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

DAEDALUS BLUE, LLC, *Plaintiff*, v. MICROSOFT CORPORATION, *Defendant*.	Case No. 6:20-cv-1152 **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 Defendant Microsoft Corporation ("Microsoft") 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 5 patents asserted in this case are the result of the work from 14 different IBM researchers, spanning a period of nearly a decade.

2. Over the years, the inventions claimed in the Asserted Patents have been licensed to many companies, including Amazon Web Services and Oracle Corporation.

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 Microsoft Corporation is a Washington Corporation with a principal place of business at One Microsoft Way, Redmond, Washington. Microsoft Corporation also maintains corporate sales offices in this District, located at 10900 Stonelake Boulevard, Suite 225, Austin, Texas, and at Concord Park II 401 East Sonterra Boulevard, Suite 300, San Antonio, Texas.

6. Microsoft conducts business in Texas and in the Western District of Texas, as set forth below.

JURISIDICTION AND VENUE

7. 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).

8. Defendant Microsoft is subject to this Court's personal jurisdiction in accordance with due process and/or the Texas Long Arm Statute because, in part, Microsoft "[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.

9. This Court also has personal jurisdiction over Defendant Microsoft because it 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, Microsoft has 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.

10. Defendant Microsoft is subject to the Court's personal jurisdiction, in part, because it regularly conducts and solicits business, or otherwise engages in other persistent courses of conduct in this district, and/or derives substantial revenue from the sale and distribution of infringing goods and services provided to individuals and businesses in this district.

11. This Court has personal jurisdiction over Defendant Microsoft because, *inter alia*, Defendant (1) has substantial, continuous, and systematic contacts with this State and this judicial district; (2) owns, manages, and operates facilities in this State and this judicial district; (3) enjoys substantial income from its operations and sales in this State and this judicial district; (4) employs Texas residents in this State and this judicial district; and (5) solicits business and market products, systems and/or services in this State and judicial district including, without limitation, those related to the infringing accused products.

12. Venue is proper in this District pursuant to at least 28 U.S.C. §1319(b)-(c) and §1400(b), at least because Defendant Microsoft, either directly or through its agents, have committed acts within this judicial district giving rise to this action, and continue to conduct business in this district, and/or has committed acts of patent infringement within this District giving rise to this action.

FACTUAL ALLEGATIONS

Daedalus Patents

13. The IBM inventions contained in the Asserted Patents in this case relate to groundbreaking improvements to cloud infrastructure, cloud management, network security, database management, data processing, and data management and have particular application in the cloud-based computing environments as will be further described below.

U.S. Patent No. 7,177,886

14. 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 1.**

15. 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.

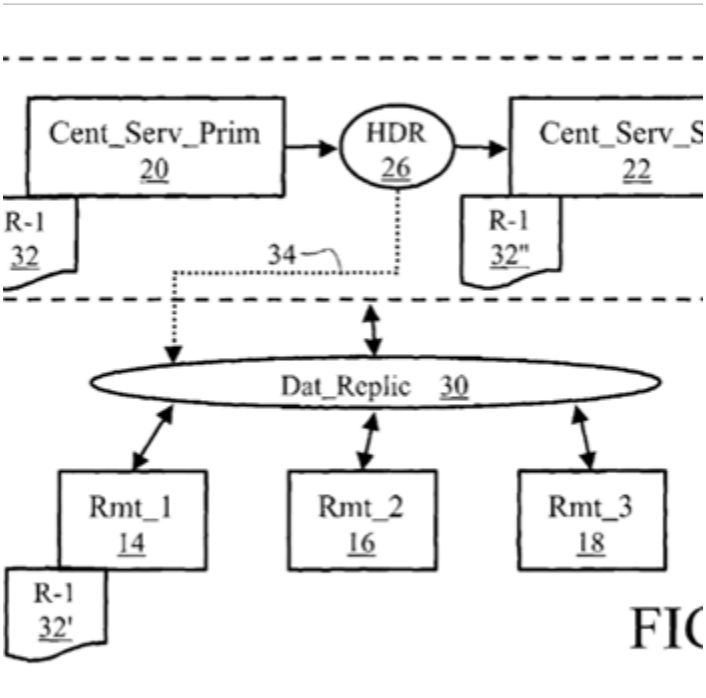
16. The '886 Patent describes, among other things, a novel apparatus configuration that improves data storage techniques that provides for faster, more reliable backup of data files to remote servers, which ensures against data loss and system failure. These inventive technological improvements solved 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. 1 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.

17. 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).

18. 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. 1 at 25-41).

19. 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. 1 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).



FIC

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.

20. 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. 1 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.

U.S. Patent No. 7,437,730

21. On October 14, 2008, the U.S. Patent and Trademark Office duly and lawfully issued United States Patent No. 7,437,730 ("the '730 Patent"), entitled "System and Method for Providing a Scalable On Demand Hosting System." A true and correct copy of the '730 Patent is attached hereto as **Exhibit 2**.

22. Daedalus is the owner and assignee of all right, title, and interest in and to the '730 Patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of it.

23. The '730 Patent describes, among other things, novel systems, methods, and devices that optimize dynamic control over the fractions of workloads handled by virtual machines across multiple servers in a cloud environment. By recognizing when one of the servers is overloaded and automatically shifting work to another, not yet overloaded server, these

inventive technological improvements solved then-existing problems in the field of virtual machine based hosting architectures, including improved server utilization in a virtual machine based hosting architecture. For example, prior to the invention of the '730 Patent, one type of data center hosting was called dedicated hosting, in which servers would be statically assigned to customers/applications based on the peak load that each customer may receive. However, since the peak load is significantly higher than the average load, this would result in lower than average server utilization.

24. To improve on server utilization, dedicated hosting solutions prior to the '730 Patent were modified to dynamically assign servers to each customer. Hosting solutions provided traffic measuring entities to determine the offered load for each customer or application, and based on that offered load, the traffic measuring entity determined the number of servers needed for each customer or application. Though an improvement over static assignment, the efficiency of this type of solution was still severely limited. For example, owing to the time needed to reassign servers, existing solutions used excessive time and resources. Then-existing systems failed to provide fine grain control and scalability in a virtual machine based hosting architecture. For example, existing systems did not dynamically adjust workload distribution among a set of virtual machines. Without such means of dynamic adjustment, existing systems failed to maintain an optimum utilization level across the set of servers. (See Ex. 2, 1:13-1:39).

25. The '730 Patent overcomes these drawbacks and improves server utilization in a virtual machine hosting architecture, for example, by describing novel and inventive systems in which finer grain control is achieved in optimizing workload distribution among multiple servers. The '730 Patent uses resource management logic to optimally distribute server

resources among virtual machines and servers, wherein the virtual machines at each of the set of servers can each serve a different workload depending on available resources. In one aspect, the resource management logic distributes server resources, such as percentage of CPU, percentage of network bandwidth, etc., according to the current and predicted resource needs of each of the multiple workloads handled by the group of virtual machines. Moreover, the system can dynamically adjust the fractions of each of the multiple workloads to provide for optimization of multiple workloads across multiple servers. For example, the system recognizes when one of the set of servers is overloaded and automatically shifts work to another of the set of servers which is not overloaded. In this way, the '730 Patent can achieve finer grain control in optimizing workloads across servers.

26. An exemplary hosting architecture diagramed in Fig. 1 of the '730 Patent is shown below, in which each of the servers (12, 14, 16) host multiple VMs, and there exists an exemplary global resource allocator (26) for allocating resources among the VMs along with resource control agents (44, 46, 48). Customer applications (18, 20, 22, 24) run on multiple VMs across multiple servers. As depicted, a load balancer (50, 52, 54, 56) is attached to each customer, however, the '730 Patent also describes how a single load balancer could be used for multiple customers.

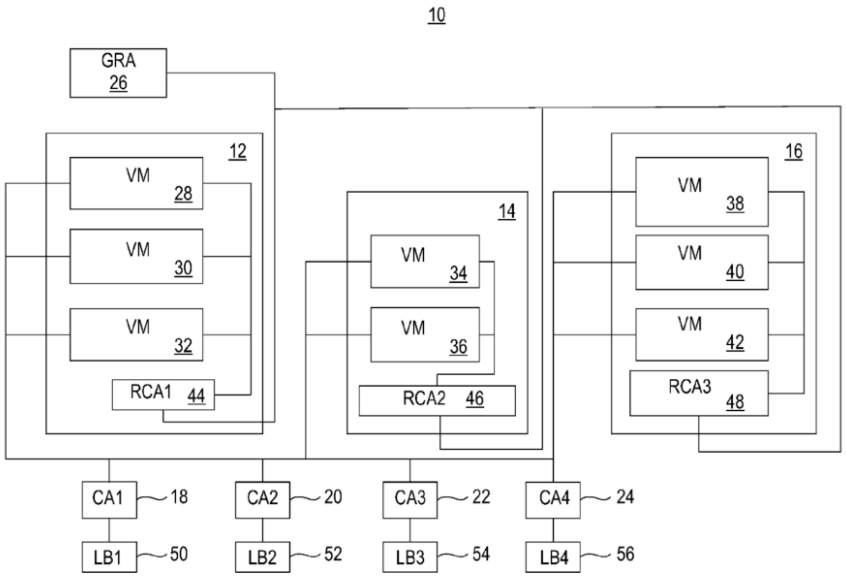


FIG. 1

(Ex. 2 at FIG. 1)

27. The '730 Patent overcomes the efficiency limitations of the prior art to optimize the distribution of workloads across multiple servers. In one embodiment, for example, the '730 Patent describes distributing workloads among virtual machines according to a server optimization device with several resource allocator components at each of the multiple server machines. These resource allocators are responsible for creating virtual machines and assigning virtual machines to workloads in response to instructions received from the global resource allocator partitioning component. (Ex 2 at 1:37-2:19). In one aspect, the '730 Patent can include a global resource allocator to monitor distribution between the set of virtual machines and a load balancer to measure the current offered load, wherein the global resource allocator utilizes the measurements from one or more load balancers to determine how to distribute the resources among the virtual machines. (*Id.* at 2:9-15). Moreover, the global resource allocator partitioning component of the '730 Patent assigns resources at each of the sever machines to the assigned virtual machines according to the identified resource requirements. (*Id.* at 2:25-32). The '730

Patent further enables finer grain control and workload optimization by reassigning the virtual machines according to changes in the identified resource requirements. (*Id.* at 2:20-36). The '730 Patent further optimizes workload across multiple servers by utilizing the global resource allocator partitioning component to issue redistribution instructions to all of the resource allocator components at each of the server machines. (*Id.* at 2:43-52). This is one example by which the '730 Patent provides for optimizing workload across the servers to prevent the over-utilization or under-utilization of any of the server machines. (*Id.* at 2:48-56).

