IN THE UNITED STATES DISTRICT COURT
FOR THE WESTERN DISTRICT OF TEXAS
WACO DIVISION

DAEDALUS BLUE, LLC, *Plaintiff*, v. ORACLE CORPORATION AND ORACLE AMERICA, INC., *Defendants*.	Case No. 6:20-cv-428 **JURY TRIAL DEMANDED**

DAEDALUS BLUE, LLC'S COMPLAINT FOR PATENT INFRINGEMENT

TO THE HONORABLE JUDGE OF SAID COURT:

Plaintiff, Daedalus Blue, LLC for its Complaint against Defendants Oracle Corporation and Oracle, America, Inc. (collectively, "Oracle"), hereby alleges as follows:

INTRODUCTION

1. The novel inventions disclosed in the Asserted Patents in this matter were invented by International Business Machines Corporation ("IBM"). IBM pioneered the field of shared resources and cloud computing. Every year, IBM spends billions of dollars on research and development to invent, market, and sell new technology, and IBM obtains patents on many of the novel inventions that come out of that work, including the Asserted Patents. The five patents asserted in this case are the result of the work from 15 different IBM researchers, spanning a period of nearly a decade.

2. Over the years, IBM has licensed its inventions—including those claimed in the Asserted Patents—to many companies, including Amazon Web Services.

THE PARTIES

3. Daedalus Blue, LLC ("Daedalus") is the current owner and assignee of the Asserted Patents.

4. Plaintiff Daedalus is a Delaware limited liability company with its principal place of business located at 51 Pondfield Road, Suite 3, Bronxville, NY 10708.

5. Defendant Oracle Corporation is a Delaware Corporation with a principal place of business at 500 Oracle Parkway Redwood City, CA 94065. Oracle Corporation also maintains regional offices in this District, located at 2300 Oracle Way, Austin, Texas, at 5300 Riata Park Court, Building B, Austin, Texas, and at 613 NW Loop 410 San Antonio, Texas.

6. Defendant Oracle America, Inc. ("Oracle America") is a Delaware Corporation with a principal place of business at 500 Oracle Parkway Redwood City, CA 94065. Oracle America also maintains regional offices in this district, located 613 NW Loop 410 San Antonio, Texas.

7. Oracle Corporation and Oracle America conduct business in Texas and in the Western District of Texas, as set forth below.

JURISIDICTION AND VENUE

8. This is an action arising under the patent laws of the United States, 35 U.S.C. § 101, *et seq*. Accordingly, this Court has subject matter jurisdiction pursuant to 28 U.S.C. §§ 1331 and 1338(a).

9. Defendants Oracle are subject to this Court's personal jurisdiction in accordance with due process and/or the Texas Long Arm Statute because, in part, Oracle "[r]ecruits Texas

residents, directly or through an intermediary located in this state, for employment inside or outside this state." *See* Tex. Civ. Prac. & Rem. Code § 17.042.

10. This Court also has personal jurisdiction over Defendants Oracle because they committed and continue to commit acts of direct and/or indirect infringement in this judicial district in violation of at least 35 U.S.C. §§ 271(a) and (b). In particular, on information and belief, Defendants have made, used, offered to sell and sold licenses for, or access to, the accused products in this judicial district, and have induced others to use the accused products in this judicial district.

11. Defendants Oracle are subject to the Court's personal jurisdiction, in part, because they regularly conduct and solicit business, or otherwise engage in other persistent courses of conduct in this district, and/or derive substantial revenue from the sale and distribution of infringing goods and services provided to individuals and businesses in this district.

12. This Court has personal jurisdiction over Defendants Oracle because, *inter alia*, Defendants (1) have substantial, continuous, and systematic contacts with this State and this judicial district; (2) own, manage, and operate facilities in this State and this judicial district; (3) enjoy substantial income from their operations and sales in this State and this judicial district; (4) employ Texas residents in this State and this judicial district; and (5) solicit business and market products, systems and/or services in this State and judicial district including, without limitation, those related to the infringing accused products.

13. Venue is proper in this District pursuant to at least 28 U.S.C. §1319(b)-(c) and §1400(b), at least because Defendants Oracle, either directly or through their agents, have committed acts within this judicial district giving rise to this action, and continue to conduct

business in this district, and/or have committed acts of patent infringement within this District giving rise to this action.

FACTUAL ALLEGATIONS

Daedalus Patents

14. The Asserted Patents in this case relate to groundbreaking improvements to computer network functionality and computer security. The techniques described in the Asserted Patents relate to computer networks and have particular application in the cloud-based computing environments as will be further described below.

15. On July 19, 2005, the U.S. Patent and Trademark Office duly and lawfully issued United States Patent No. 6,920,494 ("the '494 Patent"), entitled "Storage Area Network Methods and Apparatus with Virtual SAN Recognition." A true and correct copy of the '494 Patent is attached hereto as **Exhibit 1**.

16. Daedalus is the owner and assignee of all right, title, and interest in and to the '494 Patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of it.

17. The '494 Patent describes, among other things, novel systems and methods that improve the monitoring and discovery of network components and their topology, thereby allowing users to more efficiently monitor the network and components. These inventive technological improvements solved then-existing problems in the field of storage area networks (SANs) and methods of operating SANs. For example, as described in the '494 Patent, with the rise of the personal computer and workstations in the 1980's, demand by business users led to the development of interconnection mechanisms that permitted otherwise independent computers to access data on another computer's storage devices. A prevalent business network that emerged

was/is the local area network, typically comprising "client" computers (e.g., individual PCs or workstations) connected by a network to a "server" computer. In a storage area network, many storage devices are often placed on a network or switching fabric that can be accessed by several servers (such as file servers and web servers) which, in turn, service respective groups of clients. Sometimes even individual PCs or workstations are enabled for direct access of the storage devices. (*See* Ex. 1. at 1:24-54). The complexity engendered by having storage-area networks of shared-access storage components being used by multiple servers spread across separate local-area networks created system management problems that were addressed by the invention of the '494 Patent. (*See, e.g., id.* at 1:55-2:26).

18. Prior to the invention of the '494 Patent, a drawback in storage area networks arose in managing the proliferation of hosts and storage devices. For example, a storage area network (SAN) has one or more host digital data processors which are coupled to one or more storage devices by an interconnect, for example, a fibre channel-based fabric. Hosts are typically web or file servers for client computers but may be any digital data device that accesses and/or stores information on the storage devices. In managing the SAN connections, solutions existing before the '494 invention focused on setting switches or switch-like interfaces on the network or interconnect fabric between the hosts and storage device, electrically "blocking" certain hosts and certain storage devices. A problem with these solutions is that they permitted only zoning or switch-like control. Another problem is that, by their very nature, these solutions tended to be provider specific. (*See* Ex. 1, at 1:55-63).

19. The '494 Patent overcomes these drawbacks and improves the functioning of a computer network by improving storage area networks (SANs) and methods for operating SANs. The invention of the '494 Patent provides for provisioning and discovery of "virtual"

connections and regions within a SAN, that are not dependent on the limited zoning capabilities and connectivity of the storage fabric switch hardware. (*See, e.g.*, Ex. 1, at 6:58-7:22, 44:25-45:25, Figs. 23, 24). An exemplary depiction of a virtual SAN that can be detected by host adapters and disambiguated by a SAN manager is shown in Fig. 23 of the patent:

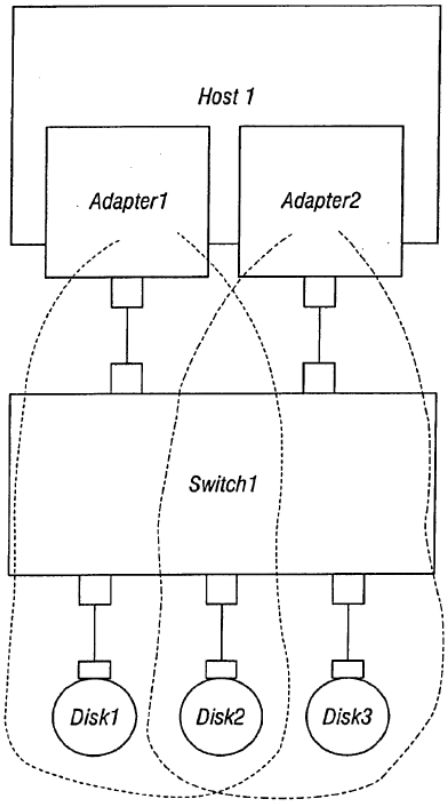


FIG. 23

This "virtual SAN" discovery and management provided a novel and unconventional solution over existing network management solutions at the time. For example, in one aspect of the invention, scanners are utilized for each virtualized region (*e.g.*, defined independent of physical connectivity by the storage resources seen and accessible by a host or group of hosts) of a SAN to collect information regarding the components and their interconnectivity; such scanners are coupled to a manager that uses that information to determine the topology of the SAN. A

scanner may run on an agent within a host. The exemplary arrangement is shown in Fig. 1 of the patent:

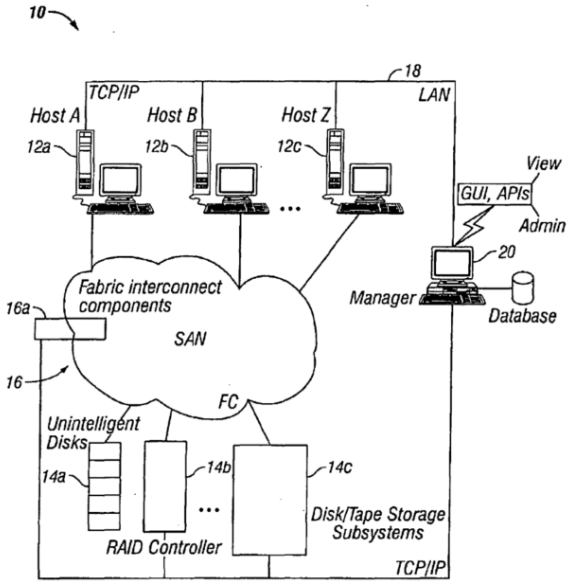


FIG. 1

20. The patent describes, for example, that a scanner may be an executable that interacts with the hosts by performing system calls and I/O control calls to gather information. (Ex. 1, at 39:4-7). The manager may then disambiguate information from the regions and discern the topology of the portion of the SAN spanned by the regions. As illustrated in Fig. 25 of the patent, the SAN manager may store the internal model store of the SAN topology, and such store may contain objects representing the components of the SAN (e.g., hosts, storage devices, interconnect), their attributes and the interrelationships therebetween. The objects may be arranged hierarchically or otherwise. (*Id.*, at 46:61-47:29). They may describe a "collection of ports that may constitute a virtual SAN." (*Id.*, at 45:25-46:60). For example:

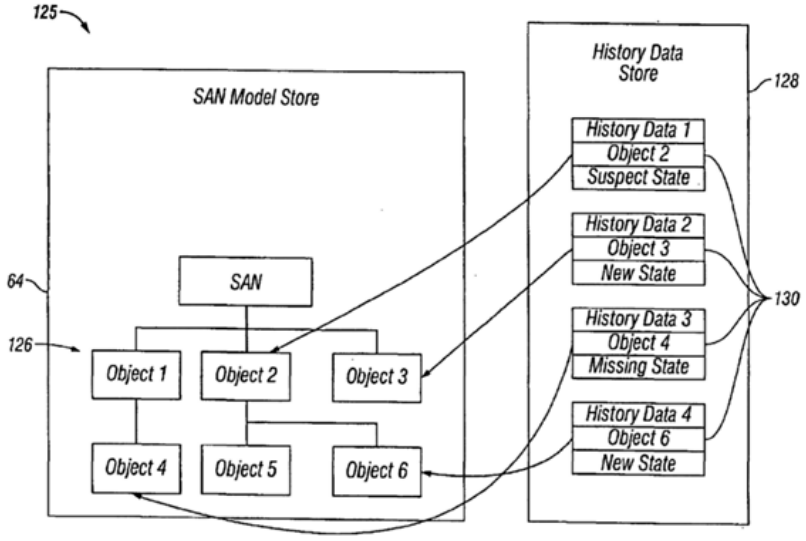


FIG. 25

21. Further, Figure 26 of the patent shows an exemplary hierarchical display that may be presented using the models depicted in Figure 25, and which may also identify information about the status of components or history data:

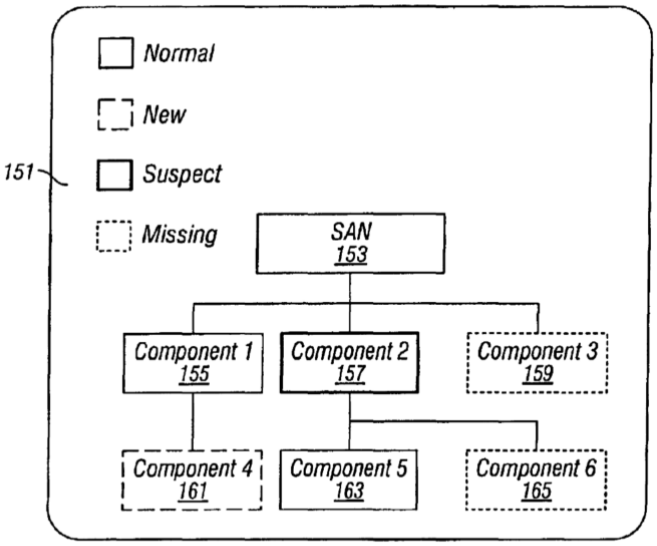


FIG. 26

22. The novel features of the invention are recited in the claims. For example, Claim 1 of the '494 Patent recites:

> A storage area network (SAN) comprising
>
> one or more regions forming at least a portion of the SAN, each region having one or more components, the components including one or more digital data processors and one or more storage devices;
>
> one or more scanners that collect, for each region, information regarding the components and their interconnectivity;
>
> a manager, coupled to the one or more scanners, that responds to the collected information to determine a topology of a portion of the SAN spanned by the regions.

(Ex. 1, at 84:16-27). Claim 1 of the '494 Patent describes claim elements, individually or as an ordered combination, that were non routine and unconventional at the time of the invention in 2001 and was an improvement over prior art. For example, it provided a way (not previously available) to collect information about network components and their interconnectivity in order to monitor and discover network components and their topology. For example, Claim 1 discloses the unconventional step that one or more scanners are maintained to collect information on components for different regions of the SAN, and that the manager is coupled to the scanners for the different regions and responds to the collected information to determine a portion of the SAN topology that spans the regions. As described above and in the disclosures of the '494 Patent, the claimed regions may be inventively and unconventionally virtualized from the storage network fabric hardware, and constitute "virtual SANs." This virtualization functionally improved the discovery and management of storage network components in the complex environment of new cloud-based and networked computing systems.

23. On February 13, 2007, the U.S. Patent and Trademark Office duly and lawfully issued United States Patent No. 7,177,886 ("the '886 Patent"), entitled "Apparatus and Method

for Coordinating Logical Data Replication with Highly Available Data Replication." A true and correct copy of the '886 Patent is attached hereto as **Exhibit 2**.

24. Daedalus is the owner and assignee of all right, title, and interest in and to the '886 Patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of it.

25. The '886 Patent describes, among other things, a novel apparatus configuration that improves data storage techniques that provides for faster, more reliable backup of data files to remote servers, which ensures against data loss and system failure. These inventive technological improvements solved then-existing problems in the field of data replication for databases. For example, as described in the '886 Patent, relational database systems distribute data across a plurality of computers, servers, or other platforms. Distributed database systems typically include a central database and various remote servers that are synchronized with the central database. (Ex. 2, at 1:34-36). The central database server provides a repository for all database contents, and its contents are preferably highly robust against server failures. (*Id.*, at 1:47-49). Remote databases which store some or all information contained in the central database are typically maintained by synchronous or asynchronous data replication. In synchronous replication, a transaction updates data on each target remote database before completing the transaction.

