance Amendment (Ex. 1005) at 2.)

58. The Patent Owner did not identify specific parts of the specification as written description support for the newly-claimed subject matter.

V. Level of Skill in the Art

59. In my opinion, a person of ordinary skill in the art (“POSITA”) for the technology of the ’459 patent would have had at least a bachelor’s degree in computer science, computer engineering, electrical engineering, or a related degree and 2-3 years of experience with administering remote computing environments. This experience could include, for example, work in cloud computing, virtualization, network communications, client-server communication analysis, load balancing, stress testing, and/or system consolidation. A POSITA would have been familiar

with designing, implementing, monitoring, evaluating, optimizing, and managing physical and virtual computing environments. More education would be acceptable in place of work experience and vice versa.

VI. Claim Construction

60. The terms “source system” and “target system” appear in each claim of the ’459 patent. According to the specification, a “source system” is “a system from which applications and/or data are to be moved.” (’459 patent at 5:57-58.) A “target system,” is stated as “a system to which [a source system’s] applications and/or data are to be moved.” (’459 patent at 5:59-60.) I apply these meanings in my analysis below.

61. For all other terms, if any, I have been instructed to perform my analysis with the understanding that the terms of the Challenged Claims should be afforded their plain and ordinary meaning. However, should Patent Owner and/or Patent Owner’s expert choose to construe additional claim terms or challenge the interpretations proposed herein, I reserve the right to provide my opinion regarding Patent Owner’s construction(s) as well as the meaning a POSITA would have ascribed to those claim terms.

VII. The '459 Patent Specification Does Not Describe Claims 1-63 Such That a POSITA Would Understand the '459 patent Inventors Were in Possession of the Claimed Systems, Methods, or Computer Readable Medium

62. I understand that to provide written description support for its claims, a patent's specification must describe the claims in enough detail that a person of ordinary skill in the art would understand that the inventors invented the claims as of the patent's priority date. Based on the insufficient disclosure of the "already placed" and "issuing instructions" limitations, a person of ordinary skill in the art would not understand that that '459 patent's inventors invented the '459 patent's claimed systems, methods, or computer readable medium.

63. Below I cite to the '459 patent specification, which is identical to the '936 application's specification as filed. My analysis equally applies to the '936 application's specification.

A. "Already Placed"

64. Independent claim 1 recites the following limitations (and the other independent claims recite similar limitations), emphases added:

evaluate compatibility between the specific source system from the plurality of source systems and one or more other source systems either *already placed* on the specific target system, or being evaluated for placement onto the specific target system . . . ;

evaluate compatibility . . . by evaluating the impact on resource utilization of the specific target system of placing the specific source system on the specific target system, in

combination with the one or more other source systems, either *already placed* on the specific target system, or being evaluated for placement onto the specific target system;

65. There are three types of compatibility analyses in the '459 patent: 1-to-1, N-to-1, and N-by-N. ('459 patent at 7:14-16 (“A collection of systems 16 to be consolidated can be analyzed in one of three modes: 1-to-1 compatibility, multi-dimensional compatibility and consolidation analyses.”).)

66. Multi-dimensional compatibility analysis further includes N-to-1 and N-by-N compatibility analyses. ('459 patent at 30:46-49 (“To assess the compatibility of transferring multiple source entities (N) to a target (1), the rule-based analysis can compute a compatibility score based on a combination of N-to-1 and N-by-N compatibility analyses.”).) None of these analyses, however, discloses or describes the concept recited above in the claims of evaluating a specific source system against “one or more other source systems [] already placed on the specific target system.” Additionally, the very concept of “one or more other source systems [] already placed on the specific target system” is inconsistent with and unsupported by the express definition of “source system” in the as-filed application that issued as the '459 patent.

1. Analysis Types

a) 1-to-1 Compatibility Analysis

67. This analysis type involves evaluating each system in the computing environment against every other system in the computing environment as both a source system and a target system on a 1-to-1 basis. The '459 patent states:

A high level flow diagram of the 1-to-1 compatibility analysis is shown in FIG. 11. The 1-to-1 compatibility analysis can take into account analysis input, including input regarding the systems 16 to be analyzed, rule set related parameters, workload related parameters, workload benchmarks and importance factors 88 used to compute overall scores.