28. The novel features of the invention are recited in the claims. For example, Claim 1 of the '730 Patent recites:

> A system to provide finer grain control in optimizing multiple workloads across multiple servers, comprising:
>
> a plurality of servers to be utilized by multiple workloads;
>
> a plurality of virtual machines at each of the plurality of servers, wherein the plurality of virtual machines at each of the plurality of servers each serve a different one of the multiple workloads; and
>
> resource management logic to distribute server resources to each of the plurality of virtual machines according to current and predicted resource needs of each of the multiple workloads utilizing the server resources,
>
> whereby, each of the multiple workloads are distributed across the plurality of servers, wherein fractions of each of the multiple workloads are handled by the plurality of virtual machines,
>
> whereby, the fractions of each of the multiple workloads handled by each of the virtual machines can by dynamically adjusted to provide for optimization of the server resources utilized by the multiple workloads across the multiple servers.

(Ex. 2 at 8:2-21). Claim 1 of the '730 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 achieve finer grain control in optimizing multiple workloads across multiple servers in a virtual

machine based hosting architecture system; improve upon the inefficiencies among dedicated hosting solutions which dynamically assigned servers to each customer; and avoid the unnecessary use of time and resources required of prior solutions to reassign servers. (Ex. 2 at Abstract; 1:23-33, 37-65; 2:1-8).

U.S. Patent No. 8,381,209

29. On February 19, 2013, the U.S. Patent and Trademark Office duly and lawfully issued United States Patent No. 8,381,209 ("the '209 Patent"), entitled "Moveable Access Control List (ACL) Mechanisms for Hypervisors and Virtual Machines and Virtual Port Firewalls." A true and correct copy of the '209 Patent is attached hereto as **Exhibit 3**.

30. Daedalus is the owner and assignee of all right, title, and interest in and to the '209 Patent, including the right to assert all causes of action arising under said patent and the right to any remedies for infringement of it.

31. The '209 Patent describes, among other things, a novel system and method that improves the control of network security of a virtual machine (VM) during the migration of the VM to a new underlying hardware device by enforcing network security and routing at a hypervisor layer when migrating the virtual machine. A hypervisor (sometimes called a virtualization manager) is a program that allows multiple VMs to share hardware resources. Each operating system running on a VM appears to have the processor, memory, and other resources all to itself. However, the hypervisor actually controls the real processor and its resources, allocating what is needed to each operating system in turn. In order to perform maintenance on or provide a fail-over for a processor device or machine, it is desirable to move or migrate a virtual machine (VM) from one processor machine or device to another processor machine or device. (Ex. 3 at 2:27-31).

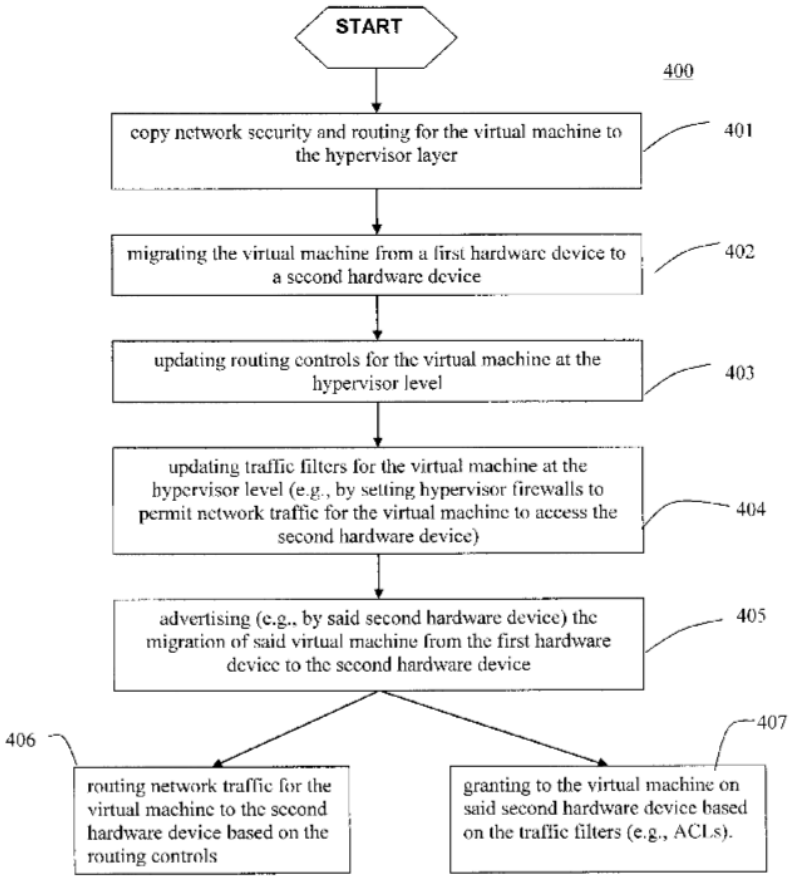
32. The ’209 Patent describes inventive technological improvements that solved then-existing problems in the field relating to VM migration. As described in the ’209 Patent, in the conventional methods and systems, it is difficult to move one virtual machine from one physical machine to another. Generally, in conventional systems, to move a virtual machine from one machine to another (e.g., from hardware 1 to hardware 2), the conventional methods and systems would merely shut down and copy from hardware 1 to hardware 2. The conventional systems and methods have difficulties with security and routing. (*Id.* at 5:30-37). For example, some conventional systems, before the date of the ’209 invention, did not have access control lists (ACLs) and provided very little security. (*Id.* at 2:24-59). In other conventional systems, an ACL would be installed on a real network switch (hardware) in order to restrict the access to the device. To migrate a virtual machine from one device to another device, a complex update scheme was required to update the ACLs in the real switches and the filters in the firewalls. (*Id.* at 3:6-9). Additionally, routing generally was provided by a mechanism known as an open shortest path first (OSPF) route.

33. To solve the problems with the conventional systems and methods, the ’209 Patent invention copies security and routing, etc. for the virtual machine to the hypervisor layer so that the user will see no difference in operation between running the virtual machine on hardware 1 or hardware 2. That is, according to the present invention, the first and second device (e.g., hardware 1 and hardware 2) would each act the same (and preferably, would each have the same internet protocol (IP) address). An important problem arises when networks are very large, such as Google and Yahoo, in which there could be a thousand servers, and no flat topography, switches and routers to protect the servers. That is, in such systems, the virtual

system is run on top of the hypervisor such that each virtual system is only as good as the security at each machine. (*Id.* at 5:38-53).

34. The ’209 Patent overcomes these drawbacks and improves the functioning of a computer network by enforcing network security and routing at a hypervisor during migration. To migrate the virtual machine from a first hardware device to a second hardware device, the ’209 invention routes network traffic for the virtual machine to the second hardware device at the hypervisor layer. The ’209 invention also may use firewalls to permit network traffic for the virtual machine to go to the second hardware device at the hypervisor layer. The hypervisor level provides traffic filtering and routing updating. Thus, the real switches do not need to be updated at the first and second hardware devices. (Ex. 3 at 5:38-62). The invention decentralizes the updating scheme by using the hypervisor layer for security and routing, thus preferably only two software components would be needed to be updated, whereas the conventional systems and methods would require all systems to be updated (e.g., routers, firewalls, etc.). The ’209 invention also is more predictable than the conventional systems and methods. Thus, the ’209 invention has an important advantage over the conventional systems of pushing all security and intelligence to the hypervisor level, instead of the OS level. That way, under the protection of the hypervisor, the ’209 invention can provide traffic filtering and routing

updating. (*Id.* at 6:3-15). An exemplary method is depicted in Figure 4 of the patent as follows:



35. The novel features of the invention are recited in the claims. For example, Claim 1 of the '209 Patent recites:

> A computer implemented method of controlling network security of a virtual machine, the method comprising enforcing network security and routing at a hypervisor layer via dynamic updating of outing controls initiated by a migration of said virtual machine from a first device to a second device.

(Ex. 3 at 15:39-43). Claim 1 of the '209 Patent describes claim elements, individually or as an ordered combination, that were non routine and unconventional at the time of the invention in 2007 and an improvement over prior art, as it provided a way (not previously available) to control network security during VM migration. For example, during VM migration, it was unconventional to enforce network security and routing at a hypervisor layer via dynamic

updating of routing controls initiated by the migration. The invention thus can provide a hypervisor security architecture designed and developed to provide a secure foundation for server platforms, providing numerous beneficial functions, such as, strong isolation, mediated sharing, and communication between virtual machines. These properties can all be strictly controlled by a flexible access control enforcement engine which can also enforce mandatory policies.

U.S. Patent No. 8,572,612

36. 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 4.**

37. 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.

38. 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. 4 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).

39. 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.

40. 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. 4 at 5:17-36).

41. 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 id.* 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).

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

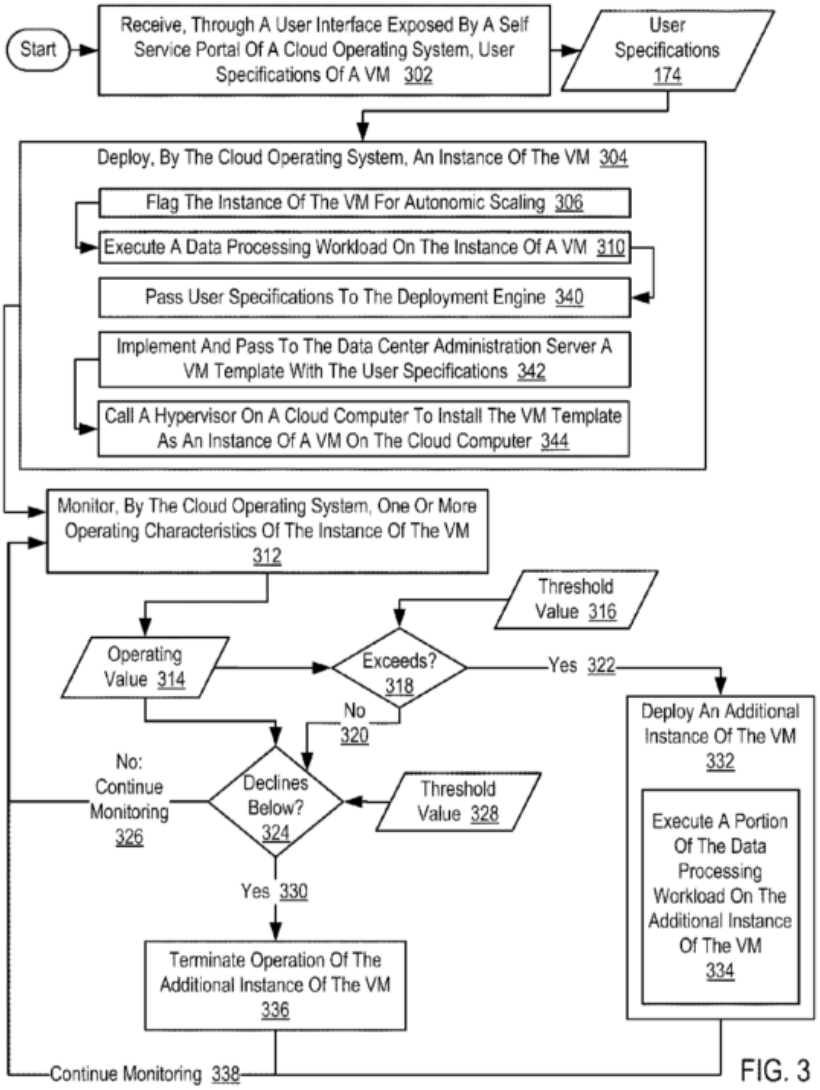


FIG. 3

43. 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. 4 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-specified 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.