26. However, as described in the '886 Patent, traditional synchronous replication methods introduce substantial delays into data processing, because the replication occurs as part of the user transaction. This increases the cost of the transaction, making it too expensive. Moreover, a problem at a single database can result in an overall system failure. Hence,

synchronous replication is usually not preferred, except in transactions which require a very high degree of robustness against database failure. (*Id*., at 2:9-24).

27. As also described in the ’886 Patent, known methods of asynchronous replication were preferred for most data distribution applications. In asynchronous replication, transaction logs of the various database servers are monitored for new transactions. When a new transaction is identified, a replicator rebuilds the transaction from the log record and distributes it to other database instances, each of which apply and commit the transaction at that instance. Such replicators have a high degree of functionality, and readily support multiple targets, bi-directional transmission of replicated data, replication to dissimilar machine types, and the like. However, asynchronous replicators have a substantial latency between database updates, sometimes up to a few hours for full update propagation across the distributed database system, which can lead to database inconsistencies in the event of a failure of the central database server. Hence, asynchronous replicators are generally not considered to be fail-safe solutions for high data availability. (Ex. 2, at 25-41).

28. The ’886 Patent overcomes these drawbacks and improves the functioning of a computer network, including computer database replication by providing fail-safe data replication in a distributed database system. This invention provides for reliable fail-safe recovery and retains the high degree of functionality of asynchronous replication. (Ex. 2, at 2:42-46). The ’886 Patent describes that, in accordance with one aspect of the invention, a database apparatus includes a critical database server having a primary server supporting a primary database instance and a secondary server supporting a secondary database instance that mirrors the primary database instance. Fig. 1 of the patent shows an exemplary arrangement

where, "[t]he central database server 12 includes a primary server 20 and a secondary server 22 that mirrors the primary server 20." (*Id.* at 4:48-50).

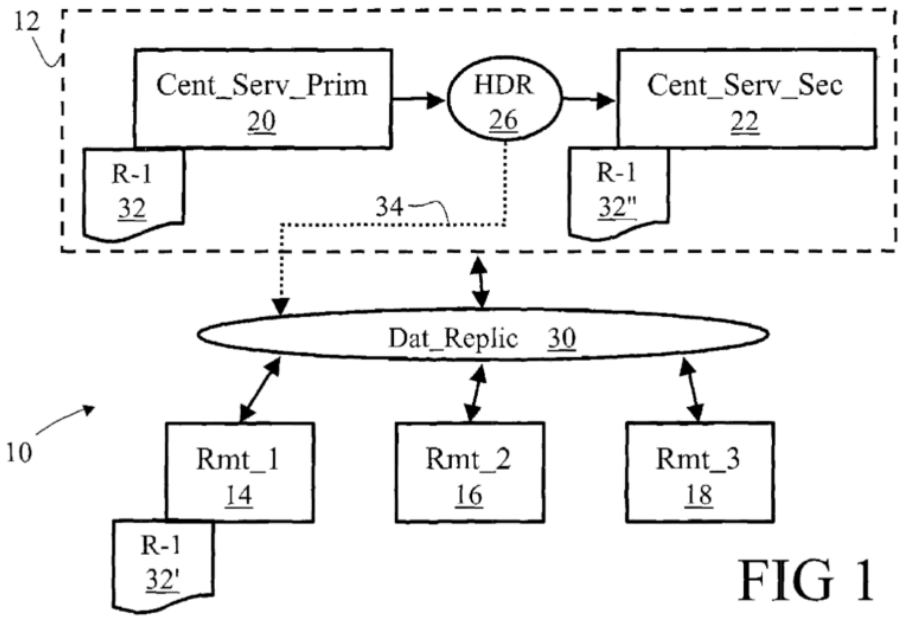


FIG 1

The secondary server generates an acknowledgment signal (34) indicating that a selected critical database transaction is mirrored at the secondary database instance. A plurality of other servers (14, 16, 18) each support a database. A data replicator communicates with the critical database server and one or more of the other servers to replicate the selected critical database transaction on at least one of said plurality of other servers responsive to the acknowledgment signal. (*Id.* at 2:56-67). This configuration of primary and secondary database resources, along with remotely provisioned database backups, was a novel and unconventional system setup that facilitated the improved reliability and failure protection enabled by the claims.

29. The novel features of the invention are recited in the claims. For example, Claim 1 of the '886 Patent recites:

> A database apparatus comprising:
>
> a critical database server including a primary server supporting a primary database instance and a secondary server supporting a secondary database instance that mirrors the primary database instance, the secondary server generating an acknowledgment signal indicating that a selected critical database transaction at the primary database instance is mirrored at the secondary database instance, the critical databases server including a mirroring component communicating with the primary and secondary servers to transfer database log file entries of the primary database instance to the secondary server, the secondary server applying and logging the transferred database log file entries to the secondary database instance and producing said acknowledgement signal subsequent to the applying and logging of the selected critical database transaction, wherein the mirroring component includes a control structure that indexes critical database transactions that are applied and logged at the secondary database instance, the acknowledgement signal corresponding to indexing in the control structure of at least one of the selected critical database transaction and a critical database transaction that commits after the selected critical database transaction;
>
> a plurality of other servers each supporting corresponding database instances; and
>
> a data replicator communicating with the critical database server and the plurality of other servers to replicate the selected critical database transaction on at least one of said plurality of other servers responsive to the acknowledgment signal.

(Ex. 2, at 10:57-11:22). Claim 1 of the ’886 Patent describes claim elements, individually or as an ordered combination, that were non routine and unconventional at the time of the invention in 2003 and an improvement over prior art, as it provided a way (not previously available) to avoid data inconsistencies among remote servers in the event of a failure of the central database primary server; provide asynchronous replication functionality that is robust with respect to primary database failure; and provide for fail-safe recovery via a high availability replication system, while retaining the broad functionality of data distribution by asynchronous replication. (*Id.*, at 3:55-67). For example, in a distributed database system, it was unconventional for a secondary server to produce an acknowledgement for applying received logs to the secondary

database and for a data replicator to wait to replicate critical database transactions in response to such acknowledgement.

30. On October 29, 2013, the U.S. Patent and Trademark Office duly and lawfully issued United States Patent No. 8,572,612 ("the '612 Patent"), entitled "Autonomic Scaling of Virtual Machines in a Cloud Computing Environment." A true and correct copy of the '612 Patent is attached hereto as **Exhibit 3.**

31. Daedalus is the owner and assignee of all right, title, and interest in and to the '612 Patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of it.

32. The '612 Patent describes, among other things, novel systems and methods that improve the data processing and the scaling of resources in a cloud computing environment by efficiently utilizing virtual machines (VM) that autonomically deploy and terminate based on workload. These inventive technological improvements solved then-existing problems in the field of cloud computing. As described in the '612 Patent, cloud computing is a cost-effective means of delivering information technology services through a virtual platform rather than hosting and operating the resources locally. Virtual machines (VMs) may reside on a single powerful blade server, or a cloud system may utilize thousands of blade servers. (*See* Ex. 3, at 1:27-36). A VM is composed of modules of automated computing machinery. (*Id.*, at 1:56-58). The hypervisor (a separate module of automated computing machinery that interacts with the host hardware) creates a particular instance of a VM. (*Id.*, at 6:7-9; 6:25:33). One of the drawbacks of cloud computing systems before the '612 Patent invention, was that the end user would lose control over the underlying hardware infrastructure, including control over scaling the number of virtual machines running an application. In such an environment, scaling of an

application would be carried out manually by a system administrator, but only when end users would report performance degradation. This technique is slow and complex, and it inherently risks a user's experiencing a poor quality of service. (*Id.*, at 1:37-50).

33. The ’612 Patent overcomes these drawbacks and improves the functioning of a computer network, for example, by disclosing an improved way of scaling virtual machine instances using autonomic scaling to deploy additional VM instances, terminate VM instances, and provide user control with little or no governance by hand.

34. In one aspect, the ’612 Patent invention describes autonomic scaling of virtual machines in a cloud computing environment. A self-service portal enables users themselves to set up VMs as they wish, according to the user’s specifications. The cloud operating system then deploys an instance of the now-specified VM in accordance with the received user specifications. The self-service portal passes the user specification to the deployment engine. The VM catalog contains VM templates, standard-form descriptions used by hypervisors to define and install VMs. The deployment engine fills in the selected template with the user specifications and passes the complete template to the data center administration server in the local data center. The data center administration server then calls a hypervisor on a cloud computer to install the instance of the VM specified by the selected, completed VM template. (*See* Ex. 3, at 5:17-36).

35. The ’612 Patent further describes that the cloud computing environment includes a plurality of virtual machines (‘VMs’), and a cloud operating system and a data center administration server operably coupled to the VMs. The cloud operating system deploys an instance of a VM and flags the instance of a VM for autonomic scaling including termination. The cloud operating system monitors one or more operating characteristics of the instance of the VM, deploys an additional instance of the VM if a value of an operating characteristic exceeds a

first predetermined threshold value, and terminates operation of the additional instance of the VM if a value of an operating characteristic declines below a second predetermined threshold value. (*See* Ex. 3, at 1:53-2:6). With autonomic scaling, the environment gracefully handles varying workloads, either increasing or decreasing, and can adapt to varying workloads transparently, smoothly, and with a minimum of difficulty for the users of the data processing service provided by such a cloud computing environment. (*See id.*, at 2:28-45).

36. Figure 3 of the '612 Patent shows a flowchart illustrating example methods of autonomic scaling:

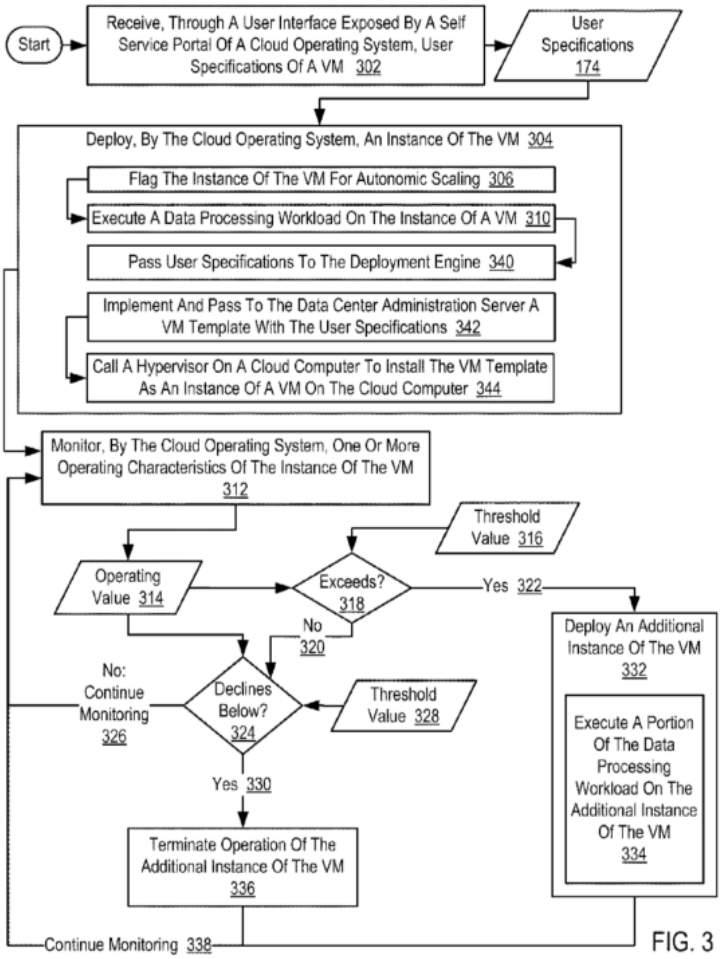


FIG. 3

37. The novel features of the invention are recited in the claims. For example, Claim 1 of the ’612 Patent recites:

> A method of autonomic scaling of virtual machines in a cloud computing environment, the cloud computing environment comprising a plurality of virtual machines (‘VMs’), the VMs comprising modules of automated computing machinery installed upon cloud computers disposed within a data center, the cloud computing environment further comprising a cloud operating system and a data center administration server operably coupled to the VMs, the method comprising:
>
> deploying, by the cloud operating system, an instance of a VM, including flagging the instance of a VM for autonomic scaling including termination and executing a data processing workload on the instance of a VM;
>
> monitoring, by the cloud operating system, one or more operating characteristics of the instance of the VM;
>
> deploying, by the cloud operating system, an additional instance of the VM if a value of an operating characteristic exceeds a first predetermined threshold value, including executing a portion of the data processing workload on the additional instance of the VM; and
>
> terminating operation of the additional instance of the VM if a value of an operating characteristic declines below a second predetermined threshold value;
>
> wherein the cloud operating system comprises a module of automated computing machinery, further comprising a self service portal and a deployment engine, and deploying an instance of a VM further comprises:
>
> passing by the self service portal user specifications for the instance of a VM to the deployment engine;
>
> implementing and passing to the data center administration server, by the deployment engine, a VM template with the user specifications; and
>
> calling, by the data center administration server, a hypervisor on a cloud computer to install the VM template as an instance of a VM on the cloud computer.

(Ex. 3, at 15:42-16:8). Claim 1 of the ’612 Patent describes claim elements, individually or as an ordered combination, that were non routine and unconventional at the time of the invention in 2010 and an improvement over prior art, as it provided a way (not previously available) to add or terminate virtual machines based on individualized thresholds, thereby efficiently utilizing resources and transparently adapting workload. For example, as noted by the U.S. Patent and

Trademark Office upon issuance, the known prior art failed to teach at least the combination of "deploying, by the cloud operating system, an additional instance of the VM if a value of an operating characteristic exceeds a first predetermined threshold value, including executing a portion of the data processing workload on the additional instance of the VM; and terminating operation of the additional instance of the VM if a value of an operating characteristic declines below a second predetermined threshold value, wherein the cloud operating system comprises a module of automated computing machinery, further comprising a self-service portal and a deployment engine, and deploying an instance of a VM further comprises: passing by the self-service portal user specifications for the instance of a VM to the deployment engine; implementing and passing to the data center administration server, by the deployment engine, a VM template with the user specifications; and calling, by the data center administration server, a hypervisor on a cloud computer to install the VM template as an instance of a VM on the cloud computer." Accordingly, the use of user-defined template structures, incorporating user specifications, for both the allocation and deallocation of virtual machine resources was described and acknowledged to be a novel and unconventional solution at the time.

38. On March 11, 2014, the U.S. Patent and Trademark Office duly and lawfully issued United States Patent No. 8,671,132 ("the '132 Patent"), entitled "System, Method, and Apparatus for Policy-Based Data Management." A true and correct copy of the '132 Patent is attached hereto as **Exhibit 4**.

39. Daedalus is the owner and assignee of all right, title, and interest in and to the '132 Patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of it.

40. The ’132 Patent describes, among other things, novel systems and methods that improve data management by prioritizing file storage operations to allow remote clients (or end users) using different computing platforms to have more efficient and less expensive access. These inventive technological improvements solved then-existing problems in the field of data storage systems. For example, prior to the invention of the ’132 Patent, distributed storage systems’ ability to automatically allocate resources to prioritize operations was severely limited. For example, existing systems suffered from saturation when many users simultaneously store, retrieve, or move data on the distributed storage system. Another problem was the lack of a method for prioritizing operations, resulting in unnecessary delays in the performance of the more important operations. Additionally, existing distributed storage systems were not capable of storing data using prioritized operations within multiple platforms. Existing systems also did not permit a user to automatically select between multiple storage options when generating files or account for the different requirements placed on these files. Yet another problem is the great variation in the equipment available to store data, wherein some files are stored in a manner that provides insufficient performance, while others take up comparatively expensive storage capacity that provides an unnecessarily expensive level of performance. (*See* Ex. 4, 1:24-2:3).