The compatibility analysis evaluates the compatibility of every specified system as source-target pairs on a 1-to-1 basis. This analysis produces a compatibility score for each system pair so that analyzing a collection of ten (10) systems 16 produces 10×10 scores. The compatibility analysis is based on the specified rule sets and workload types.

('459 patent at 9:41-54.) In other words, the 1-to-1 analysis involves evaluating one target system against one source system. It does not describe evaluating a source system while accounting for one or more other source systems, regardless of whether the one or more other source systems are "already placed" (or already transferred) on a specific target system.

b) N-to-1 Compatibility Analysis

68. Similar to the 1-to-1 compatibility analysis, the N-to-1 compatibility analysis involves evaluations only between source systems and target systems. ('459 patent at 30:49-51 (“An N-to-1 intercompatibility analysis assesses each source system against the target.”).) The example in Table 14 illustrates this concept by showing evaluations of each of source systems s1, s2, and s3 against target system s16. (See '459 patent at 31:42-47.) This does not evaluate a specific source system while accounting for one or more other source systems, regardless of whether the one or more other source systems may have been pre-evaluated, or transferred, and hence “already placed” on a specific target system.

c) N-by-N Compatibility Analysis

69. Unlike 1-to-1 and N-to-1 compatibility analyses that evaluate source systems and target systems against each other, the N-by-N compatibility analysis evaluates N source systems of a transfer set against each other. The exemplary embodiment discusses doing so one pair at a time. In the specification's example of a transfer set of three source systems (s1, s2, and s3), the '459 patent discloses an exemplary N-by-N compatibility analysis that includes “evaluating s1 against s2, s2 against s1, s2 against s3, s3 against s2, s1 against s3 and s3 against s1.” ('459 patent at 32:40-46.) In other words, each system in this example is being considered as possibly a source or a target. But these source systems that are part of a transfer set

cannot be “already placed” on a target system because a transfer set is defined in terms of “multiple sources (N) *to be transferred* to the target” system. (’459 patent at 32:40-46.) In other words, a transfer set (and thus also an N-by-N compatibility analysis) is defined in terms of source systems for *future* transfer to a target system, not *past* transfers that have already occurred.

d) Consolidation Analysis

70. The patent’s “consolidation analysis” uses the above compatibility analyses “to assess potential transfer set candidates.” (’459 patent at 33:64-67.) As I explained above, a “transfer set” is defined in the context of source systems that could be transferred in the future and thus excludes the concept of “already placed” source systems. Although the ’459 patent describes consolidation analysis as an internally iterative process, there is no evaluation of a specific source system against one or more other source systems that were already placed on a specific target. Rather, in the consolidation analysis, after a transfer set is selected as part of the consolidation solution, the transfer set’s N source systems and target system (to which the source systems are “to be transferred”) are removed from the analysis. Neither the source systems that are “to be transferred” to the transfer set’s target system nor the target system itself are considered in subsequent iterations of the transfer set evaluations.

Once a transfer set is chosen, it is added to the intermediate consolidation solution. The entities referenced by the transfer set are *removed from the list of available sources and targets* and the three preceding phases are repeated until the available sources or targets are consumed.

(’459 patent at 35:65-36:2 (emphasis added).)

71. Beyond the internal iterative process of evaluating transfer sets, there is no disclosure of a broader iterative process for the consolidation analysis—one that actually transfers source systems to target systems and then resumes an analysis. For example, Figures 1 and 2 show a “block diagram of an analysis program for evaluating the compatibility of computer systems to identify consolidation solutions” (’459 patent at 3:42-46) and Figure 29 shows “a process flow diagram of the consolidation analysis” (’459 patent at 4:36-37). But none of these figures show that the overall consolidation analysis is repeated. In every case, the overall analysis is a single-pass process.

72. Nowhere in either the ’459 patent’s description of consolidation analysis, or elsewhere in the patent, is a source system ever evaluated against or with one or more other source systems that have been “already placed” on a specific target system in a specific previous step.

2. “Source System”

73. The ’459 patent provides express definitions of “source system” and “target system.”

. . . a source system refers to a system from which applications and/or data are to be moved, and a target server or system is a system to which such applications and/or data are to be moved. For example, an underutilized environment having two systems 16 can be consolidated to a target system (one of the systems) by moving applications and/or data from the source system (the other of the systems) to the target system.