U.S. Patent No. 8,671,132

44. 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 5**.

45. 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.

46. 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. 5 1:24-2:3).

47. 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. 5 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.

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

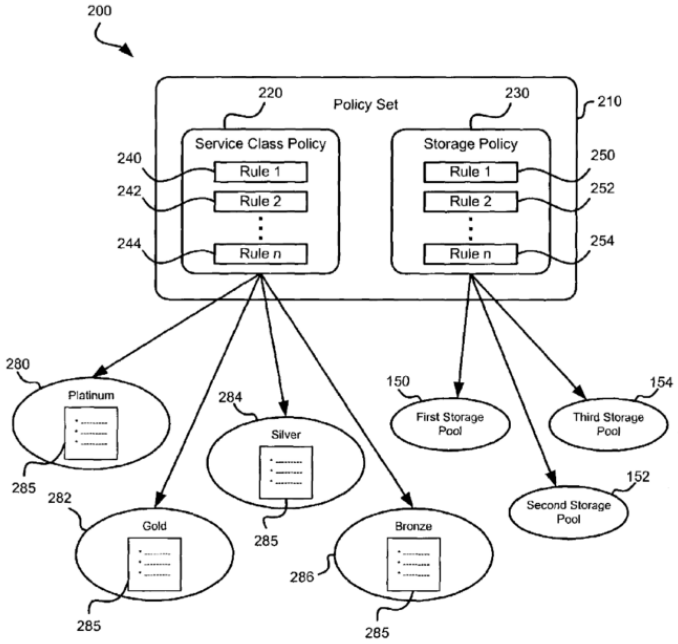


Fig. 2

49. 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. 5 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).

50. 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. 5 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.

51. The '886, '730, '209, '612, and '132 Patents are referred to hereinafter as "the Asserted Patents."

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

Microsoft's Use of the Patented Technology

53. Microsoft Corporation is a multinational computer technology company founded in 1975. Microsoft's venture into Cloud Computing started in or around 2006, and it launched the first version of its Cloud Computing platform, Windows Azure, in 2009, later renamed to Microsoft Azure. Microsoft's Cloud Computing business continues to grow. According to Microsoft's 2019 Annual report, its "commercial cloud business is the largest in the world, surpassing $38 billion in revenue for the year, with gross margin expanding to 63 percent." (*see* Microsoft 2019 Annual Report, *https://www.microsoft.com/invetor/reports/ar19/index.html*). Microsoft identifies IBM as one of its competitors for its Intelligent Cloud business segment.

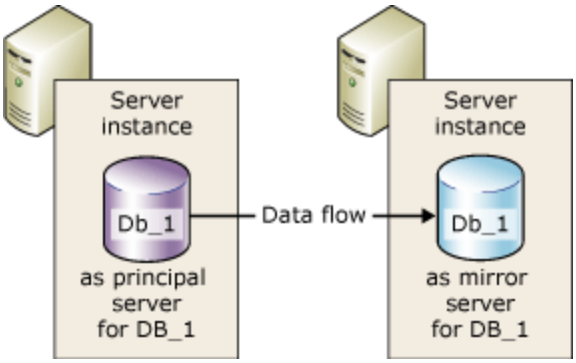
54. On information and belief, Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States various methods and/or products relating to cloud infrastructure, cloud management, network security, database management, data processing, and data management,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

55. Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Microsoft SQL Server.

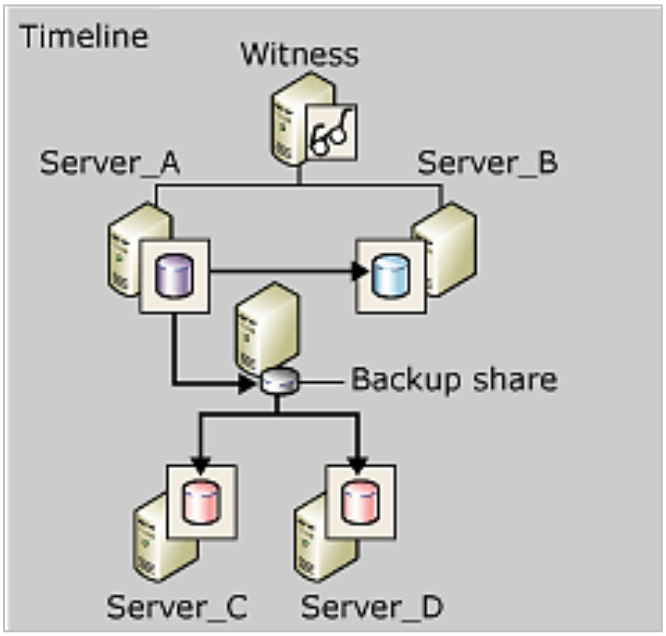
56. Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Database Mirroring feature. Database Mirroring feature is a solution for increasing the availability of a Microsoft SQL Server database.

57. Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Log Shipping feature. Log Shipping feature is a solution used within Microsoft SQL Server that automatically sends transaction log backups from a primary database on a primary server instance to one or more secondary databases on separate secondary server instances.

58. Microsoft SQL Server with Database Mirroring and Log Shipping provide a critical database server that includes a principal server (primary server) and a mirror server (secondary server), with Database Mirroring helping to backup transaction logs on the mirror server database and Log Shipping helping to replicate data among multiple remote servers. (*See, e.g.,* Database Mirroring (SQL Server), available at https://docs.microsoft.com/en-us/sql/database-engine/database-mirroring/database-mirroring-sql-server?view=sql-server-ver15; *see also* Database Mirroring and Log Shipping (SQL Server), available at https://docs.microsoft.com/en-us/sql/database-engine/database-mirroring/database-mirroring-and-log-shipping-sql-server?view=sql-server-ver15; *see also:*



(Database Mirroring (SQL Server)); *see also:*



(Database Mirroring and Log Shipping (SQL Server)).

59. Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States Microsoft Azure.

60. The Azure platform and infrastructure consists of a number of servers contained within datacenters that Microsoft manages. These physical servers host numerous virtual machine instances. (*See* Azure information system components and boundaries, available at https://docs.microsoft.com/en-us/azure/security/fundamentals/infrastructure-components).

61. In Azure, workloads are typically spread across different virtual machines to gain high throughput, performance, and to create redundancy in case a VM is impacted due to an update or other event. (*See* Availability options for virtual machines in Azure, available at https://docs.microsoft.com/en-us/azure/virtual-machines/windows/availability). For example, Availability Sets ensure that the VMs deployed on Azure are distributed across multiple isolated hardware clusters. Doing this ensures that if a hardware or software failure within Azure happens, only a subset of VMs is impacted and that the overall solution remains available and operational. Availability Sets are an essential capability when you want to build reliable cloud solutions. (*See* Tutorial: Create and deploy highly available virtual machines with Azure PowerShell, available at https://docs.microsoft.com/en-us/azure/virtual-machines/windows/tutorial-availability-sets).

62. The Microsoft Azure Resource Manager is the deployment and management service for Azure and works as a software management software that distributes/deploys server resources to the virtual machines. (*See* What is Azure Resource Manager?, available at https://docs.microsoft.com/en-us/azure/azure-resource-manager/management/overview).

63. A feature of Microsoft Azure that Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States is Azure Virtual Machines, an Infrastructure-as-a-Service (IaaS) ("Microsoft Azure IaaS").

64. Microsoft Azure IaaS is an instant computing infrastructure, provisioned and managed over the internet that quickly scales up and down with demand. (S*ee* What is IaaS?, available at https://azure.microsoft.com/en-in/overview/what-is-iaas/).

65. Within Microsoft Azure IaaS, the vertical scale up/down feature causes resources to be redistributed to individual VMs in order to scale up/down. For example, vertical scaling

involves increasing the size of individual VMs to handle increased demand, and when demand decreases, the VM size is decreased. Further, scaling can be set to happen dynamically. (*See* https://azure.microsoft.com/mediahandler/files/resourcefiles/azure-virtual-datacenter-lift-and-shift-guide/Azure_Virtual_Datacenter_Lift_and_Shift_Guide.pdf).

66. Another feature of Microsoft Azure that Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States is Azure Virtual Machines Scaling Set (VMSS) which lets users create and manage a group of load balanced VMs. (*See* Virtual Machine Scale Sets documentation, available at https://docs.microsoft.com /en-us/azure/virtual-machine-scale-sets/).

67. Azure VMSS provides automated virtual machine scaling which allows users to deploy and maintain virtual machines in Azure cloud. (*See* Azure Virtual Machine Scale Sets, available at https://azure.microsoft.com/en-in/services/virtual-machine-scale-sets/). VMSS provides for horizontal scaling, in which users can increase or decrease the number of virtual machines automatically in response to demand, based on customizable metrics, or a defined schedule. (*Id.*). (*See* What are virtual machine scale sets?, available at https://docs.microsoft. com/en-us/azure/virtual-machine-scale-sets/overview). Additionally, VMSS provides for vertical scaling. (*See* Vertical autoscale with virtual machine scale sets, available at https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-vertical-scale-reprovision)

68. Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Azure Site Recovery service, a disaster recovery as a service (DRaaS) which enables the recovery of Azure virtual machines by failing over the machines to a new hardware host. Azure Site Recovery allows users to deploy replication, failover and

recovery processes through Site Recovery to help keep applications running during planned and unplanned outages. (*See* Azure Site Recovery, available at https://azure.microsoft.com/en-us/services/site-recovery).

69. Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States products that utilize and/or work with Network Security Groups (NSGs). NSGs are a set of network security rules that control inbound and outbound traffic to a clients' VMs. For example, Network Security Groups can be used with Azure Site Recovery, and recovery plans allow users to automatically associate NSGs to failed over VMs. (*See* Azure Traffic Manager with Azure Site Recovery, available at https://docs.microsoft.com/en-us/azure/site-recovery/concepts-traffic-manager-with-site-recovery).

70. Microsoft encourages users to utilize NSGs and associate NSGs to failed over VMs automatically during failover using automation scripts. (*See* Network Security Groups with Azure Site Recovery, available at https://docs.microsoft.com/en-us/azure/site-recovery/concepts-network-security-group-with-site-recovery)

71. When a failover occurs using the Azure Site Recovery service, a new VM is created. In a newly created VM a hypervisor firewall is utilized which, by default, blocks all traffic, then rules and exceptions are dynamically modified, including by defining ACLs based on a tenant's service model.

72. Microsoft makes, uses, sells, and/or offers to sell in the United States, and/or imports into the United States the Azure Blob Storage.