41. The ’132 Patent overcomes these drawbacks and improves the functioning of a computer system, for example, by describing novel and inventive systems in which files in a data storage system are automatically processed according to the rules designated for selecting a service class and/or storage pool for a file based on the attributes of the file. In one aspect, the ’132 Patent describes an improved policy-based data management system that “prioritize files within the network, with clients that operate based on a plurality of different operating platforms…[and]…intelligently stores files in storage pools with a variety of performance levels

based policies and the nature of the storage pools." (Ex. 4, at 2:8-13). The claims of the '132 Patent are directed to specific techniques, using a file evaluation module, to apply service rules that evaluate the attributes of a client file in order to assign an appropriate classification method.

42. For example, Figure 2 is a schematic block diagram illustrating one concept of a policy implementation:

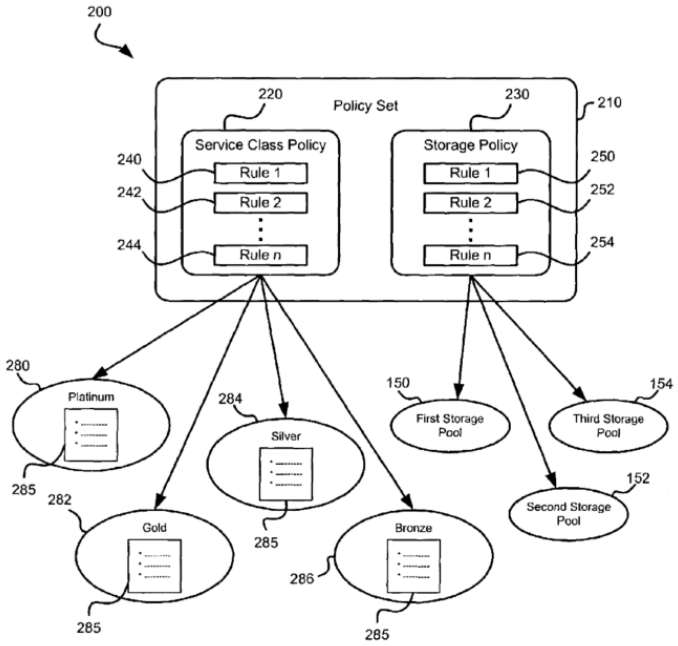


Fig. 2

43. In this diagram, policy-based management is carried out through the use of a policy set, which may include several different types of policies. The policy set is applied to each file and thus the file's attributes are used to classify the file accordingly. (*See* Ex. 4, at 7:5-20). As depicted in Figure 2, the policy set 210 includes a service class policy 220 and a storage pool policy 230. The service class policy 220 includes at least one service class rule that dictates what service class is applied to a file with a given attribute. For example, the service class policy 220 includes a first rule 240, a second rule 242, and other rules through an nth rule 244. Each of the rules 240, 242, 244 in one embodiment comprises a statement such as "If a given file

attribute is X, the file receives service class Y." (*Id.*, at 7:31-40). The storage policy 230 similarly has at least one storage pool rule that dictates which of the storage pools 150, 152, 154 should receive a file with a given attribute. (*Id.*, at 7:41-47). The service class policy 220 is used to select from among a plurality of service classes, such as the service classes 280, 282, 284, 286. (*Id.*, at 7:47-52). By way of example, the platinum service class 280 has the highest priority, followed by the gold service class 282, the silver service class 284, and finally, the bronze service class 286. The service class may be a factor in determining the appropriate storage pool. For example, all files with the bronze service class 286 may be stored in the first storage pool 150, while files with the silver service class 284 are stored in the second storage pool 152 for greater speed and data recoverability, and gold service class may be stored in the third storage pool 154 for even greater speed and recoverability. (*See id.*, at 8:25-47).

44. The novel features of the invention are recited in the claims. For example, Claim 15 of the '132 Patent recites:

> A method for handling files within a policy-based data management system, the method comprising:
>
> providing a policy set comprising at least one service class rule;
>
> receiving one or more attributes of a file from one of a plurality of clients, the clients comprising at least two different computing platforms;
>
> applying the service class rule to the file to assign a service class to the file; and
>
> conducting operations on the file in a manner according to the service class.

(Ex. 4, at 16:21-31). Thus, claim 15 of the '132 Patent describes claim elements, individually or as an ordered combination, that were non routine and unconventional at the time of the invention in 2003 and an improvement over prior art, as it provided a way (not previously available) for prioritizing files within a policy-based data management system based on the attributes of the files. For example, prior to the date of the invention, it was unconventional for a system to

include a policy set which is then used to apply the service class rule to assign a service class to a file. Based on the policy-set rules, the files' attributes are evaluated then assigned a novel and unconventional rule-based service class which dictates the handling of the files. The claims of the '132 Patent are unconventional in that they deal with automatically associating a certain policy with a file for management of the file in a storage system.

45. On May 12, 2015, the U.S. Patent and Trademark Office duly and lawfully issued United States Patent No. 9,032,076 ("the '076 Patent"), entitled "Role-Based Access Control System, Method and Computer Program Product." A true and correct copy of the '076 Patent is attached hereto as **Exhibit 5**.

46. Daedalus is the owner and assignee of all right, title, and interest in and to the '076 Patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of it.

47. The '076 Patent describes, among other things, novel techniques that improve the methods for restricting and granting user access to resources, which results in enhanced system security. These inventive technological improvements solved then-existing problems in the field of computer networks and security authorizations that control access to various resources. For example, the '076 Patent explains that, in a computer network, a company may want to control which users have access to particular resources such as servers or storage spaces. Thus, the system usually includes a resource management function which synchronizes and manages access to such resources. The resource manager within the server systems is often assigned the task of security and access control such that users requesting secure data from the resources may be allowed or denied access to that data. Traditionally, access would be provided through the use of access control lists (ACL), whereby users are associated with specific permissions to

access or to interact with various resources. In a classical role-based access control model, the ACLs are used to provide users with specific permissions to access or to interact with various resources. In such systems, whenever a permission within an ACL changes, the ACL must be recreated with the changed permission, thereby lacking the possibility to enforce different access control constraints on individual resource instances. To overcome this problem, solutions prior to the invention of the '076 Patent used extensions to the classical model defining roles to be sets of permissions on individual resources (resource-level role-based access control (RRBAC)). However, then-existing access control models did not provide instance level resource protection. (*See* Ex. 5, at 1:50-2:6). Furthermore, then-existing RRBAC models restrict a role domain associated to a specific role instances to protect exactly one sub-hierarchy of resources of the protected resource hierarchy. (*Id.* at 4:44-48)

48. The '076 Patent overcomes these drawbacks and improves the functioning of a computer system, for example, by disclosing novel and inventive systems in which access to system resources is controlled by assigning roles and super roles to groups of users. The super roles are defined by grouping a set of role instances (permissions on individual resources), wherein each super role contains all permissions assigned to each of the role instances in the grouped set of role instance. The super roles are modified by adding or removing role instances in the grouped set of role instances. In one aspect, the patent discloses an improved role-based access control system "which comprises a role definition system for defining roles to be sets of permissions on individual resources thus forming role instances, respectively, and a super role definition system for defining at least one super role by grouping a set of role instances, wherein each super role contains all permissions contained in the grouped role instances." (Ex. 5, at 2:30-35). The claims of the '076 Patent are directed to specific techniques that nest super roles

wherein each super role contains all permissions assigned to each of the role instances in the grouped set of role instances. The ’076 Patent further defines at least one super role wherein each super role is nested according to a plurality of properties including a name, a parent role, the set of role instances, and an externalization state. Through the patented invention, access control administration complexity is reduced, as well as costs and errors that could result in unintended access control configurations. Moreover, access control delegation flexibility is improved and the disclosed mechanisms can reduce the set of authorized people necessary at a specific point in time thereby improving overall security and auditability. The ’076 Patent also provided a way (not previously available) to modify super roles by adding and/or removing role instances from the grouped set of role instances.

49. The novel features of the invention are recited in the claims. For example, Claim 6 of the ’076 Patent recites:

> A role-based access control method, comprising:
>
> defining roles to be sets of permissions on individual resources, thus forming role instances, respectively;
>
> assigning at least one set of role instances to at least one group and assigning the at least one group to at least one super role; and
>
> nesting each super role according to a plurality of properties including a name, a parent role, the set of role instances, and an externalisation state,
>
> wherein each super role is modified by adding or removing the role instances from the at least one group.

(Ex. 5, at 14:20-30). Claim 6 of the ’076 Patent describes claim elements, individually or as an ordered combination, that were non routine and unconventional at the time of the invention in 2005 and an improvement over prior art, as it provided a way (not previously available) to operate a role based access control system. For example, in a system for defining roles, it was unconventional to define at least one super role wherein each super role is nested according to a

plurality of properties including a name, a parent role, the set of role instances, and an externalization state.

50. The ’494, ’886, ’612, ’132, and ’076 Patents are referred to hereinafter as “the Asserted Patents.”

51. Each of the Asserted Patents are presumed valid under 35 U.S.C. § 282.

52. Daedalus has complied with the requirements of 35 U.S.C. § 287 with respect to the Asserted Patents.

Oracle’s Use of the Patented Technology

53. Oracle Corporation is a multinational computer technology company founded in 1977. On January 28, 2010, Oracle acquired Sun Microsystems, Inc. (“Sun”). Sun is now Oracle America, Inc., a subsidiary of Oracle Corporation. Oracle’s venture into Cloud Computing started in 2012 with the launch of Oracle Cloud. (*See, e.g.*, https://www.theregister.co.uk/2012/06/07/oracle_cloud_rehash_platinum_ services/). Oracle’s Cloud Computing business grew in 2016 with its acquisition of NetSuite for nearly $10 billion. Oracle is now one of the largest cloud computing companies in the world.

54. On information and belief, Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States various methods and/or products relating to cloud infrastructure, cloud management, database management, data processing, access control, and data management products, systems, and applications, which infringe the Asserted Patents.

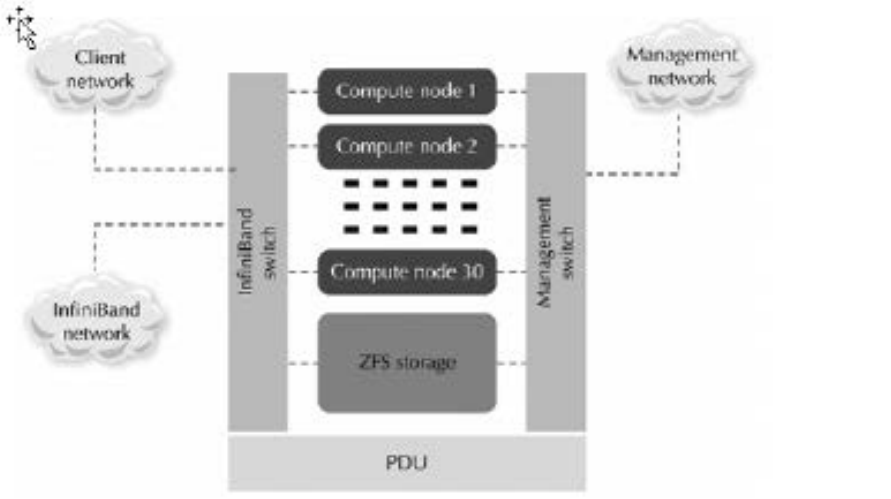
55. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Oracle Exalogic, which is delivered as a rack of hardware and contains ZFS Storage (Storage Area Network/SAN).

56. Oracle makes, uses, sells, and/or offers to sell in the United States and/or imports into the United States the Oracle FS1-2 Flash Storage Systems.

57. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Oracle Enterprise Manager, also sometimes known as the Enterprise Manager Cloud Control.

58. On information and belief, the Oracle Enterprise Manager Cloud Control can be used to monitor Oracle SANs, including the Exalogic ZFS Storage, FS1-2, as well as other Oracle or other third-party components.

59. Oracle Enterprise Manager Cloud Control is used for managing the complete enterprise IT solution, including solutions that implement aa SAN architecture, e.g. SAN System. (*See, e.g.*, Overview of Oracle Enterprise Manager Cloud Control 13c, available at https://docs.oracle.com/cd/E63000_01/EMCON/overview.htm#EMCON110; Enterprise Manager Cloud Control Administrator's Guide, available at https://docs.oracle.com/cd/E63000_01/EMADM/chapemstorage.htm#EMADM9515. The Oracle Exalogic solution is one example that implements a SAN architecture which can work with the Oracle Enterprise Manager. *See also:*



Description of "Figure 1-1 Exalogic Hardware Architecture" (img_text/GUID-78BB9C5D-8A49-4A53-B47C-E7894A1E0541-default.htm)

Fusion Middleware Enterprise Deployment Guide for Exalogic, available at https://docs.oracle.com/cd/E18476_01/doc.220/e64181/GUID-41BD511A-6841-48AE-8D68-4988243CB9A5.htm#EXADG-GUID-24481FA7-7A08-4964-81AD-1418EE2967A8.

60. In addition to Exalogic, Oracle markets and uses Enterprise Manager with a variety of Oracle designed Storage Area Networks and appliances (including for example FS1-2 Flash storage systems), as well as with custom SANs implemented by Oracle's customers.

61. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Oracle Data Guard software application. Data Guard is included with Oracle Database Enterprise Edition.

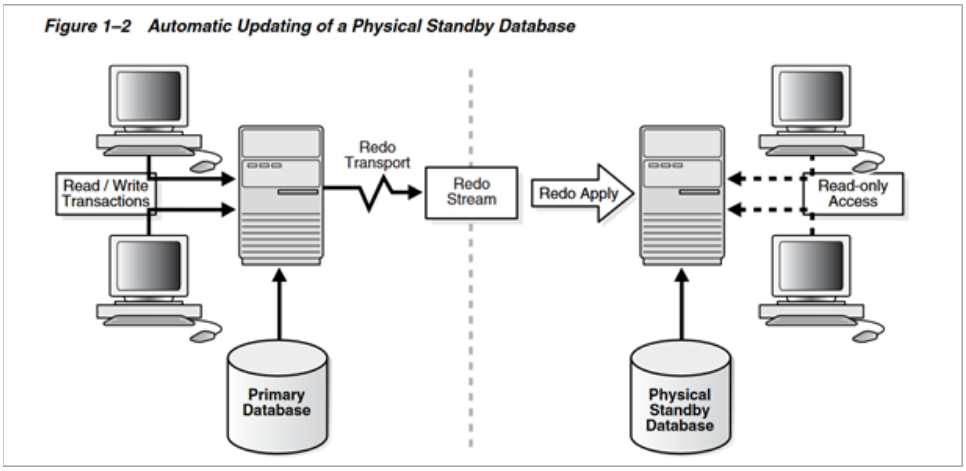
62. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States Oracle GoldenGate software application. GoldenGate software works with both Oracle Databases and certain non-Oracle databases.

63. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States Oracle Databases. Oracle Data Guard and Oracle GoldenGate are Oracle Database features used in Maximum Availability Architecture solutions.

64. On information and belief, Oracle acquired GoldenGate Software, Inc. on or around July 23, 2009. Oracle combined GoldenGate functionality with Oracle Database (10g and later) and published a first Release Note on or around October 2009.