(’459 patent at 5:57-64 (emphasis added).)

74. Like the concept of a “transfer set” that I addressed above, a “source system” is defined in the context of potential future transfers that may occur after moving applications and/or data from a source system to a target system. In the ’459 patent’s specification, a transfer of a source system to a target system occurs when the source system’s applications and/or data are considered to be moved to the target system. The fact that the source system definition specifies that applications and/or data are to be transferred *from* the source system means that a source system is more than just the applications and data such that if the applications and data are removed, there are still other source system components left behind. For example, if a source system is a physical computer, then after moving its application and/or data to a target system, the physical components of the physical computer (e.g., the CPU, memory, disks) would remain along with the computer’s operating system software.

75. The ’459 patent’s claims, however, contradict the source system definition by suggesting that a source system in its entirety (e.g., including the physical/virtual hardware, operating system, standard utilities, and libraries) is

transferred or placed on a target system instead, and not that the source system's application and/or data alone are moved.

76. The definitions "source system" and "target system" make sense in the broader context of the '459 patent, which focuses on the consolidation of systems. The '459 patent repeatedly explains that its goal is to allow a computing environment "to operate with fewer systems." ('459 patent at 8:24-25.) It accomplishes this by removing source systems from the computing environment after the source systems' application and/or data are transferred to a target system. ('459 patent at Figs. 1-3.) Nowhere does the specification disclose shuffling source systems around on target systems in a way that preserves the source systems for future evaluation.

77. Finally, moving an entire physical source system to place it on or inside another target physical system is simply impossible and nonsensical; similarly, one cannot "place" a virtual machine (operating system and all) inside another virtual machine. Not only do the '459 patent's claims suggest something impossible and nonsensical, but this suggestion also would run counter to the '459 patent's goals of *reducing* the total number of running (physical or virtual) systems.

B. "Issue Instructions"

78. The last limitation of independent claim 1 (as well as similar limitations of the other independent claims) recites:

issue instructions to place the at least one source system on the at least one target system in accordance with the determined placement

79. Yet nowhere does the '459 patent explain how to issue instructions to carry out the results of its analyses. To the contrary, the '459 patent's disclosure stops at providing analysis results without any description of how to carry out those results.

80. For example, Figure 7, reproduced below, "is a schematic block diagram of an underlying architecture for implementing the analysis program." ('459 patent at 3:55-57.) Consistent with the lack of disclosure concerning how to "issue instructions," the block diagram does not include any components that a person of ordinary skill in the art would understand to "issue instructions." Rather, the block diagram is consistent with a program that stops at producing analysis results. The blocks for "Web Interface 74," "Report Templates 72," and "Report Engine 70" all inform a person of ordinary skill in the art that the '459 patent's analysis program produces analysis results in the form of reports, graphics, or roadmaps for possible consolidation.