73. Microsoft's Azure Blog Storage is a cloud based data storage solution that transitions files based on user defined policies. (*See* Manage the Azure Blob storage lifecycle, available at https://docs.microsoft.com/en-us/azure/storage/blobs/storage-lifecycle-management-

concepts?tabs=azure-portal). Azure Blob Storage lifecycle management offers a rich, rule-based policy that is used to transition data to the appropriate access tiers. (*Id.*).

FIRST COUNT

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

74. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-73 of this Complaint as though fully set forth herein.

75. On information and belief, Microsoft 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 SQL Server, and all reasonably similar products ("the '886 Accused Products"), in violation of 35 U.S.C. § 271(a).

76. As an example, the '886 Accused Products, such as Microsoft SQL Server, comprise databases with a collection of structured data. (*See*, Databases, available at https://docs.microsoft.com/en-us/sql/relational-databases/databases/databases?view=sql-server-ver15). Microsoft SQL Server utilizes the Database Mirroring feature to help backup transaction logs on a mirror database server and the Log Shipping feature to help replicate data among remote servers. (*See, e.g.,* Database Mirroring (SQL Server); Database Mirroring and Log Shipping (SQL Server)). "Log shipping and database mirroring can work together to provide solutions for high availability and disaster recovery." (Microsoft SQL Server 2005, Database Mirroring and Log Shipping Working Together SQL Server, Best Practices Article (January

2008), at 1, available at http://download.microsoft.com/download/d/9/4/d948f981-926e-40fa-a026-5bfcf076d9b9/DBMandLogShipping.docx).

77. Microsoft SQL Server is a "critical database server" that includes a "primary server" and a "secondary server." (*See*, Database Mirroring SQL Server)). The principal server (primary server) is connected with a "primary database instance" and the mirror server (secondary sever) is connected with a "secondary database instance." (*Id.*). In the Database Mirroring configuration, the mirror database (secondary database instance) backs up and and/or "mirrors" the transaction logs of the primary database instance. (*Id.*). "In database mirroring, a read-write database whose transaction log records are applied to a read-only copy of the database (a mirror database)." (*Id.*).

78. The mirror database, which is a secondary server, mirrors "a selected critical database transaction" at the primary database instance. For instance, under the default synchronous operation, "[a]fter a mirroring session starts or resumes, the process by which log records of the principal database that have accumulated on the principal server are sent to the mirror server, which writes these log records to disk as quickly as possible to catch up with the principal server." (*Id.; see also* ALTER DATABASE (Transact-SQL) Database Mirroring, available at https://docs.microsoft.com/en-us/sql/t-sql/statements/alter-database-transact-sql-database-mirroring?view=sql-server-ver15). Once that is done, the mirror 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. "The mirror server hardens the log to disk and returns an acknowledgement to the principal server." (Database Mirroring Operating Modes, available at https://docs.microsoft.com/en-us/sql/database-engine/database-mirroring/database-mirroring-operating-modes?view=sql-server-ver15).

79. Microsoft SQL Server is "a critical database server" with Database Mirroring, which is a "mirroring component" solution that can communicate with the primary server and the secondary server(s) for transferring log records. (*See* Database Mirroring (SQL Server)). Database Mirroring "transfers database log files entries of the primary database instance" to the mirror server. (*Id.*). "After a mirroring session starts or resumes, the process by which log records of the principal database that have accumulated on the principal server are sent to the mirror server, which writes these log records to disk as quickly as possible to catch up with the principal server." (*Id.*). The transaction log records are "applied" and "logged" to the mirror database. "Database mirroring involves redoing every insert, update, and delete operation that occurs on the principal database onto the mirror database as quickly as possible. Redoing is accomplished by sending a stream of active transaction log records to the mirror server which applies log records to the mirror database in sequence, as quickly as possible." (Database Mirroring (SQL Server)).

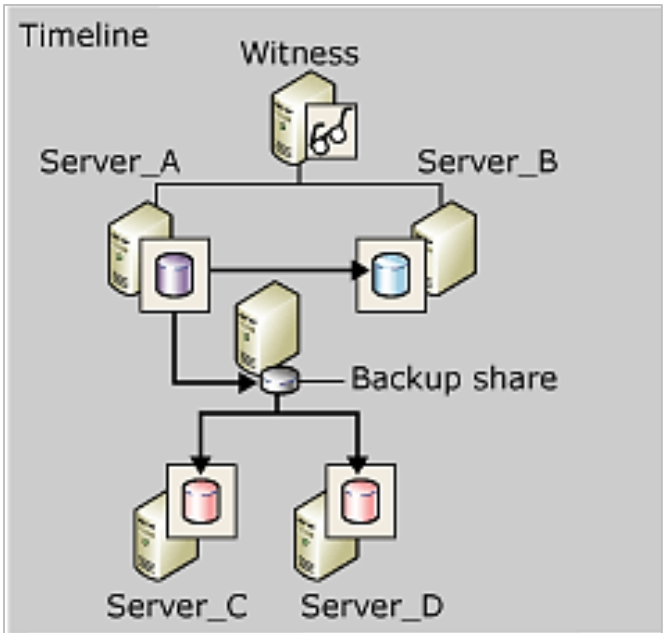
80. After the transaction is performed at the principal database, the database log file entries of the selected critical database transaction are transferred, applied, and logged to the secondary database instance. (*Id.*). After the database file log entries have been written to the mirror database, the mirror server sends an "acknowledgement signal" to the principal server. (*See* Database Mirroring Operating Modes).

81. In Microsoft SQL Server, Database Mirroring, a mirroring component, includes a "control structure." For example, the database mirroring sessions are monitored by Database Mirroring Monitor. (*See, e.g.,* Database Mirroring Monitor Overview, available at https://docs.microsoft.com/en-us/sql/database-engine/database-mirroring/database-mirroring-monitor-overview?view=sql-server-ver15; Monitoring Database Mirroring (SQL Server),

available at https://docs.microsoft.com/en-us/sql/database-engine/database-mirroring/monitoring-database-mirroring-sql-server?view=sql-server-ver15). The Database Mirroring Monitor repeatedly updates the Database Mirroring Status Table, the control structure, by calling sp_dbmmonitorupdate. (*See id.*). "The first time sp_dbmmonitorupdate runs, it creates the database mirroring status table and the dbm_monitor fixed database role in the msdb database. sp_dbmmonitorupdate usually updates the mirroring status by inserting a new row into the status table for every mirrored database on the server instance. (*Id.*). Thus, the Database Mirroring Status Table "indexes critical database transactions that are applied and logged at the secondary database instance" by maintaining and updating the status of the logs of the mirror server. (*See* Monitoring Database Mirroring (SQL Server)).

82. In the Database Mirroring configuration, the acknowledgement that is returned to the principal server is an "acknowledgement signal corresponding to indexing in the control structure." (*See* Database Mirroring Operating Modes). "The mirror server hardens the log to disk and returns an acknowledgement to the principal server." (*Id.*). After the principal server receives the acknowledgement from the mirror server, it confirms to the client that a transaction has committed. (*See* Database Mirroring Operating Modes).

83. In Microsoft SQL Server, the Log Shipping feature is used to ship data to "a plurality of other servers." (*See* Database Mirroring and Log Shipping (SQL Server)). Each of the additional servers "support corresponding database instances." (*See also:*



1. Initially, Server_A is both the principal server for mirroring and the primary server for log shipping. Server_B is the mirror server and is also configured as an alternate primary server, which is currently inactive.

   Server_C and Server_D are log shipping secondary servers that copy shipped log backups from a backup share on a remote host computer.

(*Id.*).

84. In Log Shipping, Backup Jobs is a "data replicator" that "communicates" between the critical database server and the plurality of other servers. (*See* Database Mirroring and Log Shipping (SQL Server)). For example, "[d]uring a log shipping session, backup jobs on the primary database create log backups in a backup folder. From there, the backups are copied by the copy jobs of the secondary servers." (*Id.*).

85. In Log Shipping, Backup Jobs "replicate the selected critical database transaction on at least one of said plurality of other servers" by sending transaction log backups from a

primary database on a primary server instance to one or more secondary databases on separate secondary server instances. (*See* About Log Shipping SQL Server, available at https://docs.microsoft.com/en-us/sql/database-engine/log-shipping/about-log-shipping-sql-server?view=sql-server-ver15). Additionally, the replication is done "responsive to the acknowledgement signal." For example, an acknowledgement is sent to the principal server in Database Mirroring. (*See* Database Mirroring Operating Modes). Typically, when combining Data Mirroring and Log Shipping, "the mirroring session is established before log shipping." (Database Mirroring and Log Shipping (SQL Server)).

86. 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, Microsoft 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).

87. On information and belief, Microsoft 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, amongst 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, Microsoft 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.*, Database Mirroring and Log Shipping (SQL Server); Database Mirroring Operating Modes; Database Mirroring (SQL Server); Database Mirroring Monitor Overview; About Log Shipping (SQL Server)). On information and belief, Microsoft has had knowledge of the '886 Patent at least as of 2009.

Despite this knowledge of the '886 Patent, Microsoft 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, Microsoft (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.

88. On information and belief, Microsoft has known about the '886 Patent and its contents since at least about July 2009. On information and belief, Microsoft and the inventors of U.S. Patent No. 8,069,141 ("Microsoft's '141 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 Notice of References Cited by the patent examiner during the prosecution of Microsoft's '394 Patent. Microsoft, having learned of the likelihood of infringement of the '886 Patent, nevertheless acted in a way that infringed.

89. On information and belief, by using the '886 Accused Products as encouraged and assisted by Microsoft, Microsoft'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, Microsoft knew or was willfully blind to the fact that that its actions would induce its customers' direct infringement of the '886 Patent

90. Microsoft'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

91. On information and belief, Microsoft will continue to infringe the '886 Patent unless enjoined by this Court.

92. As a result of Microsoft'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 Microsoft's infringement, but in no event less than a reasonable royalty with interest and costs. Microsoft'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.

SECOND COUNT

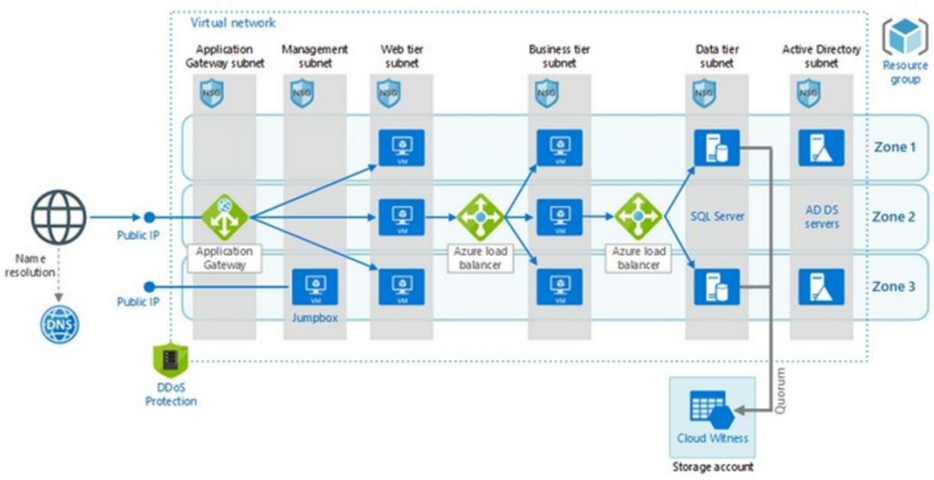
(Infringement of U.S. Patent No. 7,437,730)

93. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-92 of this Complaint as though fully set forth herein.