65. Oracle Database with Oracle Data Guard and Oracle GoldenGate provides a critical database server that includes a primary server and a standby server (secondary server), with Oracle Data Guard helping to backup transaction logs on the standby database server and Oracle GoldenGate helping to replicate data among multiple remote servers. (*See, e.g.*, Oracle Data Guard, Concepts and Administration (February 2014), at 30, available at

https://docs.oracle.com/cd/E11882_01/server.112/e41134.pdf.; *see also Features* for Maximizing Availability, available at https://docs.oracle.com/database/121/HAOVW/hafeatures.htm #HAOVW11953; *see also:*



Oracle Data Guard, Concepts and Administration (February 2014), at 30).

66. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Oracle Cloud Infrastructure.

67. Oracle Cloud Infrastructure offers multiple virtual machine instances in a cloud computing environment and has an autoscaling feature which allows a number of virtual machine instances to be adjusted based on the CPU-utilization. (*See, e.g.*, Autoscaling, available at https://docs.cloud.oracle.com/enus/iaas/Content/Compute/Tasks/ autoscalinginstancepools.htm; *see also* Overview of the Compute Service, available at https://docs.cloud.oracle.com/en-us/iaas/Content/Compute/Concepts/computeoverview.htm).

68. On information and belief, Oracle's Cloud Infrastructure's autoscaling features are one of the reasons Zoom recently selected Oracle Cloud Infrastructure for its Core Online Meeting Service. (https://www.oracle.com/corporate/pressrelease/zoom-selects-oracle-to-support-growth-042820.html)

69. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Oracle Hierarchical Storage Manager ("HSM") file system.

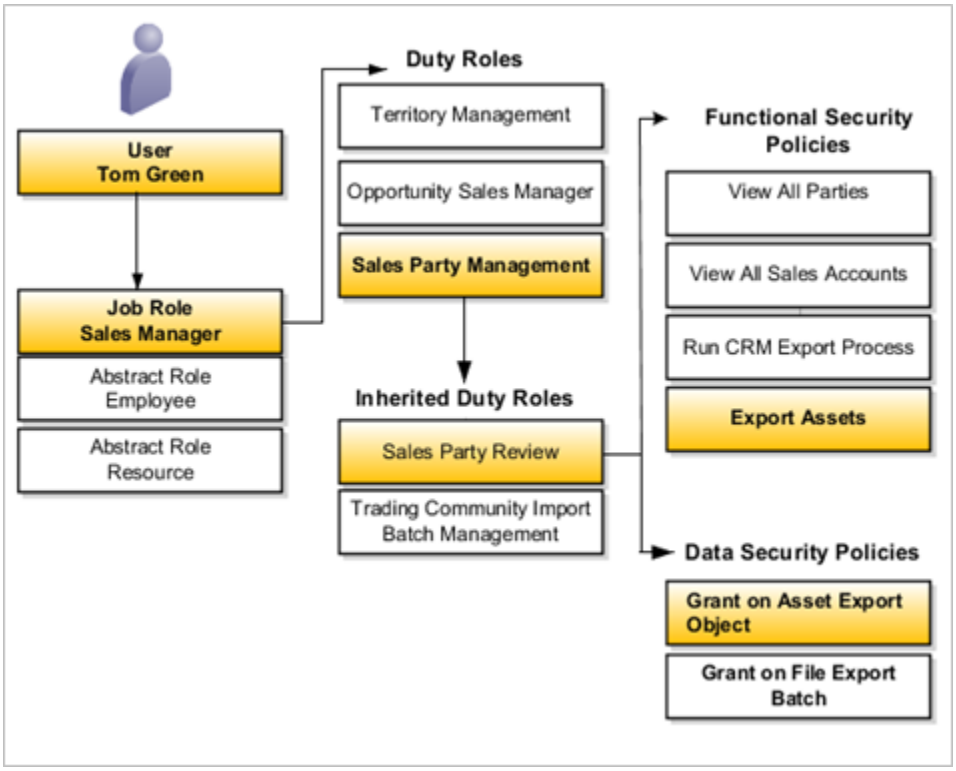
70. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Oracle StorageTek QFS which can be combined with the HSM.

71. Oracle Hierarchical Storage Manager ("HSM") file system is a data management system that backs up files based on user defined policies. (*See, e.g.,* Oracle Hierarchical Storage Manager and StorageTek QFS Software (July 2019), at 21, available at https://docs.oracle.com /cd/E71197_01/SAMIC/E78138-07.pdf). Oracle HSM has a flexible, proactive policy-based storage tiering that streamlines data management processes. "[B]y combining Oracle HSM with Oracle's StorageTek QFS shared file system, data management infrastructure can be optimized." (*See, e.g.*, Oracle Hierarchical Storage Manager and StorageTek QFS, at 1, available at http://www.oracle.com/us/products/servers-storage/storage/faq-storagetek-sam-2285024.pdf).

72. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Oracle Sales Cloud.

73. Oracle makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States numerous products that work with Fusion Applications with role-based access control. One such product is the Oracle Sales Cloud.

74. Oracle Sales Cloud provides role-based access control for better accessibility of application resources. Oracle Sales Cloud is integrated with Fusion Applications to provide better services to users, including preventing unauthorized access to application resources. (*See, e.g.,* Understanding Role-Based Security, available at https://docs.oracle.com/cd/E83857_01/ saas/sales/r13-update17d/fafsi/understanding-role-based-security.html; *see also*:



Creating Job, Abstract, and Duty Roles, available at https://docs.oracle.com/cd/E83857_01/saas/sales/r13-update17d/oscus/creating-job-abstract-and-duty-roles.html).

FIRST COUNT

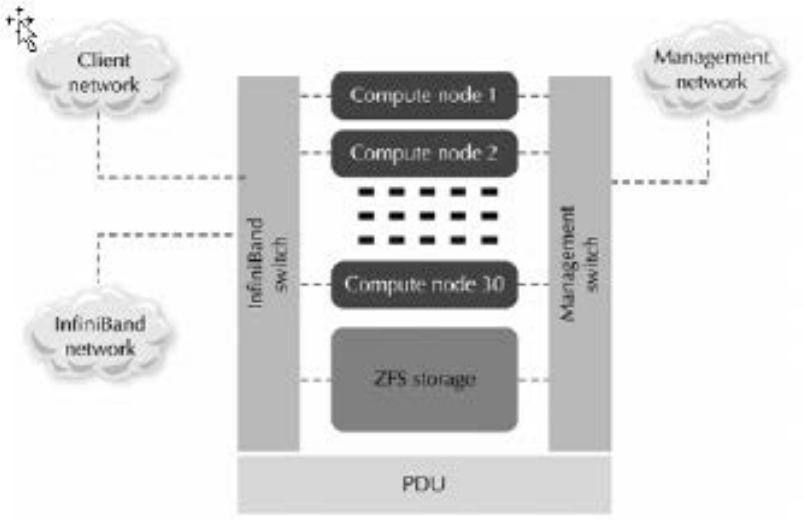
(Infringement of U.S. Patent No. 6,920,494)

75. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-74 of this Complaint as though fully set forth herein.

76. On information and belief, Oracle has directly infringed and continues to directly infringe one or more claims of the ’494 Patent, including at least claim 1 of the ’494 Patent, in the state of Texas, in this judicial district, and elsewhere in the United States by, among other things, making, using, selling, offering for sale, and/or importing into the United States products that embody one or more of the inventions claimed in the ’494 Patent, including but not limited to the above-identified Oracle Enterprise Manager Cloud Control, with SAN System, and Oracle

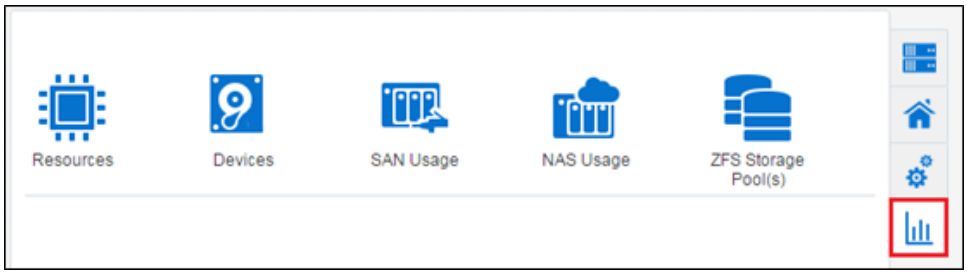
Exalogic Products, FS1-2 Flash Storage System, and all reasonably similar products ("the '494 Accused Products"), in violation of 35 U.S.C. § 271(a).

77. As an example, the '494 Accused Products, including Oracle Enterprise Manager Cloud Control with Oracle Exalogic, implement Storage Area Networks ("SAN"). Specifically, Oracle Enterprise Manager Cloud Control manages a cloud infrastructure that implements a "storage area network." Oracle Enterprise Manager Cloud control can be implemented with Oracle Exalogic having SAN storage device. (*See, e.g.*, Fusion Middleware Enterprise Deployment Guide for Exalogic; *see also:*



Description of "Figure 1-1 Exalogic Hardware Architecture" (img_text/GUID-78BB9C5D-8A49-4A53-847C-E7694A1E0541-default.htm)

Id.). Oracle Enterprise Manager Cloud Control uses a Charts tab as part of its graphical user interface, displaying usage statistics of SAN. (*See, e.g.*, About Charts, available at https://docs.oracle.com/cd/E63000_01/EMADM/chapemstorage.htm#EMADM9515; s*ee also:*



Id.).

78. Oracle ZFS Storage is a component of Oracle Exalogic which is connected in a SAN. (*See* Fusion Middleware Enterprise Deployment Guide for Exalogic). Oracle ZFS Storage has "one or more regions forming at least a portion of the SAN, each region having one or more components, and the components including one or more digital processors and one or more storage devices." For instance, in Oracle ZFS Storage, zones are set up in the SAN switch(es). (*See, e.g.*, Understanding the Use of Fibre Channel in the Oracle ZFS Storage Appliance, available at https://www.oracle.com/technetwork/server-storage/sun-unified-storage/documentation/o12-019-fclun-7000-rs-1559284.pdf). For example, as shown below, "the colored lines show the logical 'zoned' connections." (*See:*

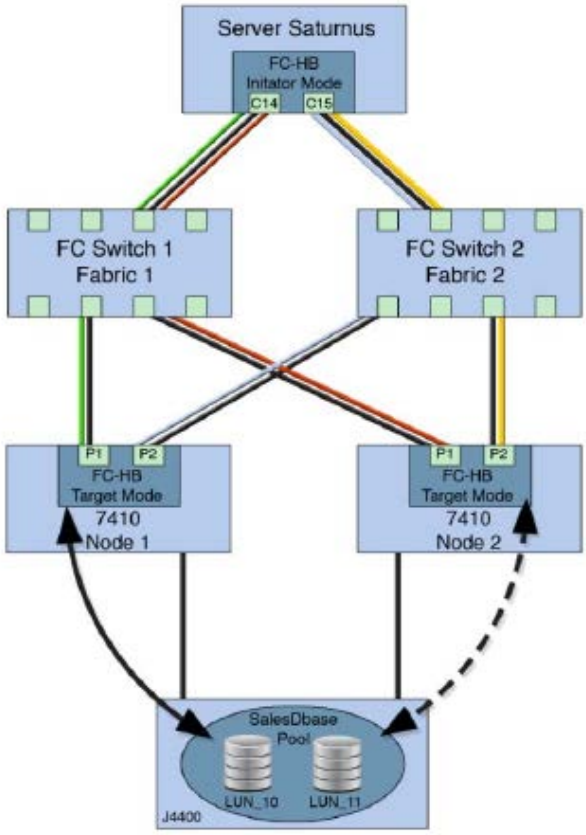
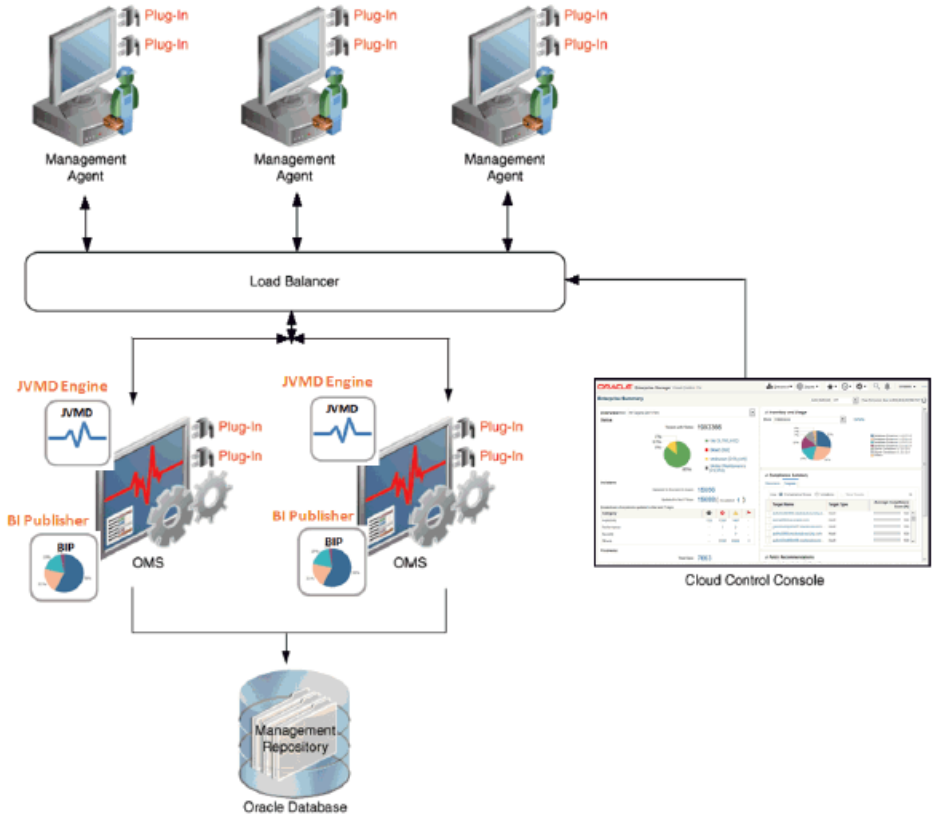


Figure 5. SAN wiring diagram

Id. at Fig. 5).

79. In Oracle Enterprise Manager Cloud Control System, Management Agents along with plug-ins are "one or more scanners that collect, for each region, information regarding the components and their interconnectivity." For instance, Management Agents along with plug-ins are installed to manage and monitor the targets running on the host, including servers, storage appliances, storage for a host, and network resources. (*See, e.g.*, Overview of Enterprise Manager Systems Infrastructure, available at https://docs.oracle.com/en/enterprise-manager/cloud-control/enterprise-manager-cloud-control/13.3.1/emadm/overview-enterprise-manager-systems-infrastructure.html#GUID-8480617C-B985-4C8F-BD0E-9EA63067E6B8; *see also* Cloud Control Extensibility Programmer's Guide (May 2017), available at https://docs.oracle.com/cd/cloud-control-13.3/EMPRG/EMPRG.pdf). A fabric represents the physical network targets including the connection between targets. Oracle Enterprise Manager Cloud Control discovers and manages Ethernet fabrics and InfiniBand fabrics. Datalinks are also discovered, where the datalink layer manages networks that share resources on Fibre Channel Zones. (*See* Fusion Middleware Enterprise Deployment Guide for Exalogic). As a result, "interconnectivity information," such as fabric datalinks are collected when network switch or network-enabled devices are discovered and monitored. (*See id.*). Using the monitored information, the Oracle Enterprise Manager Cloud Control shows information related to the components and the relationship between them. (*See:*

Figure 1-2 Enterprise Manager Cloud Control Architecture



Enterprise Manager Cloud Control Introduction, at Fig. 1-2, available at https://docs.oracle.com/cd/E63000_01/EMCON/overview.htm#CJADBJDH).