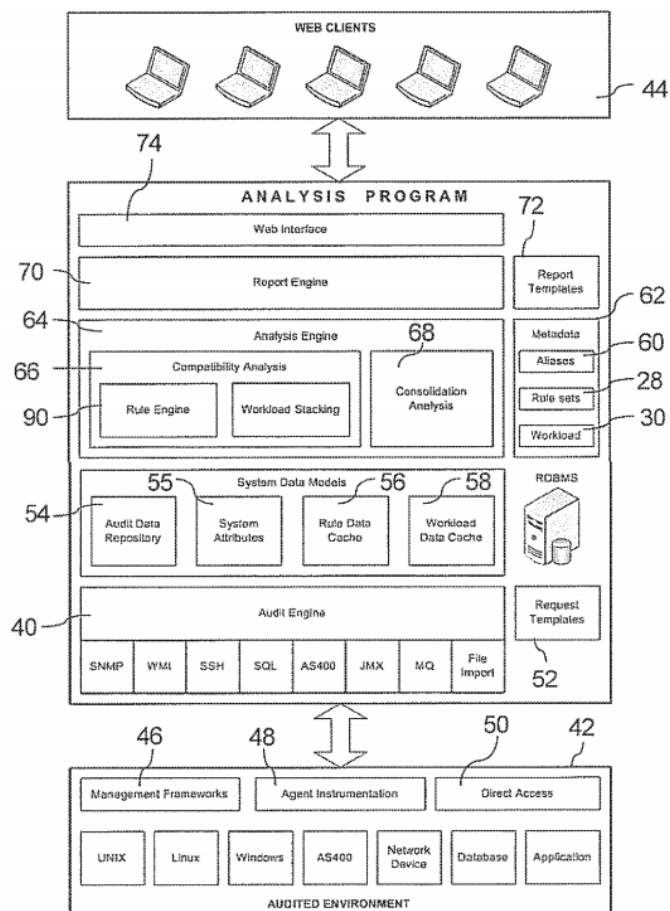


Figure 7

81. There is nothing in Figure 7 (or the rest of the patent) that indicates the program actually carries out the results, such as by issuing instructions. (*See, e.g.*, '459 patent, Fig 9 (showing output as "Analysis Results" or "Consolidation Solution"), Figs. 11, 17, 21, 24(a)-(d), 29 (showing output as "Analysis Results"; Figs. 14, 15 (showing output as "Compatibility Results"; Fig. 18 (showing output as "Compatibility score"); Fig. 30 (showing output as "Consolidation Solution").)

82. The examples of the compatibility and consolidation analyses all stop at choosing or displaying results—they do not disclose executing on the results. For example, in the “Consolidation Analysis” example (starting at 33:59), the last step is “Choose Best Consolidation Solution.” (’459 patent at 36:10.) There is no disclosure of implementing the “best consolidation solution.” Similarly, the “Auto Fit Example” (starting at 36:20) also ends with “Choose Best Consolidation Solution.” (’459 patent at 37:63.) The “1-to-1” (’459 patent at 38:10) and “Multi-Dimensional” (’459 patent at 38:46) compatibility analysis examples end with, respectively, “compatibility maps and corresponding analysis map details” (’459 patent at 38:44-45) and an “overall compatibility details report” (’459 patent at 38:59-60).

83. If the inventors of the (lengthy) ’459 patent had developed techniques for issuing instructions to carry out the consolidation solutions that the rest of the patent describes, a person of ordinary skill in the art would have expected additional detail about how those instructions were created and executed. This sort of transfer of applications and/or data from one system to another requires careful attention to minimize service downtime, to avoid loss of data, to avoid data or application corruption, to prevent network or other bottlenecks while possibly large amounts of data are transferred, and to address other technical and practical considerations. None of that is disclosed or even alluded to in the ’459 patent.

C. Incorporated Applications

84. The '459 patent incorporates several applications by reference.

This application is a continuation of U.S. patent application Ser. No. 14/341,471 filed on Jul. 25, 2014, which is a continuation of U.S. patent application Ser. No. 11/738,936 filed on Apr. 23, 2007, which is a continuation-in-part of U.S. patent application Ser. No. 11/738,936 and also a continuation-in-part of U.S. patent application Ser. No. 11/535,308 both filed on Sep. 26, 2006, both of which claim priority from U.S. Provisional Patent Application No. 60/745,322 filed Apr. 21, 2006, all incorporated herein by reference.

('459 patent at 1:8-17.) The specifications for U.S. Patent Application Nos. 14/341,471 and 11/738,936 are identical to the '459 patent's specification and thus do not provide any additional description of the above limitations beyond what is in the '459 patent's specification.

85. The specifications for U.S. Patent Application Nos. 11/738,936 and 11/535,308 and U.S. Provisional Application No. 60/745,322 have different disclosures than the '459 patent's specification. But these specifications do nothing to support the above limitations in the '459 patent's claims. These applications do not disclose evaluation of a source system with any other source system, much less an "already placed" source system. In particular, these other specifications do not include the concept of N-by-N compatibility analysis that evaluates the source

systems of a transfer set against another. In fact, these other specifications do not include the concept of a transfer set at all.

86. Finally, they do not disclose issuing instructions to carry out any analysis results. They merely include the same analysis results and reports disclosed in the '459 patent.

In signing this Declaration, I recognize that the Declaration will be filed as evidence in a contested case before the Patent Trial and Appeal Board of the United States Patent and Trademark Office. I also recognize that I may be subject to cross-examination in the case. If required, I will appear for cross-examination at the appropriate time.

I declare 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 that these statements were made with knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under section 1001 of Title 18 of the United States Code.

Dated: July 6, 2021

A handwritten signature in black ink, appearing to read "Erez Zadok", written in a cursive style.

Dr. Erez Zadok