94. On information and belief, Microsoft has directly infringed and continues to directly infringe one or more claims of the '730 Patent, including at least claim 1 of the '730 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 '730 Patent, including but not limited to the above-identified Microsoft Azure IaaS (Infrastructure-as-a-Service), and all reasonably similar products ("the '730 Accused Products"), in violation of 35 U.S.C. § 271(a).

95. As an example, the '730 Accused Products, comprise "a system to provide finer grain control in optimizing multiple workloads across multiple servers." The Azure platform includes Availability Zones which are physically separate locations within an Azure region, with each Availability Zone made up of one or more datacenters. (*See* Azure geographies, available at https://azure.microsoft.com/en-us/global-infrastructure/geographies/). The Azure datacenters are partitioned into clusters of servers, with virtual machines running across multiple servers, and multiple workloads are then distributed across the servers of Available Zones. (*See* Troubleshoot allocation failures when you create, restart, or resize VMs in Azure, available at

https://docs.microsoft.com/en-us/azure/virtual-machines/troubleshooting/allocation-failure; Tutorial: Create and deploy highly available virtual machines with Azure PowerShell; Windows N-Tier application on Azure with SQL Server, available at https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/n-tier/n-tier-sql-server). With multiple workloads distributed among multiple servers, Microsoft Azure IaaS is a virtual machine hosting architecture "system that provide[s] finer grain control in optimizing multiple workloads". (*See, e.g.*, Windows N-tier application on Azure with SQL Server). The below image depicts an example of how Azure IaaS deploy multiple workloads across multiple servers in an N-Tier application:



Id.).

96. The '730 Accused Products contain a "plurality of servers to be utilized by multiple workloads." The Azure platform's multiple Availability Zones contain datacenters, which house multiple servers that are utilized by multiple workloads. (*See, e.g.,* Windows N-tier application on Azure with SQL Server (showing for example, multiple workloads on Web tier subnet and Business tier subnet); Azure geographies; *and see* How does Microsoft Azure work?, available at https://www.youtube.com/watch?v=KXkBZCe699A&app=desktop).

97. The Availability Zones of Microsoft's Azure IaaS platform include "a plurality of virtual machines at each of the plurality of servers." For example, each Azure Availability Zone consists of multiple servers, and at each of the multiple servers, there exists multiple virtual machines. (*See, e.g.*, Windows N-tier application on Azure with SQL Server). Additionally, the "virtual machines at each of the plurality of servers each serve a different one of the multiple workloads." The virtual machines within the multiple servers across each Azure Availability Zone serve a different workload. By way of example, in the N-Tier Applications, workloads such as Web tier subnet and Business tier subnet are served by the servers. (*See, e.g.*, Windows N-tier application on Azure with SQL Server).

98. The '730 Accused Products include a "resource management logic to distribute server resources to each of the plurality of virtual machines." For instance, Microsoft Azure Resource Manager, a deployment and management service for Azure, is a "resource management logic." Microsoft Azure Resource Manager provides a management layer that enables the user to create, update, and delete resources in their Azure subscription and distributes Azure server resources to the virtual machines by deploying, managing, and monitoring all the resources for the user's solution as a resource group, rather than handling these resources individually. (*See* What is Azure Resource Manager?). Microsoft's Azure Resource Manager further "distribute[s] server resources to each of the plurality of virtual machines" using the Azure Resource Manager template, which defines the resources to be deployed to a resource group. (*See id.*) Microsoft Azure Resource Manager templates deploy virtual machine scale sets, which includes vertical auto scaling. (*See* Azure Virtual Machine Scale Sets, available at https://azure.microsoft.com/

en-in/services/virtual-machine-scale-sets/; Microsoft Azure Virtual Datacenter: Lift and Shift Guide, available at https://azure.microsoft.com/mediahandler/files/resourcefiles/azure-virtual-datacenter-lift-and-shift-guide/Azure_Virtual_Datacenter_Lift_and_Shift_Guide.pdf).

99. Microsoft's Azure Resource Manager template defines resources to deploy, and can be used with Virtual Machine Scale Sets with vertical scaling to "distribute server resources to each of the plurality of virtual machines according to current and predicted resource needs of each of the multiple workloads utilizing the server resources[.]" (*See* Microsoft Azure Virtual Datacenter: Lift and Shift Guide). For example, the Azure Resource Manager distributes server resources to each of the virtual machines according to "current" "resource needs of each of the multiple workloads utilizing the server resources," as virtual scaling involves increasing the size of individual virtual machines to handle increased demands. (*See id*.) Furthermore, the Azure Resource Manager distributes server resources to each of the plurality of virtual machines according to "predicted resource needs of each of the multiple workloads utilizing the server resources," enabling scaling to happen dynamically, based on resource usage. (*See id.)* That is, if workload increases are predictable (for example, based on a specific day each month), vertical scaling can be scheduled to increase the virtual machine size during that period of time. (*See id.)*

100. In the '730 Accused Products' systems, "each of the multiple workloads are distributed across the plurality of servers." By way of example, in the N- Tier Applications of Azure IaaS, multiple workloads of Web Tier subnet and Business Tier subnet are distributed across the multiple Availability Zones. (*See* Windows N-tier application on Azure with SQL Server). Each Microsoft Availability Zone consists of multiple servers. (*See* Azure geographies; How does Microsoft Azure work?). Additionally, in the N-Tier Applications exemplar, "fractions of each of the multiple workloads are handled by the plurality of virtual machines."

Indeed, each of the plurality of servers across a plurality of Availability Zones contain multiple virtual machines, with each virtual machine handling fractions of workloads such as Web Tier subnet and Business Tier subnet. (*See* Windows N-tier application on Azure with SQL Server). These fractions of each of the multiple workloads handled by each of the virtual machines in Microsoft Azure IaaS "can be dynamically adjusted to provide for optimization of the server resources utilized by the multiple workloads across the multiple servers." For example, through the vertical scale up/down feature. (*See id*.; *see also* Vertical autoscale with virtual machine scale sets).

101. With the vertical scale up/down feature in the Microsoft Azure IaaS, individual virtual machines can be scaled up or down dynamically based on the current and predictable workload demands "to provide for optimization of the server resources utilized by the multiple workloads across the multiple servers." (*See* Windows N-tier application on Azure with SQL Server; Microsoft Azure Virtual Datacenter: Lift and Shift Guide). That is, when an individual virtual machine is scaled up as per the demand on server resources, the fractions of each of the workloads which are running on the scaled up virtual machines will get increased as well. (*See id.)* Likewise, when the demand for server resources is decreased, individual virtual machines are scaled down, so that the fractions of workloads running on the scaled down virtual machines will be decreased as well. (*See id*.)

102. 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 '730 Accused Products, Microsoft has injured Daedalus and is liable to Daedalus for directly infringing one or more claims of the '730 Patent, including without limitation claim 1 pursuant to 35 U.S.C. § 271(a).

103. On information and belief, Microsoft is inducing and/or has induced infringement of one or more claims of the '730 Patent, including at least claim 1, as a result of, amongst other activities, instructing, encouraging, and directing its customers on the use of the '730 Accused Products in an infringing manner in violation of 35 U.S.C. § 271(b). Through its website, instructional guides, and manuals, Microsoft provides its customers with detailed explanations, instructions, and information on how to use and implement the '730 Accused Products which demonstrate active steps taken to encourage direct infringement. (*See*, *e.g.*, Vertical autoscale with virtual machine scale sets; Microsoft Azure Virtual Datacenter: Lift and Shift Guide; How does Microsoft Azure work?). On information and belief, Microsoft has had actual knowledge of the '730 Patent at least as of 2009. Despite this knowledge of the '730 Patent, Microsoft has continued to engage in activities to encourage and assist its customers in the use of the '730 Accused Products. Thus, on information and belief, Microsoft (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.

104. On information and belief, Microsoft has known about the '730 Patent and its contents since at least about June 2009. On information and belief, Microsoft and the inventors of U.S. Patent Application No. 12/492,385 ("Microsoft's 12/492,385 Patent App.") knew of the '730 Patent and its contents when the '730 Patent was cited as a reference in an Information Disclosure Statement by the inventors during the prosecution of Microsoft's 12/492,385 Patent App. The '730 Patent was later also cited as a reference in Information Disclosure Statements during the prosecution of U.S. Patent Application Nos. 12/640,272; 11/437,142; 12/640,318 and U.S. Patent Nos. 8,849,469; 9,595,054 ; 9,450,838; 9,063,738, and 9,207,993 (collectively "Microsoft Additional Patents and Patent Applications"). Thus, on information and belief,

Microsoft and the inventors of Microsoft Additional Patents and Patent Applications also knew of the '730 Patent and its contents during the prosecution of Microsoft Additional Patents and Patent Applications.

105. Furthermore, on information and belief, Microsoft and the inventors of U.S. Patent Application No. 11/437,142 ("Microsoft's 11/437,142 Patent App.") knew of the '730 Patent and its contents when the '730 Patent was cited by the patent examiner in a final office action dated February 18, 2011 as a prior art reference to reject claims for obviousness during the prosecution of Microsoft's 11/437,142 Patent App. Lastly, on information and belief, Microsoft and the inventors of U.S. Patent Application No. 12/651,260 Patent App.") knew of the '730 Patent and its contents when the '730 Patent was cited by the patent examiner as a prior art reference in an office action dated October 24, 2012 during the prosecution of Microsoft's 12/651,260 Patent App. Microsoft, having learned of the likelihood of infringement of the '730 Patent, nevertheless acted in a way that infringed.

106. On information and belief, by using the '730 Accused Products as encouraged and assisted by Microsoft, Microsoft's customers have directly infringed and continue to directly infringe one or more claims of the '730 Patent, including at least Claim 1. On information and belief, Microsoft knew or was willfully blind to the fact that that its actions would induce its customers' direct infringement of the '730 Patent.

107. Microsoft's infringement of the '730 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.

108. On information and belief, Microsoft will continue to infringe the '730 Patent unless enjoined by this Court.

109. As a result of Microsoft's infringement of the '730 Patent, Daedalus has suffered monetary damages, and seeks recovery, in an amount to be proven at trial, adequate to compensate for Microsoft's infringement, but in no event less than a reasonable royalty with interest and costs. Microsoft's infringement of Daedalus' rights under the '730 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. 8,381,209)

110. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-109 of this Complaint as though fully set forth herein.

111. On information and belief, Microsoft has directly infringed and continues to directly infringe one or more claims of the '209 Patent, including at least claim 1 of the '209 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 and services that embody one or more of the inventions claimed in the '209 Patent, including but not limited to the above-identified Azure Site Recovery with Network Security Groups, and all reasonably similar products ("the '209 Accused Products"), in violation of 35 U.S.C. § 271(a).