80. In Oracle Enterprise Manager Cloud Control System, the Enterprise Manager, a "manager" that is "coupled to one or more scanners," "responds to the collected information to determine a topology of a portion of the SAN spanned by the regions." For instance, the Enterprise Manager allows the user to view the relationship between storage servers and storage clients. Further, the Enterprise Manager responds to the information collected by the Managing Agent and the plug-ins to determine a topology. (*See, e.g.*, Enterprise Manager Cloud Control Introduction, at Fig. 1-2; *see also* About Storage Configuration Topology, available at https://docs.oracle.com/cd/E63000_01/EMADM/chapemstorage.htm#EMADM9565).

81. By making, using, offering for sale, and/or selling products in the United States and/or importing products into the United States, including but not limited to the '494 Accused Products, Oracle has injured Daedalus and is liable to Daedalus for directly infringing one or more claims of the '494 Patent, including without limitation claim 1 pursuant to 35 U.S.C. § 271(a).

82. On information and belief, Oracle is inducing and/or has induced infringement of one or more claims of the '494 Patent, including at least claim 1, as a result of, among other activities, instructing, encouraging, and directing its customers on the use of the '494 Accused Products in an infringing manner in violation of 35 U.S.C. § 271(b). Through its website, instructional guides, and manuals, Oracle provides its customers with detailed explanations, instructions, and information on how to use and implement the '494 Accused Products which demonstrate active steps taken to encourage direct infringement. (*See, e.g.*, Cloud Control Extensibility Programmer's Guide (May 2017), at 2-1; Overview of Oracle Enterprise Manager Cloud Control; About Storage Configuration Topology; Overview of Enterprise Manager Systems Infrastructure; SAN Fundamentals: How Fibre Channel SANs are Built, Secured and Managed (April 2007), at 2). On information and belief, Oracle has had actual knowledge of the '494 Patent at least as of 2010. Despite this knowledge of the '494 Patent, Oracle has continued to engage in activities to encourage and assist its customers in the use of the '494 Accused Products. Thus, on information and belief, Oracle (1) had actual knowledge of the patent; (2) knowingly induced its customers to infringe the patent; and (3) had specific intent to induce the patent infringement.

83. On information and belief, Oracle has known about the '494 Patent and its contents since at least about January 2010 when it acquired Sun and its employees who had

knowledge of the ’494 Patent. On information and belief, Sun and the inventors of U.S. Patent Application Nos. 10/608,882, and 10/127,898 (collectively, “Sun Patent Applications”) knew of the ’494 Patent and its contents when the ’494 Patent was cited in Information Disclosure Statements during the prosecution of the Sun Patent Applications. When Oracle acquired Sun, it also acquired employees from Sun, including inventors of the Sun Patent Applications. Furthermore, on information and belief, the Sun Patent Applications were assigned to Oracle on February 12, 2010. Oracle, having learned of the likelihood of infringement of the ’494 Patent, nevertheless acted in a way that infringed.

84. On information and belief, by using the ’494 Accused Products as encouraged and assisted by Oracle, Oracle’s customers have directly infringed and continue to directly infringe one or more claims of the ’494 Patent, including at least Claim 1. On information and belief, Oracle knew or was willfully blind to the fact that that its actions would induce its customers’ direct infringement of the ’494 Patent.

85. Oracle’s infringement of the ’494 Patent has been and continues to be deliberate and willful, and, this is therefore an exceptional case warranting an award of enhanced damages and attorneys’ fees and costs pursuant to 35 U.S.C. §§ 284-285.

86. On information and belief, Oracle will continue to infringe the ’494 Patent unless enjoined by this Court.

87. As a result of Oracle’s infringement of the ’494 Patent, Daedalus has suffered monetary damages, and seeks recovery, in an amount to be proven at trial, adequate to compensate for Oracle’s infringement, but in no event less than a reasonable royalty with interest and costs. Oracle’s infringement of Daedalus’ rights under the ’494 Patent will continue to

damage Daedalus, causing irreparable harm for which there is no adequate remedy at law, unless enjoined by this Court.

SECOND COUNT

(Infringement of U.S. Patent No. 7,177,886)

88. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-87 of this Complaint as though fully set forth herein.

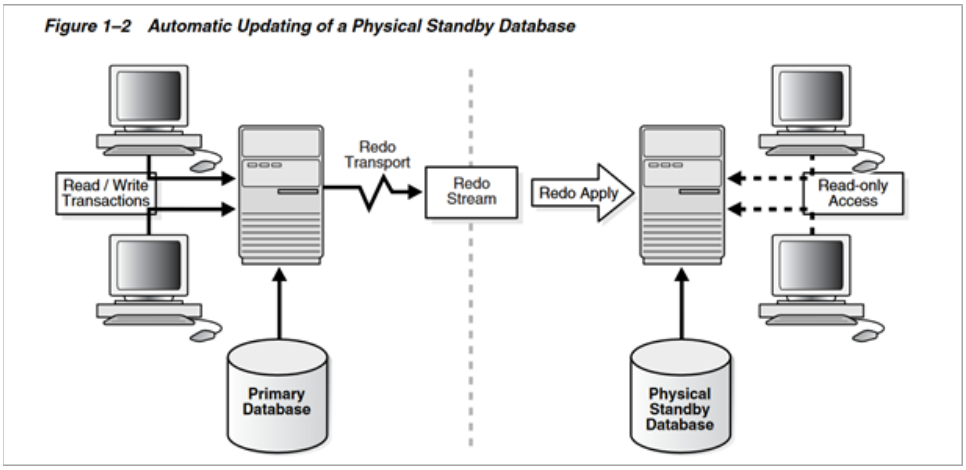
89. On information and belief, Oracle has directly infringed and continues to directly infringe one or more claims of the ’886 Patent, including at least Claim 1 of the ’886 Patent, in the state of Texas, in this judicial district, and elsewhere in the United States by, among other things, making, using, selling, offering for sale, and/or importing into the United States products that embody one or more of the inventions claimed in the ’886 Patent, including but not limited to the above-identified Oracle Database, Oracle Data Guard and Oracle GoldenGate products, and all reasonably similar products (“the ’886 Accused Products”), in violation of 35 U.S.C. § 271(a).

90. As an example, the ’886 Accused Products, such as Oracle Database, comprise of relational database management systems. (*See, e.g.*, Relational Database Management System (RDBMS), available at https://docs.oracle.com/database/121/CNCPT/intro.htm#CNCPT88783). Oracle Database utilizes Oracle Data Guard, to help backup transaction logs on a standby database server, and Oracle GoldenGate, to help replicate data among remote servers. (*See, e.g.*, Features for Maximizing Availability).

91. Oracle Data Guard configuration is “a critical database server” that includes a “primary server” and a “secondary server.” (*See* Oracle Data Guard, Concepts and Administration (February 2014), at 30). The primary server is connected with a “primary

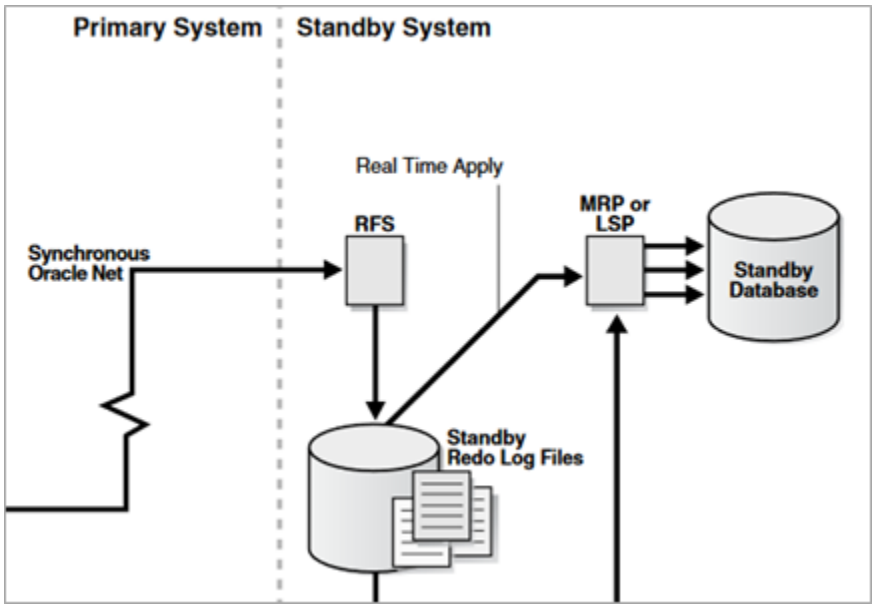
database instance" and the secondary server is connected with a "secondary database instance." (*Id.*). In the Oracle Data Guard configuration, the standby database instance (secondary database instance) backs up and/or "mirrors" the transaction logs of the primary database instance. (*See id.* at 28). "[T]he redo data needed to recover a transaction must be written to both the online redo log and to the standby redo log on at least one synchronized standby database before the transaction commits." (*Id.* at 70). The standby database mirrors "a selected critical database transaction" at the primary database instance. (*See id*.). Once that is done, the standby server "generates an acknowledgement" to the primary database indicating that a selected database transaction at the primary database instance is mirrored at the secondary database instance. (*See id*. at 70).

92. Oracle Data Guard services are comprised of "a critical database server" with redo transport services, which is "a mirroring component" that can communicate with the primary server and the secondary server(s) for transferring redo data. (*See* Oracle Data Guard, Concepts and Administration (February 2014), at 29). Redo transport services "transfer" redo data from the primary database instance to the standby database connected to the secondary server. (*See Id*. at 30 and 73). Redo transport services transmit "database log file entries" to store the log files of the transactions performed in the primary database. (*See id.* at 73). The database log file entries transmitted from the primary database are "logged" to the standby database instance and apply services automatically "apply" the database log file entries to the standby database to maintain consistency with the primary database. (*See id.* at 73; s*ee also:*



Id. at 30).

93. Oracle Data Guard configuration includes a remote file server as a secondary server which is connected to a secondary database and applies and logs the database log file entries to the secondary database instance. (*See, e.g.*, Oracle Data Guard, Concepts and Administration (February 2014), at 90; s*ee also:*



Id. at 30). After the transaction is performed at the primary database included in the Data Guard, the database log file entries of the selected critical database transaction are transferred, applied, and logged to the secondary database instance. (*Id*.). After the database file log entries have

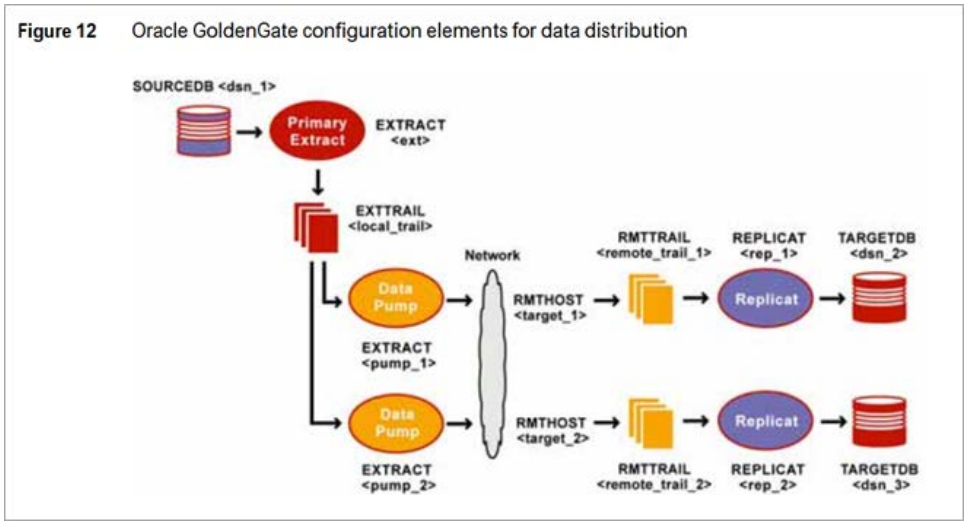
been written to the secondary database, the secondary server sends an acknowledgement signal to the primary server. (*See id*., at 70 and 76).

94. Oracle Data Guard utilizes redo transport services, a mirroring component, that include "a control structure." For example, redo transport services monitor the redo transport status using V$ARCHIVE_DEST_STATUS, which contains a list of parameters related to configuration information for the archived redo log destinations. (*See* Oracle Data Guard, Concepts and Administration (February 2014), at 83). V$ARCHIVE_DEST_STATUS "indexes critical database transactions that are applied and logged at the secondary database instance." (*See id*. at 86; *see also* Oracle Database Reference (August 2015), at 848, available at https://docs.oracle.com/cd/E11882_01/server.112/e40402.pdf). For instance, V$ARCHIVE_DEST_STATUS identifies the thread number and the sequence number of the most recent redo logs that are applied and logged at the standby database instance. (*See* Oracle Data Guard, Concepts and Administration (February 2014), at 86).

95. Oracle Data Guard configuration includes "an acknowledgement signal," "AFFIRM," which acknowledges that a redo transport destination received redo data after writing it to the standby redo log. (*See* Oracle Data Guard, Concepts and Administration (February 2014), at 231). The AFFIRM acknowledgement signal "corresponds to indexing in the control structure," V$ARCHIVE_DEST_STATUS. (*See id*). The AFFIRM acknowledgement signal is used to specify that redo data received from a redo source database is not acknowledged until it has been written to the standby redo log. (*See id.* at 76).

96. Oracle GoldenGate is used for data distribution among multiple servers, which are connected to corresponding database instances, such as TARGETDB. (*See* Oracle GoldenGate,

Windows and UNIX Administrator's Guide (April 2012), at 10, 65, and 67, available at https://docs.oracle.com/cd/E35209_01/doc.1121/e29397.pdf; s*ee also*:



Id. at 67).

97. Oracle GoldenGate "supports" synchronization of a source database to any number of target systems, where "[t]he target location can be a single server disk location, multiple disk locations, or multiple servers and disk locations." (*See* Oracle GoldenGate 12c: Real-Time Access to Real-Time Information (March 2016), at 10, available at http://www.oracle.com/us/products/middleware/data-integration/oracle-goldengate-realtime-access-2031152.pdf).

98. Oracle GoldenGate has a data replicator which "communicates" between a critical database server and multiple other servers responsive to an acknowledgement signal. (*See* Oracle GoldenGate, Windows and UNIX Administrator's Guide (April 2012), at 67). For instance, the Oracle GoldenGate REPLICAT module and EXTRACT module communicate between a SOURCEDB and multiple TARGETDBs to replicate the selected database transaction on at least one of the multiple other servers, after an acknowledgement signal has been transmitted. (*See, e.g., id.* at 14-15, and 67; Oracle Data Guard, Concepts and Administration

(February 2014), at 96; Oracle Data Guard Concepts and Administration, at 5; Data Guard Protection Modes, available at https://docs.oracle.com/cd/E18283_01/server.112/e17022/protection.htm; Oracle GoldenGate Reference for Oracle GoldenGate for Windows and Unix, available at https://docs.oracle.com/goldengate/1212/gg-winux/GWURF/gg_parameters017.htm#GWURF413).

99. By making, using, offering for sale, and/or selling products in the United States and/or importing products into the United States, including but not limited to the '886 Accused Products, Oracle has injured Daedalus and is liable to Daedalus for directly infringing one or more claims of the '886 Patent, including without limitation claim 1 pursuant to 35 U.S.C. § 271(a).

100. On information and belief, Oracle is inducing and/or has induced infringement of one or more claims of the '886 Patent, including at least claim 1, as a result of, among other activities, instructing, encouraging, and directing its customers on the use of the '886 Accused Products in an infringing manner in violation of 35 U.S.C. § 271(b). Through its website, instructional guides, and manuals, Oracle provides its customers with detailed explanations, instructions, and information on how to use and implement the '886 Accused Products which demonstrate active steps taken to encourage direct infringement. (*See*, *e.g.*, Oracle Data Guard, Concepts and Administration (February 2014), at 30; Oracle GoldenGate, Windows and UNIX Administrator's Guide (April 2012); Oracle GoldenGate 12c: Real-Time Access to Real-Time Information (March 2016); Oracle GoldenGate Reference for Oracle GoldenGate for Windows and Unix). On information and belief, Oracle has had knowledge of the '886 Patent at least as of 2008. Despite this knowledge of the '886 Patent, Oracle has continued to engage in activities to encourage and assist its customers in the use of the '886 Accused Products. Thus, on

information and belief, Oracle (1) had actual knowledge of the patent; (2) knowingly induced its customers to infringe the patent; and (3) had specific intent to induce the patent infringement.

101. On information and belief, Oracle has known about the ’886 Patent and its contents since at least about October 2008. On information and belief, Oracle and the inventors of U.S. Patent No. 7,693,882 (“Oracle’s ’882 Patent”) knew of the ’886 Patent and its contents when the application that eventually became the ’886 Patent was cited as a reference in a non-final rejection by the patent examiner during the prosecution of Oracle’s ’882 Patent. Furthermore, on information and belief, Oracle and the inventors of U.S. Patent No. 8,122,108 (“Oracle’s ’108 Patent”) also knew of the ’866 Patent and its contents when the application that eventually became the ’866 Patent was again cited as a reference by the patent examiner in an office action dated December 27, 2010 during the prosecution of Oracle’s ’108 Patent. Lastly, the application that eventually became the ’866 Patent was also cited in an Information Disclosure Statement during the prosecution of a third Oracle patent: U.S. Patent No. 9,384,103. Oracle, having learned of the likelihood of infringement of the ’886 Patent, nevertheless acted in a way that infringed.

102. On information and belief, by using the ’886 Accused Products as encouraged and assisted by Oracle, Oracle’s customers have directly infringed and continue to directly infringe one or more claims of the ’886 Patent, including at least claim 1. On information and belief, Oracle knew or was willfully blind to the fact that that its actions would induce its customers’ direct infringement of the ’886 Patent.

103. Oracle’s infringement of the ’886 Patent has been and continues to be deliberate and willful, and, this is therefore an exceptional case warranting an award of enhanced damages and attorneys’ fees and costs pursuant to 35 U.S.C. §§ 284-285

104. On information and belief, Oracle will continue to infringe the ’886 Patent unless enjoined by this Court.

105. As a result of Oracle’s infringement of the ’886 Patent, Daedalus has suffered monetary damages, and seeks recovery, in an amount to be proven at trial, adequate to compensate for Oracle’s infringement, but in no event less than a reasonable royalty with interest and costs. Oracle’s infringement of Daedalus’ rights under the ’886 Patent will continue to damage Daedalus, causing irreparable harm for which there is no adequate remedy at law, unless enjoined by this Court.

THIRD COUNT

(Infringement of U.S. Patent No. 8,572,612)

106. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-105 of this Complaint as though fully set forth herein.

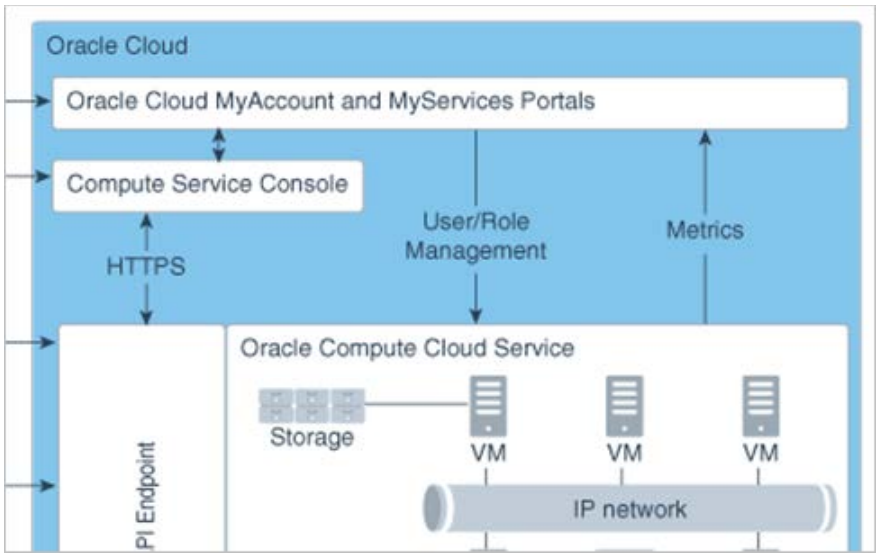
107. On information and belief, Oracle has directly infringed and continues to directly infringe one or more claims of the ’612 Patent, including at least claim 1 of the ’612 Patent, in the state of Texas, in this judicial district, and elsewhere in the United States by, among other things, making, using, selling, offering for sale, and/or importing into the United States products that embody one or more of the inventions claimed in the ’612 Patent, including but not limited to the above-identified Oracle Cloud Infrastructure products, and all reasonably similar products (“the ’612 Accused Products”), in violation of 35 U.S.C. § 271(a).

108. As an example, the ’612 Accused Products, such as Oracle Cloud Infrastructure, contain “a method of autonomic scaling of virtual machines in a cloud computing environment” which allows a number of virtual machine instances to be adjusted based on the CPU-utilization. (*See* Autoscaling; Overview of the Compute Service). Oracle Cloud Infrastructure is “a cloud

computing environment" which provides a set of cloud services. (*See* Welcome to Oracle Cloud Infrastructure, available at https://docs.cloud.oracle.com/en-us/iaas/Content/GSG/Concepts/baremetalintro.htm). Oracle Cloud Infrastructure offers multiple virtual machine instances, the number of which can be automatically adjusted by its autoscaling feature. (*See* Overview of the Compute Service; *see also* Autoscaling).

109. Oracle Cloud Infrastructure's virtual machines contain "modules of automated computing machinery installed upon computers disposed within a data center." For instance, the Oracle-provided image is a template of a virtual hard drive, which determines the operating system (automated computing machinery) and other software for the virtual machine instances. (*See* Overview of the Compute Service). Oracle cloud computers are deployed in data center regions. (*See* Oracle Cloud Infrastructure Data Regions, available at https://www.oracle.com/in/cloud/architecture-and-regions.html). That is, these automated computing machinery images for virtual machine instances are running on cloud computers deployed in Oracle's data center.

110. Oracle Linux for Oracle Cloud Infrastructure is a "cloud operating system" in Oracle's cloud computing environment. (*See* Oracle Linux for Oracle Cloud Infrastructure, at 2, available at https://www.oracle.com/a/ocom/docs/oracle-linux-for-cloud-infrastructure-faq.pdf). Oracle's cloud computing environment further comprises Oracle Compute Cloud Service, "a data center administration server operably coupled to the virtual machine" instances. (*See* About Compute Classic, available at https://docs.oracle.com/en/cloud/iaas/compute-iaas-cloud/stcsg/compute-opc.html). That is, Oracle Compute Cloud Service helps to launch, manage, and terminate multiple virtual machine instances on Oracle cloud. (*See also*:



Id.).

111. Oracle Linux for Oracle Cloud Infrastructure is a "cloud operating system" that "deploys virtual machines." For instance, Oracle Cloud Infrastructure helps a user launch virtual machine instances on Oracle cloud. (*See* Oracle Linux for Oracle Cloud Infrastructure, at 2; *see also* Creating an Instance, available at https://docs.cloud.oracle.com/enus/iaas/Content/Compute/Tasks/launchinginstance.htm; *see also* Creating an Instance Pool, available at https://docs.cloud.oracle.com/en-us/iaas/Content/Compute/Tasks/creatinginstancepool.htm). Via Oracle Cloud Infrastructure's autoscaling feature, the virtual machines are automatically adjusted based on a user defined configuration. (*See* Creating an Instance Pool). The autoscaling configuration "flags the instance of a virtual machine for autonomic scaling" by, *e.g.*, creating an autoscaling configuration indicating that the instance is eligible for scaling and specifying the thresholds that the performance metric must reach to trigger a scaling event. (*See* Autoscaling).

112. Oracle Cloud Infrastructure's autonomic scaling "includes terminating and executing a data processing workload on the instance of a virtual machine." For instance, users select a scale-Out Operator and Threshold percentage at which to increase the number of instances to the instance pool. Similarly, users select a Scale-In Operator and Threshold

Percentage at which to decrease the number of instances from the instance pool. (*See* Autoscaling).

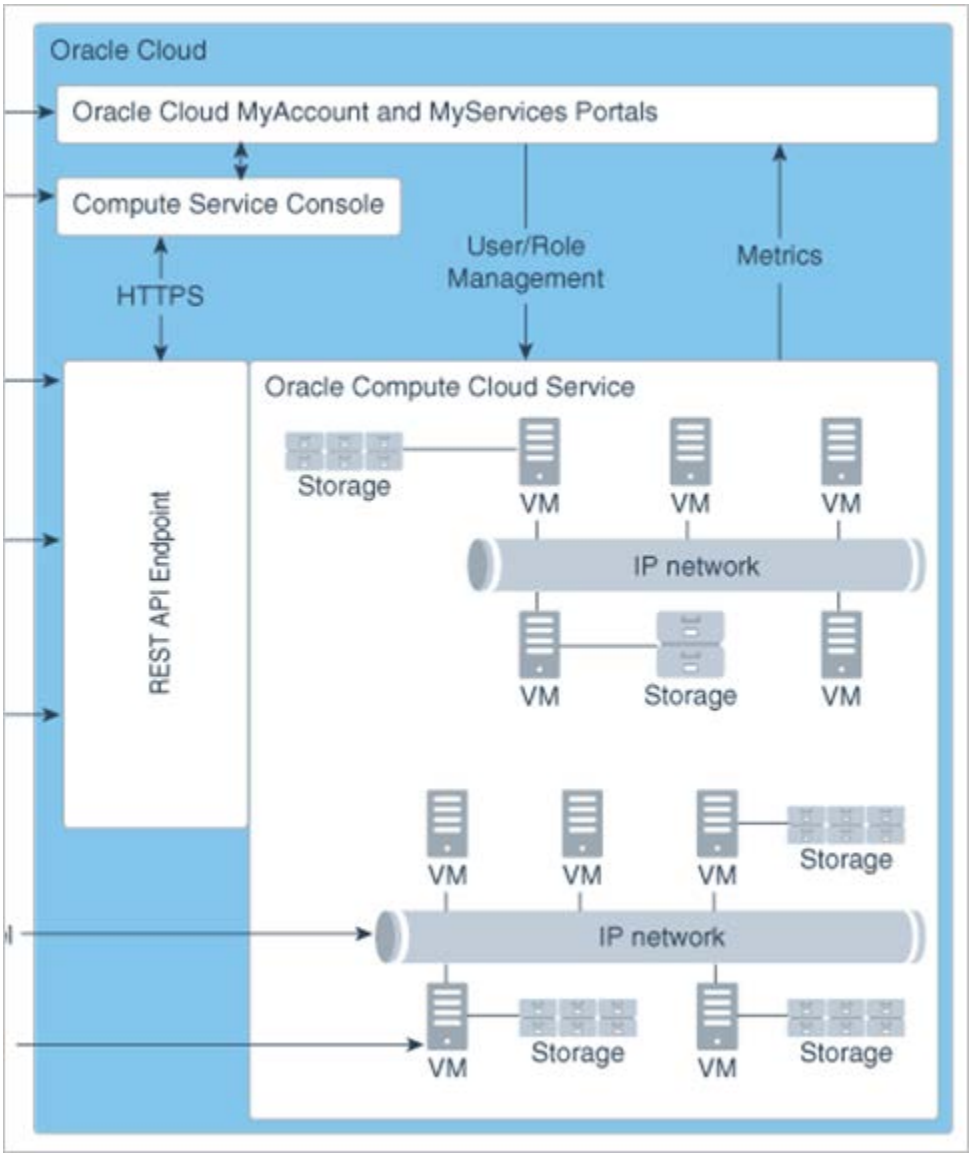
113. Oracle Linux for Oracle Cloud Infrastructure "monitors" the "operating characteristics" of the virtual machine instances. Performance metrics monitor the performance of the virtual instances such as CPU utilization. (*See* Oracle Linux for Oracle Cloud Infrastructure, at 2; *see* Autoscaling).

114. Oracle Linux for Oracle Cloud Infrastructure "deploys an additional instance of the virtual machine if a value of an operating characteristic exceeds a first predetermined threshold value." For example, the Oracle Linux operating system can be configured to increase the number of virtual machines by 10 instances when the CPU utilization exceeds 90%. (*See, e.g.*, Oracle Linux for Oracle Cloud Infrastructure, at 2; *see also* Autoscaling; *see also* Creating an Instance Pool). And when the CPU utilization increases and new virtual machine instances are added to the instance pool, incoming traffic is routed to the new virtual machine instances, thereby "executing a portion of the data processing workload on the additional instance of the virtual machine." (*See id.*).

115. Oracle Cloud Infrastructure "terminates the operation of the additional instance of the virtual machine if a value of an operating characteristic declines below a second predetermined threshold value." For example, the autoscaling can be configured so that when CPU utilization is less than 50%, the Oracle Cloud Infrastructure removes 5 of virtual machine instances from the instance pool. (*See* Autoscaling).

116. Oracle Linux acts as "a module of automated computing machinery" on which Oracle Cloud Infrastructure runs. (*See* Overview of the Compute Service; *see also* Oracle Linux for Oracle Cloud Infrastructure). Oracle Cloud Infrastructure, running on Oracle Linux

operating system, consists of MyServices Portals, "self service portals" where all virtual machine specifications are received from the user, and Oracle Compute Service Console, "a deployment engine" that deploys and manages virtual machine instances. (*See* About Compute Classic; *and see* Oracle Linux for Oracle Cloud Infrastructure, at 2; s*ee also*:



About Compute Classic).

117. In Oracle Cloud Infrastructure, "self service portal user specifications" are received from the user. For instance, the user provides virtual machine specifications, such as

the number of CPUs, amount of memory, and other resources in the Change Shape section in the MyServicesPortal to create a virtual machine instance. (*See, e.g.* Creating an Instance; *see also* Right-Size Your VM Instances to Support Your Workload, available at https://blogs.oracle.com/cloud-infrastructure/right-size-your-vm-instances-to-support-your-workload; *see also* Compute Shapes, available at https://docs.cloud.oracle.com/en-us/iaas/Content/Compute/References/computeshapes.htm; *see also* Right-Size Your VM Instances to Support Your Workload; *see also* Compute Shapes). Further, the self service portal "passes" the user specifications to the "deployment engine" to create the virtual machine instances. Oracle Cloud MyServicesPortals communicate with a Compute Service Console to pass user specifications, such as Shapes, thereby allowing the Computer Service Console to check whether the specifications are compatible. (*See, e.g.*, About Compute Classic; *see also* Changing the Shape of an Instance, available at https://docs.cloud.oracle.com/en-us/iaas/Content/Compute/Tasks/resizinginstances.htm).