112. As an example, the '209 Accused Products, including Azure Site Recovery with Network Security Groups, implement a method of "controlling network security of a virtual machine." Azure Site Recovery enables migration to Azure for on-premises virtual machines. (*See* Network Security Groups with Azure Site Recovery). Azure Site Recovery with Network Security Groups enables users to "control[] network security" by applying security rules to client

instances which are "virtual machines." (*See* https://azure.microsoft.com/en-us/services/site-recovery/).

113. The '209 Accused Products' method of controlling network security of a virtual machine "comprise[s] enforcing network security and routing at a hypervisor layer." For example, Azure Network Security Groups contain security rules that allow or deny inbound network traffic to, or outbound network traffic from, several types of Azure resources based on source or destination IP address, port, protocol, etc. (*See* Network security groups, available at https://docs.microsoft.com/en-us/azure/virtual-network/security-overview#network-security-groups; Network Security Groups with Azure Site Recovery). On each Azure physical server node, there is a hypervisor that runs directly over the hardware. The hypervisor divides a node into a variable number of guest virtual machines. (*See* Azure information system components and boundaries). Azure Network Security Groups are associated to subnets or to virtual machines, and cloud services deployed in the classic deployment model, and to subnets or network interfaces in the Resource Manager Deployment model. (*See* How network security groups filter network traffic, available at https://docs.microsoft.com/en-us/azure/virtual-network/network-security-group-how-it-works). Using Azure Site Recovery with Network Security Groups, routing and network security are "enforce[ed]" at the hypervisor layer." (Azure information system components and boundaries.).

114. Using Azure Site Recovery with Network Security Groups, the network security and routing at the hypervisor layer is further enforced "via dynamic updating of routing controls initiated by a migration of said virtual machine from a first device to a second device." For example, Azure Site Recovery enables replication and "migration" for Azure "virtual machines," by creating the replica virtual networks on the target region and creating the required mappings

between the source and target virtual networks." (*See e.g.,* Network Security Groups with Azure Site Recovery). During the migration, Azure Site Recovery with Network Security Groups provide a mechanism to control network security of a virtual machine. Furthermore, a user can "associate NSGs to failed over VMs automatically during failover, using automation scripts with Site Recovery's powerful recovery plans." (*Id.*). Upon failover, a new virtual machine is created on the new device and the hypervisor firewall is implemented in the hypervisor and configured by the fabric controller (FC) agent. By default, when a virtual machine is created, the hypervisor firewall blocks all traffic and then the fabric controller agent adds rules and exceptions in the filter to allow authorized traffic. Azure Network Security Group rules are programmed to define the network access based on the tenants' service model and thus "routing controls" are "dynamically updated." (*See e.g.,* The Azure production network, available at https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/security/fundamentals/production-network.md).

115. 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 '209 Accused Products, Microsoft has injured Daedalus and is liable to Daedalus for directly infringing one or more claims of the '209 Patent, including without limitation claim 1 pursuant to 35 U.S.C. § 271(a).

116. On information and belief, Microsoft is inducing and/or has induced infringement of one or more claims of the '209 Patent, including at least claim 1, as a result of, amongst other activities, instructing, encouraging, and directing its customers on the use of the '209 Accused Products in an infringing manner in violation of 35 U.S.C. § 271(b). Through its website, instructional guides, and manuals, Microsoft provides its customers with detailed explanations,

instructions, and information on how to use and implement the '209 Accused Products which demonstrate active steps taken to encourage direct infringement. (*See*, *e.g.*, Network Security Groups with Azure Site Recovery; Azure information system components and boundaries; Create and customize recovery plans, available at https://docs.microsoft.com/en-us/azure/site-recovery/site-recovery-create-recovery-plans). On information and belief, Microsoft has had actual knowledge of the '209 Patent at least as of 2019. Despite this knowledge of the '209 Patent, Microsoft has continued to engage in activities to encourage and assist its customers in the use of the '209 Accused Products. Thus, on information and belief, Microsoft (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.

117. On information and belief, Microsoft has known about the '209 Patent and its contents since at least about December 2019. On information and belief, Microsoft and the inventors of U.S. Patent No. 10,616,136 ("Microsoft's '136 Patent") knew of the '209 Patent and its contents when the application that eventually became the '209 Patent was cited by the patent examiner as a reference in a Notice of References Cited during the prosecution of Microsoft's '136 Patent. Microsoft, having learned of the likelihood of infringement of the '209 Patent, nevertheless acted in a way that infringed.

118. On information and belief, by using the '209 Accused Products as encouraged and assisted by Microsoft, Microsoft's customers have directly infringed and continue to directly infringe one or more claims of the '209 Patent, including at least Claim 1. On information and belief, Microsoft knew or was willfully blind to the fact that that its actions would induce its customers' direct infringement of the '209 Patent.

119. Microsoft's infringement of the '209 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.

120. On information and belief, Microsoft will continue to infringe the '209 Patent unless enjoined by this Court.

121. As a result of Microsoft's infringement of the '209 Patent, Daedalus has suffered monetary damages, and seeks recovery, in an amount to be proven at trial, adequate to compensate for Microsoft's infringement, but in no event less than a reasonable royalty with interest and costs. Microsoft's infringement of Daedalus' rights under the '209 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. 8,572,612)

122. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-121 of this Complaint as though fully set forth herein.

123. On information and belief, Microsoft 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 Microsoft Azure VMSS products, and all reasonably similar products ("the '612 Accused Products"), in violation of 35 U.S.C. § 271(a).

124. As an example, the '612 Accused Products, such as Microsoft Azure Virtual Machine Scale Sets ("VMSS"), implement "a method of autonomic scaling of virtual machines in a cloud computing environment" that allows for dynamic deploying and maintenance of virtual machine instances in Azure cloud, wherein the virtual machine scaling is automatically adjusted based on the workload increase or decrease in load. (*See* Microsoft Azure Virtual Machine Scale Sets). Microsoft Azure is "a cloud computing environment comprising a plurality of virtual machines ('VMs')." Azure is a cloud computing platform and infrastructure for building, deploying, and managing applications and services through a network of datacenters, wherein Azure creates virtual machines (VMs) based on the resource need. (*See* Azure information system components and boundaries, available at https://docs.microsoft.com/en-us/azure/security/fundamentals/infrastructure-components). Microsoft Azure VMSS offers multiple virtual machine instances, the number of which can be automatically adjusted by its autoscaling feature. (*See* Azure Virtual Machine Scale Sets; Automatically scales a virtual machine scale set in the Azure portal, available at https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/virtual-machine-scale-sets-autoscale-portal; Azure Virtual Machine Scale Sets; what are virtual machine scale sets?, available at https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/overview).

125. Microsoft Azure VMSS's virtual machines contain "modules of automated computing machinery installed upon computers disposed within a data center." For instance, in Azure, a virtual machine is comprised of multiple operating systems that run side-by-side with a hypervisor to manage them. (*See* What is a virtual machine?, available at https://azure.microsoft.com/en-in/overview/what-is-a-virtual-machine/*)*. In Azure VMSS, these virtual machines are "installed upon cloud computers," and these virtual machines run on physical servers

(blades/nodes). (*See* What is a virtual machine?; Azure information system components and boundaries). Further, applications run on virtual machine instances in a scale set require installation of the modules of automated computing machinery. (*See* Azure-docs, available at https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/virtual-machine-scale-sets/virtual-machine-scale-sets-deploy-app.md). In Microsoft Azure, the "cloud computers" are disposed within an Azure datacenter. (*See* Virtual Machine Troubleshooting Azure VM allocations, available at https://docs.microsoft.com/en-us/azure/virtual-machines/troubleshooting/allocation-failure).

126. Microsoft Azure is a "cloud operating system" in Microsoft's cloud computing environment. (*See* Azure Virtual Machine Scale Sets; Automatically scales a virtual machine scale set in the Azure portal). Microsoft Azure further comprises a Fabric Controller which independently manages the virtual machines running on servers and is therefore "a data center administration server operably coupled to the virtual machines." (*See* Azure Virtual Machine Scale Sets; Automatically scales a virtual machine scale set in the Azure portal). That is, the Fabric Controller is installed in the same physical blade server as the virtual machines, and the fabric controller provides the data center level functions to manage virtual machines in the cloud. (*See id.*)

127. Microsoft Azure's "cloud operating system" "deploy[s]…an instance of a virtual machine." Azure runs on a base operating system and Azure VMSS is a part of the base operating system, wherein the VMSS deploys an instance of a virtual machine on Azure cloud. (*See* Azure Virtual Machine Scale Sets; Automatically scales a virtual machine scale set in the Azure portal; Azure Virtual Machine Scale Sets; what are virtual machine scale sets?). For example, Azure deploys an operating system in which the kernel and other core components

have been modified and optimized to support the Azure environment, where Azure creates virtual machine instances based on resource need. (*See id*.)

128. Microsoft Azure VMSS deploys "an instance of a virtual machine, including flagging the instance of a virtual machine for autonomic scaling." Specifically, Azure VMSS helps a user create virtual machines based on certain user defined rules, and when those defined thresholds are met, autoscale rules take action to adjust the capacity of the user's scale set. (*See* Azure Virtual Machine Scale Sets; Automatically scales a virtual machine scale set in the Azure portal). Via Microsoft Azure VMSS's autoscaling feature, the virtual machines are automatically adjusted based on a user defined configuration. (*See id*.). The autoscaling configuration "flags the instance of a virtual machine for autonomic scaling" by specifying the thresholds that the performance metric must reach to trigger a scaling event. (*See id.*)

129. Microsoft Azure VMSS's autonomic scaling "include[es] terminating and executing a data processing workload on the instance of a virtual machine." For instance, when resource demand declines, the autonomic scaling feature of VMSS can terminate the additional virtual machines. (*See* Azure Virtual Datacenter: Lift and Shift Guide). Furthermore, the autonomic scaling can execute a data processing workload on the instance of a virtual machine, for example, when the virtual machine instances are created and the user's applications are deployed, the scale set starts to distribute traffic to them through the load balancer. (*See* Azure Virtual Machine Scale Sets, what are virtual machine scale sets?).