118. Oracle Service Console, "a deployment engine" in Oracle Cloud Infrastructure, "implements and passes to the data center administration server" the "virtual machine template with the user specifications." (*See* Oracle Compute Cloud Service). Specifically, after collecting virtual machine specifications from MyServicesPortals, Oracle Service Console passes the Shapes via HTTPS to the Oracle Compute Cloud Service. (*See* About Compute Classic; *see also* Welcome to Oracle Cloud Infrastructure; *see also* Overview of the Compute Service). Oracle Compute Cloud Service creates virtual machine instances based on the virtual machine specifications defined in the Shape. (*See* Overview of the Compute Service; *see also* Compute Shapes).

119. Oracle Compute Cloud Service "calls" Oracle Linux KVM, "a hypervisor" on Oracle Cloud, "to install the virtual machine template as an instance of a virtual machine on the cloud computer." (*See* Getting Started: Oracle Linux KVM Image for Oracle Cloud Infrastructure, available at https://community.oracle.com/docs/DOC-1023677; *see also* Announcing Oracle Linux Virtualization Manager, available at https://blogs.oracle.com/virtualization/announcing-oracle-linux-virtualization-manager; *see also* Overview of the Compute Service). For instance, in Oracle Cloud Infrastructure, Oracle Linux KVM hypervisor creates virtual machine instances on Oracle Cloud based on the Shapes. (*See id*.; *see also* Getting Started: Oracle Linux KVM Image for Oracle Cloud Infrastructure).

120. By making, using, offering for sale, and/or selling products in the United States and/or importing products into the United States, including but not limited to the '612 Accused Products, Oracle has injured Daedalus and is liable to Daedalus for directly infringing one or more claims of the '612 Patent, including without limitation claim 1 pursuant to 35 U.S.C. § 271(a).

121. On information and belief, Oracle is inducing and/or has induced infringement of one or more claims of the '612 Patent, including at least claim 1, as a result of, among other activities, instructing, encouraging, and directing its customers on the use of the '612 Accused Products in an infringing manner in violation of 35 U.S.C. § 271(b). Through its website, instructional guides, and manuals, Oracle provides its customers with detailed explanations, instructions, and information on how to use and implement the '612 Accused Products which demonstrate active steps taken to encourage direct infringement. (*See, e.g.*, Creating an Instance; Right-Size Your VM Instances to Support Your Workload; Compute Shapes; Getting Started: Oracle Linux KVM Image for Oracle Cloud Infrastructure; Announcing Oracle Linux

Virtualization Manager). On information and belief, Oracle has had knowledge of the '612 Patent at least as of 2014. Despite this knowledge of the '612 Patent, Oracle has continued to engage in activities to encourage and assist its customers in the use of the '612 Accused Products. Thus, on information and belief, Oracle (1) had actual knowledge of the patent; (2) knowingly induced its customers to infringe the patent; and (3) had specific intent to induce the patent infringement.

122. On information and belief, Oracle has known about the '612 Patent and its contents since at least about April 2014. On information and belief, Oracle and the inventors of U.S. Patent Application No. 13/423,024 knew of the '612 Patent and its contents when the '612 Patent was cited as a reference by the Patent examiner during the prosecution of U.S. Patent Application No. 13/423,024. Oracle, having learned of the likelihood of infringement of the '612 Patent, nevertheless acted in a way that infringed.

123. On information and belief, by using the '612 Accused Products as encouraged and assisted by Oracle, Oracle's customers have directly infringed and continue to directly infringe one or more claims of the '612 Patent, including at least claim 1. On information and belief, Oracle knew or was willfully blind to the fact that that its actions would induce its customers' direct infringement of the '612 Patent.

124. Oracle's infringement of the '612 Patent has been and continues to be deliberate and willful, and, this is therefore an exceptional case warranting an award of enhanced damages and attorneys' fees and costs pursuant to 35 U.S.C. §§ 284-285.

125. On information and belief, Oracle will continue to infringe the '612 Patent unless enjoined by this Court.

126. As a result of Oracle's infringement of the '612 Patent, Daedalus has suffered monetary damages, and seeks recovery, in an amount to be proven at trial, adequate to compensate for Oracle's infringement, but in no event less than a reasonable royalty with interest and costs. Oracle's infringement of Daedalus' rights under the '612 Patent will continue to damage Daedalus, causing irreparable harm for which there is no adequate remedy at law, unless enjoined by this Court.

FOURTH COUNT

(Infringement of U.S. Patent No. 8,671,132)

127. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-126 of this Complaint as though fully set forth herein.

128. On information and belief, Oracle has directly infringed and continues to directly infringe one or more claims of the '132 Patent, including at least Claim 15 of the '132 Patent, in the state of Texas, in this judicial district, and elsewhere in the United States by, among other things, making, using, selling, offering for sale, and/or importing into the United States products that embody one or more of the inventions claimed in the '132 Patent, including but not limited to the above-identified Oracle Hierarchical Storage Manager and Oracle StorageTek QFS products, and all reasonably similar products (" the '132 Accused Products"), in violation of 35 U.S.C. § 271(a).

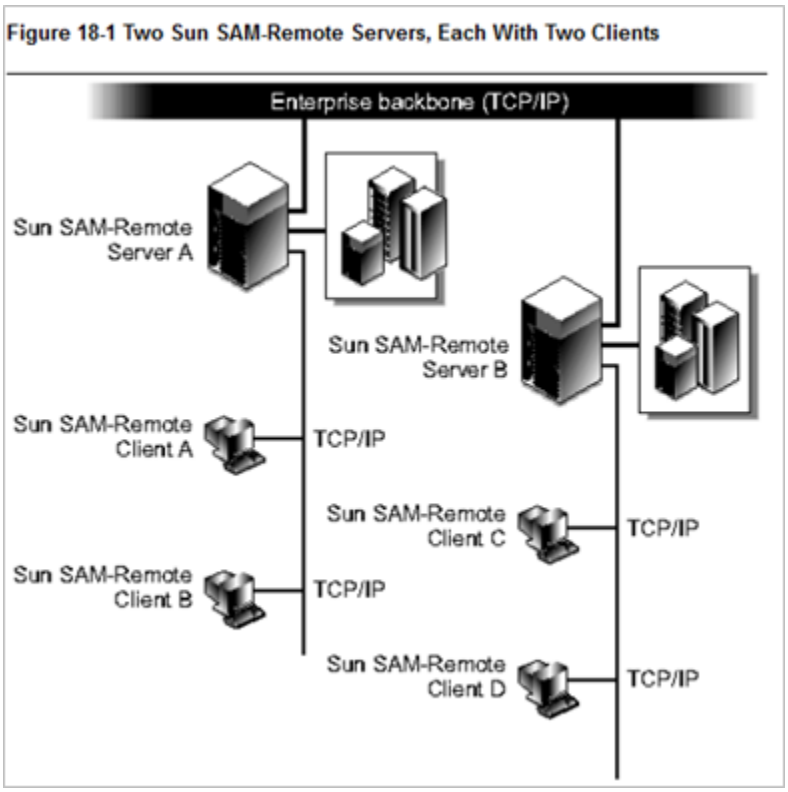
129. As an example, the '132 Accused Products, including Oracle HSM Storage Manager, implement a method for "handling files within a data management system" to back up files based on user defined "policies." (*See* Oracle Hierarchical Storage Manager and StorageTek QFS Software (July 2019), at 21).

130. Oracle HSM's method for handling files "provides a policy set comprising at least one service class rule." For example, every active file in the Oracle HSM's archiving file system belongs to exactly one archive set. (*See* Oracle Hierarchical Storage Manager and StorageTek QFS Software (July 2019), at 22). Each archive set consists of policies that control the archiving process for that archive set. (*Id.*). Further, each archive set has a "service class rule" that defines selection criteria based on the file's attributes. (*See id.*).

131. StorageTek QFS which is integrated in Oracle HMS uses the sam-archiverd daemon to "receive one or more attributes of a file from one of a plurality of clients." For instance, the sam-archiverd daemon communicates between the file evaluation module – the sam-arfind process – and the SAM-Remote clients. The sam-archiverd daemon starts the sam-arfind process to archive files from the SAM-QFS file system, which are the SAM-Remote clients. (*See* Sun QFS, Sun SAM-FS, and Sun SAM QFS, File System Administrator's Guide, at 244-245, 275, available at https://docs.oracle.com/cd/E19314-01/816-2542-10/816-2542-10.pdf). One attribute of a file received by Oracle HMS may be the file's size. (*See* Oracle Hierarchical Storage Manager and StorageTek QFS Software (July 2019), at 22).

132. The sam-archiverd daemon in Oracle HMS is configured to communicate with "clients comprising at least two different computing platform." Oracle HMS with StorageTek QFS allows the Sun SAM-Remote server and clients to be configured to provide multiple archive copies between two or more Oracle Solaris host systems. (*See, e.g.*, Sun Storage Archive Manager 5.3 Configuration and Administration Guide (June 2012), available at https://docs.oracle.com/cd/E22586_01/html/E22572/glatd.html). For example, Sun SAM-Remote clients run Oracle SAM-QFS software, which can be installed on both Linux and Solaris

operating systems. Therefore, SAM-Remote clients can be Linux or Solaris based systems. (*See, e.g.,* About Shared File Systems and the Linux Client, available at https://docs.oracle.com /cd/E22586_01/html/E22570/gledk.html; s*ee also*:



Sun Storage Archive Manager 5.3 Configuration and Administration Guide (June 2012)).

133. StorageTek QFS which is integrated in Oracle HMS uses the sam-arfind process to "apply the service class rule to the file to assign a service class." Sam-arfind identifies the policy set that defines the archiving policies for the file by comparing the file's attributes, such as file size, to the selection criteria defined by each policy set. (*See* Oracle Hierarchical Storage Manager and StorageTek QFS Software (July 2019), at 22). Once it has located the correct policy set and the corresponding parameters, sam-arfind checks the archive age and then assigns a priority, or service class, to the file. (*See id.*).

134. Oracle HSM with StorageTeck QFS uses the sam-arcopy process to "conduct operations on the file in a manner directed by the service class." For instance, after sam-arfind has identified the files that need archiving, prioritized them, and added them to archive requests for each archive set, it returns the requests to the sam-archiverd daemon. And once the archive requests are scheduled, sam-archieverd calls an instance of the sam-arcopy process for each request and drive scheduled. The sam-arcopy instances then copy the data files to archive files on archival media, update the archiving file system's metadata to reflect the existence of the new copies, and update the archive logs. (*See*, *e.g.*, Oracle Hierarchical Storage Manager and StorageTek QFS Software (July 2019), at 23).

135. By making, using, offering for sale, and/or selling products in the United States and/or importing products into the United States, including but not limited to the '132 Accused Products, Oracle has injured Daedalus and is liable to Daedalus for directly infringing one or more claims of the '132 Patent, including without limitation claim 8 pursuant to 35 U.S.C. § 271(a).

136. As a result of Oracle's infringement of the '132 Patent, Daedalus has suffered monetary damages, and seeks recovery, in an amount to be proven at trial, adequate to compensate for Oracle's infringement, but in no event less than a reasonable royalty with interest and costs.

FIFTH COUNT

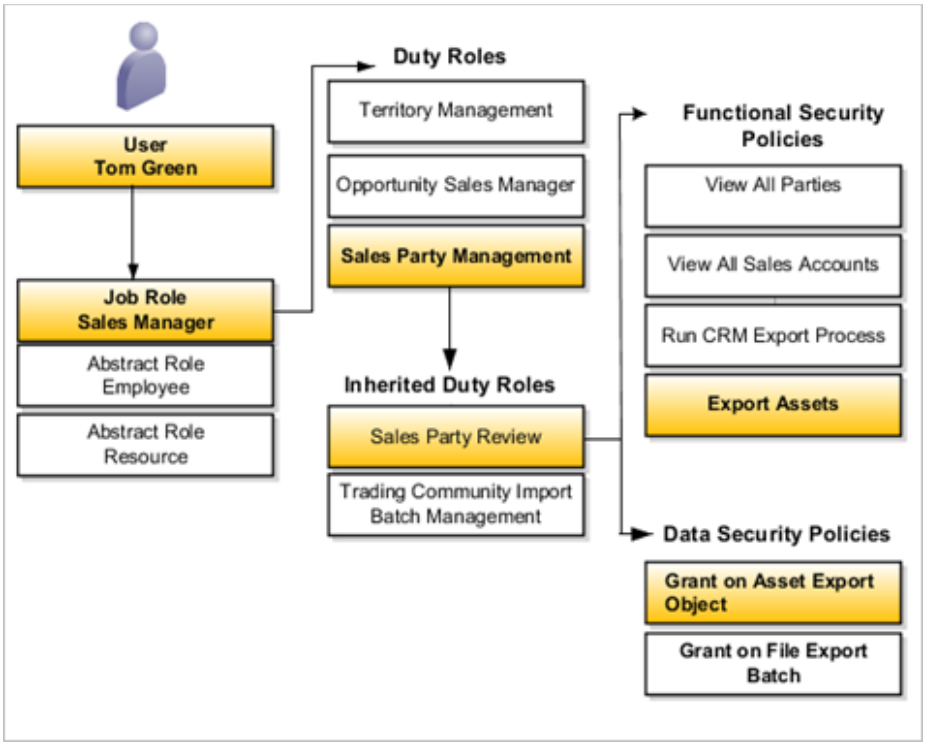
(Infringement of U.S. Patent No. 9,032,076)

137. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-136 of this Complaint as though fully set forth herein.

138. On information and belief, Oracle has directly infringed and continues to directly infringe one or more claims of the ’076 Patent, including at least Claim 6 of the ’076 Patent, in the state of Texas, in this judicial district, and elsewhere in the United States by, among other things, making, using, selling, offering for sale, and/or importing into the United States products that embody one or more of the inventions claimed in the ’076 Patent, including but not limited to the above-identified Fusion Application products, including the exemplary Oracle Sales Cloud which is integrated with Fusion Application products, and all reasonably similar products (“the ’076 Accused Products”), in violation of 35 U.S.C. § 271(a).

139. As an example, the ’076 Accused Products, including Oracle Sales Cloud with Fusion Applications, implement “a role-based access control method” to prevent unauthorized access to resources. (*See* Understanding Role-Based Security; and *see*, available at https://docs.oracle.com/cd/E83857_01/saas/sales/r13-update17d/oscus/authorization-with-role-based-access-control.html).

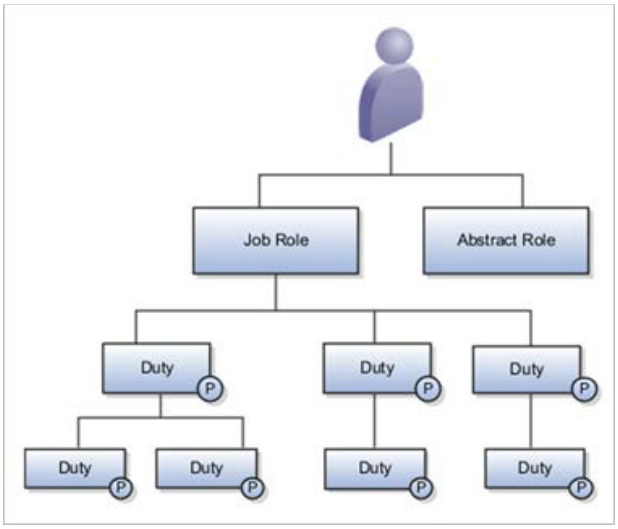
140. In Oracle Fusion Application and Oracle Sales Cloud’s role-based access control method includes “defining roles to be sets of permissions on individual resources, thus forming role instances.” For example, users are assigned roles such as duty roles, and roles are assigned privileges to access protected resources. That is, users gain access to application data and functions when the user is assigned a role. (*See* Authorization with Role-Based Access Control). In Oracle Sales Cloud, when privileges are added to a duty role, thereby binding resources with that duty role, a role instance is formed. (*See:*



Authorization with Role-Based Access Control).

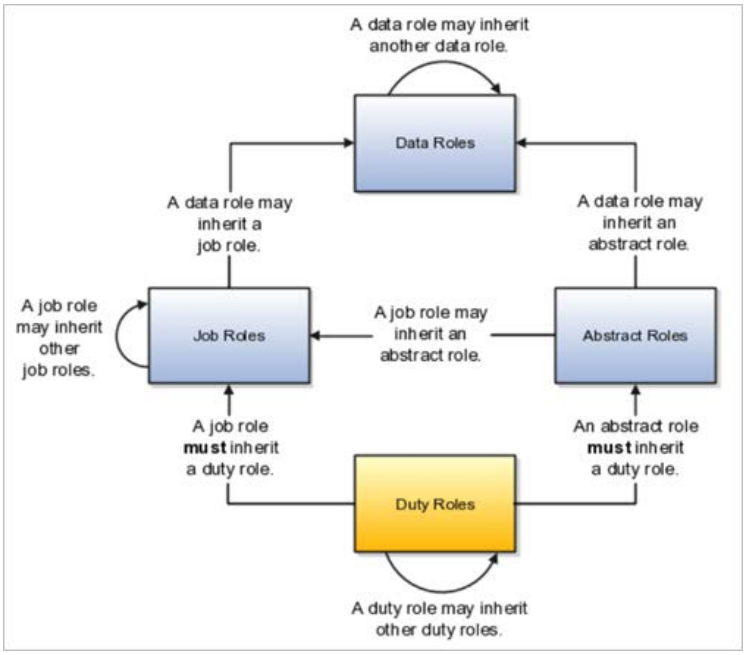
141. Oracle Fusion Application and Oracle Sales Cloud uses role hierarchies to "assign at least one set of role instances to at least one group." Role hierarchies allow duty roles and their associated privileges to be grouped to represent a feature set in Oracle Sales Cloud. (*See, e.g.*, Authorization with Role-Based Access Control*; see also* Creating Job, Abstract, and Duty Roles).

142. Oracle Fusion Application integrated in Oracle Sales Cloud further "assigns at least one group of a set of role instances to at least one super role." In Oracle Sales Cloud, each role can be linked to other roles in a parent-child format to form a hierarchy of roles. (*See* Authorization with Role-Based Access Control. For example, job roles and abstract roles are super roles). Users are assigned job and abstract roles, which inherit the privileges and permissions of duty roles. (*See*:

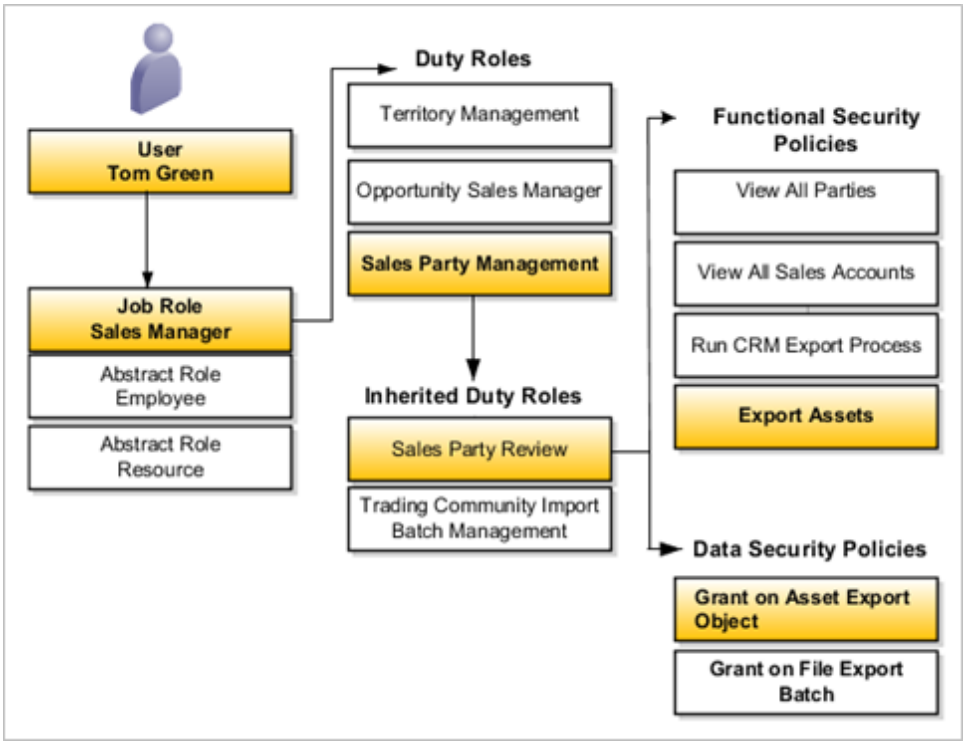


Id.).

143. In Oracle Fusion Application integrated in Oracle Sales Cloud, super roles such as job roles, abstract roles, and some duty roles, are "nested according to properties including a name, a parent role, the set of role instances, and an externalization state." For instance, the abstract and job roles must inherit, or are nested according to, duty roles. (*See* Role-Based Access Control). Additionally, job roles may inherit abstract roles or other job roles. (*Id.; see also*:



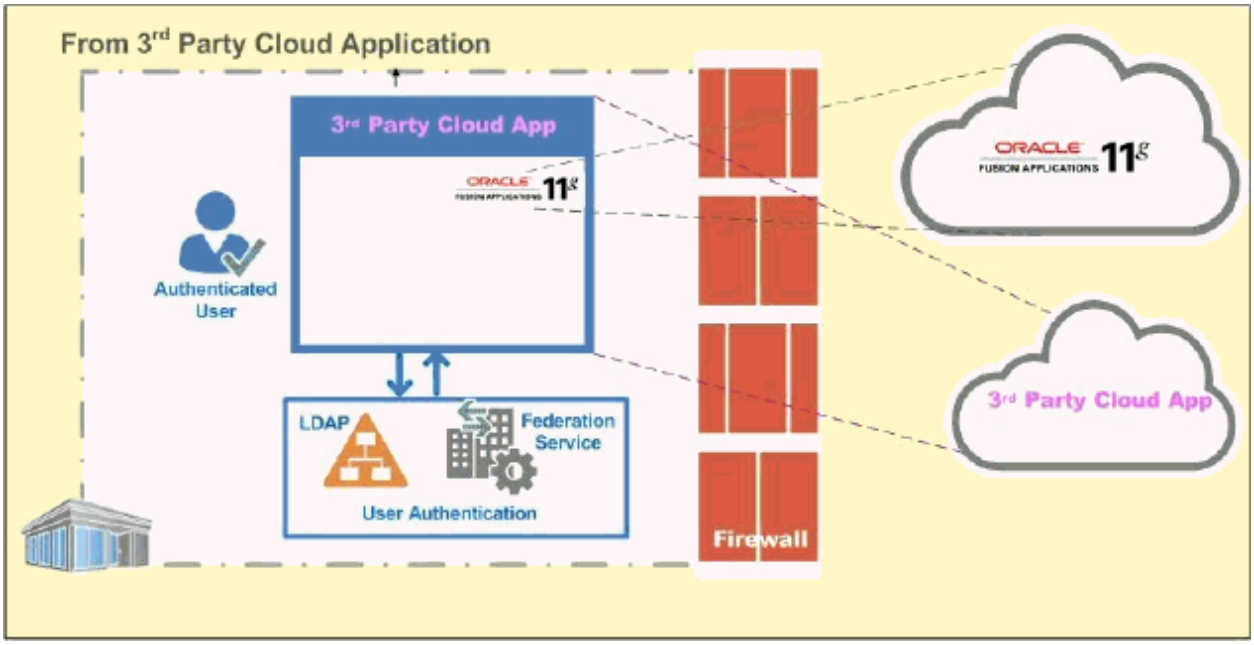
Id.). Also, Oracle Fusion Application nests each super role with a name and set of role instances, such as the "Sales Manager job role," which "inherits duty roles including the Sales Party Management duty role and the Opportunity Sales Manager duty role." (*See* Authorization with Role-Based Access Control; *see also*:



Id.). These sets of role instances are nested to the functional security and data security policies and resources. (*Id.*).

144. Oracle Sales Cloud nests each super role according to "an externalisation state," requiring an external authentication to be performed when the user logs into Oracle Sales Cloud, and only after successful authentication is the user session established, with all the roles assigned to the user loaded into the session repository. (*See, e.g.*, Authorization with Role-Based Access Control. For instance, Oracle Sales Cloud instructs how to create an Authorization Policy and a Security Module definition. (*See, e.g*, Managing Policies and Roles, available at

https://docs.oracle.com/cd/E21764_01/admin.1111/e14096/manage_policies_roles.htm#ESADR479; *See also:*



Fusion Applications Single Sign On – Business User perspective, available at https://blogs.oracle.com/functionalarchitecture/fusion-applications-single-sign-on-business-user-perspective).

145. Oracle Sales Cloud provides for each super role to be "modified by adding or removing the role instances from at least one group." For instance, authorization to access a particular protected resource by a role can be changed. (*See* Determining Authorization Permissions, available at https://docs.oracle.com/cloud/latest/fmw122100/OWSMS/configure-authorization.htm#OWSMS4351).

146. By making, using, offering for sale, and/or selling products in the United States and/or importing products into the United States, including but not limited to the '076 Accused Products, Oracle has injured Daedalus and is liable to Daedalus for directly infringing one or

more claims of the ’076 Patent, including without limitation claim 6 pursuant to 35 U.S.C. § 271(a).

147. On information and belief, Oracle is inducing and/or has induced infringement of one or more claims of the ’076 Patent, including at least claim 6, as a result of, among other activities, instructing, encouraging, and directing its customers on the use of the ’076 Accused Products in an infringing manner in violation of 35 U.S.C. § 271(b). For example, through its website, instructional guides, and manuals, Oracle provides its customers with detailed explanations, instructions, and information on how to use and implement the ’076 Accused Products which demonstrate active steps taken to encourage direct infringement. (*See, e.g.*, Authorization with Role-Based Access Control; *see* Role-Based Access Control; *and see* Creating Job, Abstract, and Duty Roles). On information and belief, Oracle has had knowledge of the ’076 Patent at least as of 2011. Despite this knowledge of the ’076 Patent, Oracle has continued to engage in activities to encourage and assist its customers in the use of the ’076 Accused Products. Thus, on information and belief, Oracle (1) had actual knowledge of the patent; (2) knowingly induced its customers to infringe the patent; and (3) had specific intent to induce the patent infringement.

148. On information and belief, Oracle has known about the ’076 Patent and its contents since about at least January 2011. On information and belief, Oracle and the inventor of U.S. U.S. Patent No. 8,769,604 (“Oracle’s ’604 Patent”) knew of the ’076 Patent and its contents when the application that eventually became the ’076 Patent was cited as a reference by the patent examiner during the prosecution of Oracle’s ’604 Patent. Oracle, having learned of the likelihood of infringement of the ’076 Patent, nevertheless acted in a way that infringed.

149. On information and belief, by using the ’076 Accused Products as encouraged and assisted by Oracle, Oracle’s customers have directly infringed and continue to directly infringe one or more claims of the ’076 Patent, including at least claim 6. On information and belief, Oracle knew or was willfully blind to the fact that that its actions would induce its customers’ direct infringement of the ’076 Patent.

150. Oracle’s infringement of the ’076 Patent has been and continues to be deliberate and willful, and, this is therefore an exceptional case warranting an award of enhanced damages and attorneys’ fees and costs pursuant to 35 U.S.C. §§ 284-285.

151. On information and belief, Oracle will continue to infringe the ’076 Patent unless enjoined by this Court.

152. As a result of Oracle’s infringement of the ’076 Patent, Daedalus has suffered monetary damages, and seeks recovery, in an amount to be proven at trial, adequate to compensate for Oracle’s infringement, but in no event less than a reasonable royalty with interest and costs. Oracle’s infringement of Daedalus’ rights under the ’076 Patent will continue to damage Daedalus, causing irreparable harm for which there is no adequate remedy at law, unless enjoined by this Court.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for judgment and *see*ks relief against Oracle as follows:

a. For judgment that Oracle has infringed and continues to infringe the claims of the ’494, ’886, ’612, ’132, and ’076 Patents;

b. For a permanent injunction against Oracle and its respective officers, directors, agents, servants, affiliates, employees, divisions, branches, subsidiaries, parents,

and all other acting in active concert therewith from infringement of the ’494, ’886, ’612, ’132, and ’076 Patents;

c. For an accounting of all damages sustained by Plaintiff as the result of Oracle’s acts of infringement;

d. For a mandatory future royalty payable on each and every future sale by Oracle of a product that is found to infringe one or more of the Daedalus Patents and on all future products which are not colorably different from products found to infringe;

e. For a judgment and order finding that Oracle’s infringement is willful and awarding to Plaintiff enhanced damages pursuant to 35 U.S.C. § 284;

f. For a judgment and order requiring Oracle to pay Plaintiff’s damages, costs, expenses, and pre- and post-judgment interest for its infringement of the ’494, ’886, ’612, ’132, and ’076 Patents as provided under 35 U.S.C. § 284;

g. For a judgment and order finding that this is an exceptional case within the meaning of 35 U.S.C. § 285 and awarding to Plaintiff its reasonable attorneys’ fees; and

h. For such other and further relief in law and in equity as the Court may deem just and proper.

DEMAND FOR JURY TRIAL

Pursuant to Rule 38(b) of the Federal Rules of Civil Procedure, Plaintiff demands a trial by jury in this action of all issues triable by a jury.

Dated: May 26, 2020

Respectfully Submitted,

/s/ _____

Denise M. De Mory (*Pro Hac Vice Pending)*
Cal. Bar No. 168076
ddemory@bdiplaw.com
Robin Curtis (*Pro Hac Vice Pending)*
Cal. Bar No. 271702
rcurtis@bdiplaw.com
Jennifer L. Gilbert (*Pro Hac Vice Pending)*
Cal. Bar. No. 255820
jgilbert@bdiplaw.com
Corey Johanningmeier (*Pro Hac Vice Pending*)
Cal. Bar. No. 251297
cjohanningmeier@bdiplaw.com
Brenda Entzminger *(Pro Hac Vice Pending)*
Cal Bar No. 226760
bentzminger@bdiplaw.com
Jenna Fuller (*Pro Hac Vice Pending)*
Cal. Bar No. 324658
jfuller@bdiplaw.com
Gail Jefferson (*Pro Hac Vice Pending)*
Cal. Bar No. 325874
gjefferson@bdiplaw.com
Nicolas Mancuso (*Pro Hac Vice Pending)*
Cal. Bar No. 271668
nmancuso@bdiplaw.com
BUNSOW DE MORY LLP
701 El Camino Real
Redwood City, CA 94063
Telephone: (650) 351-7248
Facsimile: (415) 426-4744

Attorney in Charge for Plaintiff, Daedalus Blue, LLC

B. Russell Horton
State Bar No. 10014450
GEORGE BROTHERS KINCAID & HORTON, L.L.P.
114 West 7th Street, Ste. 1100
Austin, Texas 78701
(512) 495-1400 telephone
(512) 499-0094 facsimile
rhorton@gbkh.com

Attorney for Plaintiff, Daedalus Blue, LLC