130. Microsoft Azure VMSS "monitor[s]" the "operating characteristics of the instance of the virtual machines." The VMSS enables application monitoring for Azure virtual machines and Azure virtual machine scale sets. (*See* Deploy the Azure Monitor Application Insights Agent on Azure virtual machines and Azure virtual machine scale sets, available at

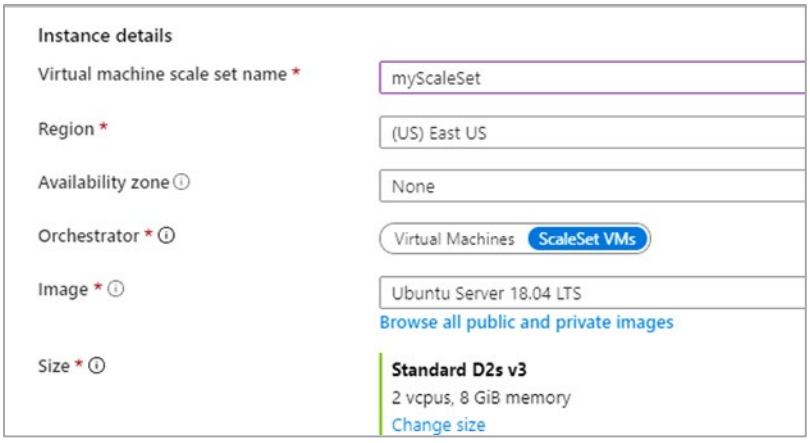
https://docs.microsoft.com/en-us/azure/azure-monitor/app/azure-vm-vmss-apps). For example, VMSS monitors operating characteristics such as CPU, memory, disk, and network performance counters from the virtual machines. (*See* Azure Virtual Machine Scale Sets; what are virtual machine scale sets?).

131. Microsoft Azure VMSS "deploy[s]…an additional instance of the virtual machine if a value of an operating characteristic exceeds a first predetermined threshold value." With Azure virtual machine scale sets, the number of virtual machines can automatically increase in response to demand or a defined schedule. (*See* Azure Virtual Machine Scale Sets; what are virtual machine scale sets?). For example, a user can create a rule that increases the number of virtual machine instances in a scale set when an operating characteristic, such as CPU load, is greater than 70% over a 10-minute period. (*See* Automatically scale a virtual machine scale set in the Azure portal). And when the CPU utilization threshold value is met, the number of virtual machine instances in increased by 20%. (*See id*.) When these virtual machine instances are created and the user's applications are deployed, the scale set starts to distribute traffic to them through the load balancer, thereby "executing a portion of the data processing workload on the additional instance of the virtual machine." (*See id*.).

132. Microsoft Azure VMSS "terminat[es 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 load is less than 30% over a 10-minute period, the number of virtual machine instances is decreased by 20%. (*See* Automatically scale a virtual machine scale set in the Azure portal)

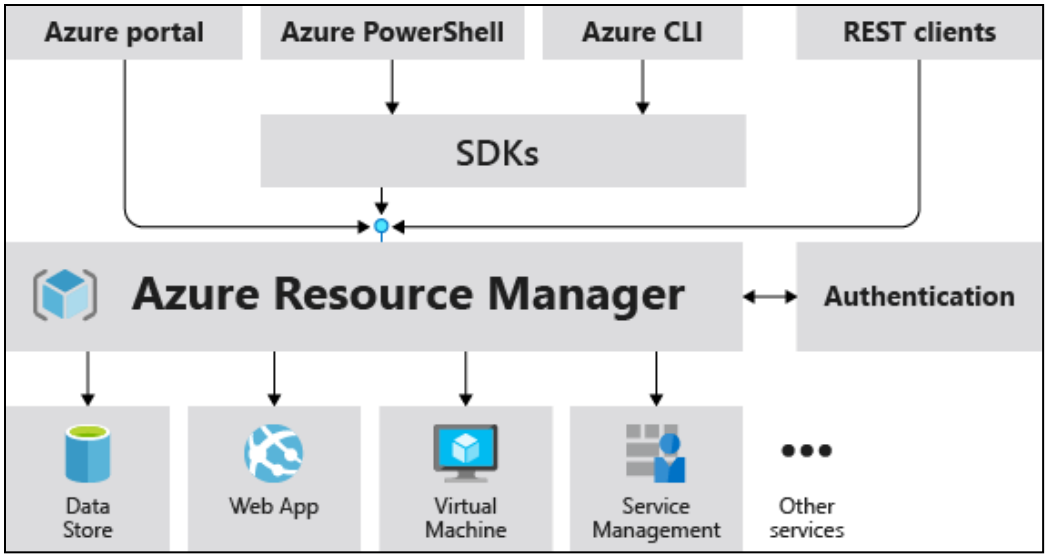
133. Each Azure physical server node of Microsoft Azure VMSS has one root virtual machine, which runs the host operating system, thereby acting as "a module of automated

computing machinery." (*See* Azure Virtual Machine Scale Sets; what are virtual machine scale sets?). Azure further "compris[es] a self service portal," as Azure cloud computing platform provides an Azure portal where a user can define the rules for autoscaling. (*See id.*). The Azure portal enables the user to create a virtual machine scale set, in which the user can scale the number of virtual machines in the scale set manually, or define rules to autoscale based on resource usage like CPU, memory demand, or network traffic. (*See* Quickstart: Create a virtual machine scale set in the Azure portal, available at https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/quick-create-portal; Quickstart: Create a Linux virtual machine scale set with an ARM template, available at https://docs.microsoft.com/en-us/azure/virtual-machine-scale-sets/quick-create-template-linux; *see also:*



Quickstart: Create a virtual machine scale set in the Azure portal).

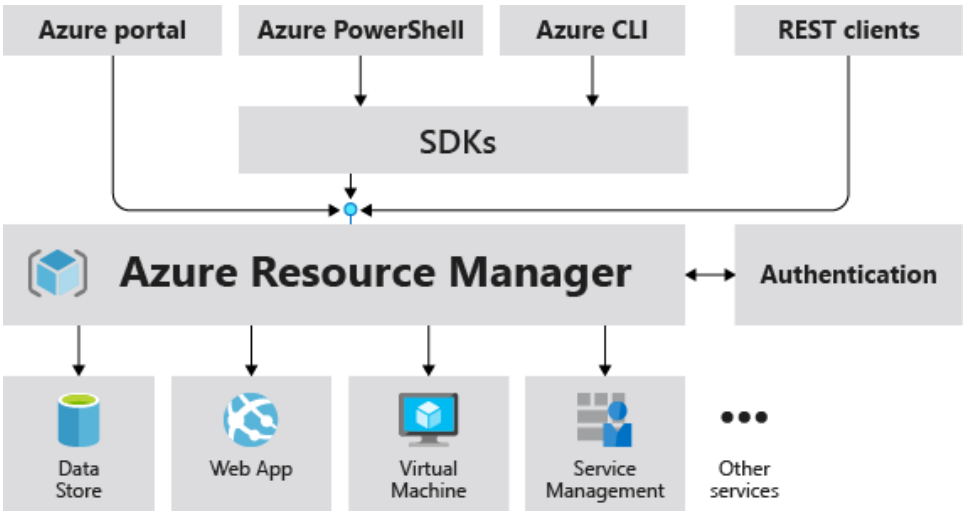
134. The Azure cloud computing platform further comprises an Azure Resource Manager as "a deployment engine," as the Azure Resource Manager is the deployment and management service for Azure. (*See* What is Azure Resource Manager?). Azure's VMSS feature allows the user to create a virtual machine scale set and deploy a sample application with an Azure Resource Manager template with user specifications. (*See* Quickstart: Create a Linux virtual machine scale set with an ARM template; *see also:*



What is Azure Resource Manager?).

135. Microsoft Azure's VMSS feature "deploy[s] an instance of a virtual machines" on Azure cloud. For example, an Azure virtual machine scale set can automatically increase or decrease the number of virtual machine instances that run the user's application, as a virtual machine scale set allows the user to deploy and manage a set of identical, auto-scaling virtual machines. (*See* Quickstart: Create a virtual machine scale set in the Azure portal).

136. These virtual machine instances are deployed on Azure cloud, wherein the "self service portal," such as an interface, receives "user specifications" from the user, e.g., the name of the virtual machine, type and the size of CPU. (*See* Quickstart: Create a virtual machine scale set in the Azure portal). Those user specifications for the virtual machine instance are further "pass[ed]" by the Azure portal to the Azure Resource Manager, which is Azure's "deployment engine." (*See id.*; What is Azure Resource Manager?). As shown below, the user utilizes Azure portal to create the virtual machine specifications, and after the user specifies the virtual machine specifications, those user specifications are passed to the Azure Resource Manager to create an Azure Resource Manager template using the user specifications. (*See id.*; *see also:*



What is Azure Resource Manager?).

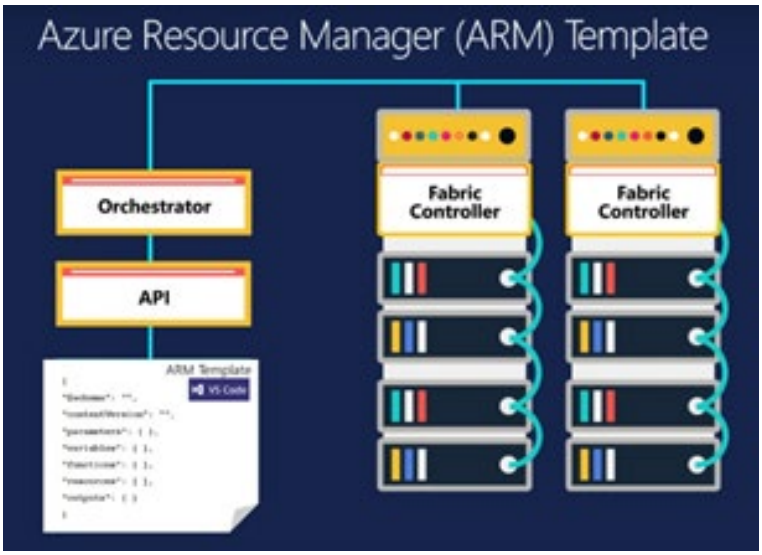
137. In Azure, the deployment engine further "implement[s] and pass[ses] to the data center administration server…a virtual machine template with user specifications." Specifically, Azure Resource Manager, the "deployment engine," creates a Azure Resource Manager template, "a virtual machine template with user specifications" based on virtual machine specifications. (*See* Quickstart: Create a Linux virtual machine scale set with an ARM template; *see also:*

To create a scale with a template, you define the appropriate resources. The core parts of the virtual machine scale set resource type are:

Property	Description of property	Example template value
type	Azure resource type to create	Microsoft.Compute/virtualMachineScaleSets
name	The scale set name	myScaleSet
location	The location to create the scale set	East US

Id.). *Azure* Resource Manager then "passes" the Azure Resource Manager template to the fabric controller, "the data center administration server," to allocate Azure resources for virtual machines. (*See* Azure information system components and boundaries; How does Azure work?,

available at https://docs.microsoft.com/bs-cyrl-ba/azure/cloud-adoption-framework/get-started/what-is-azure*;* Azure Resource Manager (ARM) Template, available at https://azurehangout.com/azure-resource-manager-arm-template-structure-01-schema-element/; *see also:*



(*See* Azure Resource Manager (ARM) Template).

138. In Azure, the fabric controller "call[s]" an Azure hypervisor to "install the virtual machine template as an instance of a virtual machine on the cloud computer." The fabric controller communicates with the hypervisor to "install" the Azure Resource Manager template to create the virtual machine on the Azure cloud. (*See* Azure information system components and boundaries; *see also* Quickstart: Create a virtual machine scale set in the Azure portal).

139. 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, Microsoft 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).

140. On information and belief, Microsoft 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, Microsoft 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.*, Azure Virtual Datacenter: Lift and Shift Guide; Automatically scales a virtual machine scale set in the Azure portal; Deploy the Azure Monitor Application Insights Agent on Azure virtual machines and Azure virtual machine scale sets; Deploy the Azure Monitor Application Insights Agent on Azure virtual machines and Azure virtual machine scale sets; What is Azure Resource Manager; Azure Resource Manager (ARM) Template). On information and belief, Microsoft has had knowledge of the '612 Patent at least as of 2014. Despite this knowledge of the '612 Patent, Microsoft 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, Microsoft (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.

141. On information and belief, Microsoft has known about the '612 Patent and its contents since at least about June 2014. On information and belief, Microsoft and the inventors of U.S. Patent 9,722,945 ("Microsoft's '945 Patent") knew of the '612 Patent and its contents when the '612 Patent was cited as a reference in an Information Disclosure Statement by the inventors during the prosecution of Microsoft's '945 Patent. The '612 Patent was further cited as a reference in Information Statements during the prosecution of U.S. Patent Nos. 9,847,918;

9,665,432; 9,842,039 (collectively, "Microsoft Additional Patents"). Thus, on information and belief, Microsoft and the inventors of Microsoft Additional Patents knew of the '612 Patent and its contents during the prosecution of Microsoft Additional Patents. Lastly, On information and belief, Microsoft and the inventors of U.S. Patent No. 9,858,153 ("Microsoft's '153 Patent") knew of the '612 Patent and its contents when the '612 Patent was cited as a reference in a Notice of References Cited by the patent examiner during the prosecution of Microsoft's '153 Patent. Microsoft, having learned of the likelihood of infringement of the '612 Patent, nevertheless acted in a way that infringed.

142. On information and belief, by using the '612 Accused Products as encouraged and assisted by Microsoft, Microsoft'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, Microsoft knew or was willfully blind to the fact that that its actions would induce its customers' direct infringement of the '612 Patent.

143. Microsoft'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.

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

145. As a result of Microsoft'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 Microsoft's infringement, but in no event less than a reasonable royalty with interest and costs. Microsoft'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.

FIFTH COUNT

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

146. Daedalus incorporates by reference the allegations set forth in Paragraphs 1-145 of this Complaint as though fully set forth herein.

147. On information and belief, Microsoft 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 Microsoft Azure Blog Storage, and all reasonably similar products (" the '132 Accused Products"), in violation of 35 U.S.C. § 271(a).

148. As an example, the '132 Accused Products, such as Microsoft Azure Blob Storage, implement a method for "handling files within a policy-based data management system" to transition data between multiple access tiers. (*See* Manage the Azure Blob storage lifecycle).

149. Microsoft Azure Blob Storage's method for transitioning data "provides a policy set comprising at least one service class rule." For example, up to 100 rules can exist within Azure Blob Storage's rich, rule-based policy, but at least one rule is required in a policy. (*See* Manage the Azure Blob storage lifecycle). Each rule within a policy is a "service class rule" that defines the actions to be taken based on the files' attributes. (*Id.*).

150. Microsoft Azure Blob Storage "receives one or more attributes of a file from one of a plurality of clients." For example, using Azure Storage Explorer, files are transferred from a

client's local disk(s) to Azure Blob Storage. (*See* Quickstart: Use Azure Storage Explorer to create a blob, available at https://docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-storage-explorer). Attributes of a file received by Microsoft Azure Blob Storage include file name, file size, last modification date and time. (*Id.*).

151. The Azure Storage Explorer is configured to communicate with "clients comprising at least two different computing platform" as it can be installed on systems with Windows, Macintosh, or Linux OS and is thus (*See* Quickstart: Use Azure Storage Explorer to create a blob).

152. Additionally, in Microsoft Azure Blob Storage, "service class rules" are "applied" to the files. For example, rules in Azure Blob Storage's policy dictate what "[a]ctions are applied to the filtered blobs when the run condition is met." (*See* Manage the Azure Blob storage lifecycle). One sample rule is to "[t]ier blob to cool tier 30 days after last modification." (*Id.*). After the service class rule is applied to the file, the file is "assigned a service class." *See:*

```
"actions": {
  "baseBlob": {
    "tierToCool": { "daysAfterModificationGreaterThan": 30 },
    "tierToArchive": { "daysAfterModificationGreaterThan": 90 },
    "delete": { "daysAfterModificationGreaterThan": 2555 }
  },
  "snapshot": {
    "delete": { "daysAfterCreationGreaterThan": 90 }
  }
```

(*Id.*). In the above example, the service class "tierToCool" is assigned. "The policy transitions blobs that haven't been modified in over 30 days to cool storage." (*Id.*).

153. Once a file has been assigned a service class, Microsoft Azure Blob Storage "conducts operations on the file in a manner according to the service class." For example, once a

file in the hot tier has been assigned "tierToCool" because it has not been modified in over 30 days, it will be transitioned to the cool tier. (*See* ManagementPolicyBaseBlob.TierToCool Property, available at https://docs.microsoft.com/en-us/dotnet/api/microsoft.azure.management.storage.fluent.models.managementpolicybaseblob.tiertocool?view=azure-dotnet; *see also:*

Gets or sets the function to tier blobs to cool storage. Support blobs currently at Hot tier

C# Copy

```
[Newtonsoft.Json.JsonProperty(PropertyName="tierToCool")]
public Microsoft.Azure.Management.Storage.Fluent.Models.DateAfterModification
TierToCool { get; set; }
```

(*Id.*).

154. 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, Microsoft has injured Daedalus and is liable to Daedalus for directly infringing one or more claims of the '132 Patent, including without limitation claim 15 pursuant to 35 U.S.C. § 271(a).

155. On information and belief, Microsoft is inducing and/or has induced infringement of one or more claims of the '132 Patent, including at least claim 15, as a result of, amongst other activities, instructing, encouraging, and directing its customers on the use of the '132 Accused Products in an infringing manner in violation of 35 U.S.C. § 271(b). Through its website, instructional guides, and manuals, Microsoft provides its customers with detailed explanations, instructions, and information on how to use and implement the '132 Accused Products which demonstrate active steps taken to encourage direct infringement. (*See*, *e.g.*, Manage the Azure Blob Storage lifecycle; QuickStart: Use Azure Storage Explorer to create a blob; Blob storage,

available at https://azure.microsoft.com/en-us/services/storage/blobs/; Introduction to Azure Blob Storage, available at https://docs.microsoft.com/en-us/azure/storage/blobs/storage-blobs-introduction; Manage Azure Blob Storage resources with Storage Explorer, available at https://docs.microsoft.com/en-us/azure/vs-azure-tools-storage-explorer-blobs). On information and belief, Microsoft has had knowledge of the '132 Patent at least as of 2012. Despite this knowledge of the '132 Patent, Microsoft has continued to engage in activities to encourage and assist its customers in the use of the '132 Accused Products. Thus, on information and belief, Microsoft (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.

156. On information and belief, Microsoft has known about the '132 Patent and its contents since at least about June 2012. On information and belief, Microsoft and the inventors of U.S. Patent No. 9,501,656 ("Microsoft's '656 Patent") knew of the '132 Patent and its contents when the application that eventually became the '132 Patent was cited as a reference in an Information Disclosure Statement by the inventors during the prosecution of Microsoft's '656 Patent. Similarly, on information and belief, Microsoft and the inventors of U.S. Patent No. 9,678,839 ("Microsoft's '839 Patent") knew of the '132 Patent and its contents when the application that eventually became the '132 Patent was cited as a reference in an Information Disclosure Statement by the inventors during the prosecution of Microsoft's '839 Patent.

157. On information and belief, Microsoft and the inventors 9,037,541 ("Microsoft's '541 Patent") also knew of the '132 Patent and its contents when the application that eventually became the '132 Patent was cited by the patent examiner in a final office action dated June 7, 2013 as a prior art reference to reject claims for obviousness during the prosecution of Microsoft's '541 Patent. In fact, during prosecution, not only were claims amended to overcome

these rejections, but detailed analyses of the differences between of Microsoft's '541 Patent and the '172 Patent (including quotes from the '172 Patent) were submitted in the Applicant Remarks.

158. Furthermore, on information and belief, Microsoft and the inventors of U.S. Patent No. 9,244,615 ("Microsoft's '615 Patent") also knew of the '132 Patent and its contents when the application that eventually became the '132 Patent was cited by the patent examiner in an office action dated March 2, 2015 as a prior art reference to reject claims for obviousness during the prosecution of Microsoft's '615 Patent. In response, Microsoft amended a claim to overcome the rejections and an analysis of the differences between of Microsoft's '615 Patent and the '172 Patent were submitted in the Applicant Remarks.

159. Lastly, on information and belief, Microsoft and the inventors of U.S. Patent No. 9,880,759 ("Microsoft's '759 Patent") knew of the '132 Patent and its contents when the application that eventually became the '132 Patent was cited by the patent examiner in an office action dated October 6, 2016 as a prior art reference to reject claims for obviousness during the prosecution of Microsoft's '759 Patent. Similarly, during prosecution, not only were claims amended to overcome the rejections, but detailed analyses of the differences between of Microsoft's '759 Patent and the '132 Patent (including quotes from the '132 Patent) were submitted in the Applicant Remarks. Microsoft, having learned of the likelihood of infringement of the '132 Patent, nevertheless acted in a way that infringed.

160. On information and belief, by using the '132 Accused Products as encouraged and assisted by Microsoft, Microsoft's customers have directly infringed and continue to directly infringe one or more claims of the '132 Patent, including at least claim 15. On information and

belief, Microsoft knew or was willfully blind to the fact that that its actions would induce its customers' direct infringement of the '132 Patent.

161. Microsoft's infringement of the '132 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.

162. On information and belief, Microsoft will continue to infringe the '132 Patent unless enjoined by this Court.

163. As a result of Microsoft'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 Microsoft's infringement, but in no event less than a reasonable royalty with interest and costs. Microsoft'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.

PRAYER FOR RELIEF

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

a. For judgment that Microsoft has infringed and continues to infringe the claims of the '886, '730, '209, '612, and '132 Patents;

b. For a permanent injunction against Microsoft 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 '886, '730, '209, '612, and '132 Patents;

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

d. For a mandatory future royalty payable on each and every future sale by Microsoft 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 Microsoft's infringement is willful and awarding to Plaintiff enhanced damages pursuant to 35 U.S.C. § 284;

f. For a judgment and order requiring Microsoft to pay Plaintiff's damages, costs, expenses, and pre- and post-judgment interest for its infringement of the '886, '730, '209, '612, and '132 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: December 16, 2020

Respectfully Submitted,

/s/ B. Russell Horton

Denise M. De Mory (*Pro Hac Pending)*
Cal. Bar No. 168076
ddemory@bdiplaw.com
Jennifer L. Gilbert (*Pro Hac Pending)*
Cal. Bar No. 255820
jgilbert@bdiplaw.com
Robin Curtis (*Pro Hac Pending)*
Cal. Bar No. 271702
rcurtis@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