UNITED STATES DISTRICT COURT
WESTERN DISTRICT OF TEXAS
AUSTIN DIVISION

EDGE NETWORKING SYSTEMS, LLC **PLAINTIFF,** **V.** **MICROSOFT CORPORATION,** **DEFENDANT.**	**CIVIL ACTION NO.:** 24-cv-215 **JURY TRIAL DEMANDED**

COMPLAINT FOR PATENT INFRINGEMENT

NATURE OF THE CASE

1. This action is for patent infringement arising under the patent laws of the United States, 35 U.S.C. §§ 1, et seq. As further stated herein, Edge Networking Systems, LLC ("Edge Networking") alleges that Microsoft infringes one or more claims of patents owned by Edge Networking. Accordingly, Edge Networking seeks monetary damages and injunctive relief in this action.

THE PARTIES

2. Plaintiff Edge Networking Systems, LLC is a Texas company with its principal place of business at 5900 Balcones Drive, Ste. 100, Austin Texas 78731.

3. On information and belief, Microsoft is a corporation organized and existing under the laws of the state of Washington, with a principal place of business in this district located at 10900 Stonelake Boulevard, Suite 225, Austin, Texas 78759.

JURISDICTION AND VENUE

4. This Court has exclusive subject matter jurisdiction over this case pursuant to 28 U.S.C. §§ 1331 and 1338(a) on the grounds that this action arises under the Patent Laws of the United States, 35 U.S.C. § 1 et seq., including, without limitation, 35 U.S.C. §§ 271, 281, 284, and 285.

5. This Court has personal jurisdiction over Microsoft because it has conducted and continues to regularly conduct business within the State of Texas and this District. Microsoft has purposefully and voluntarily availed itself of the privileges of conducting business in the United States, the State of Texas, and this District by continuously and systematically placing goods into the stream of commerce through an established distribution channel with the expectation that they will be purchased by consumers in this District. Microsoft directly and/or through intermediaries (including distributors, sales agents, and others), ships, distributes, sells, offers to sell, imports, advertises, makes, and/or uses its products (including but not limited to the products accused of infringement herein) in the United States, the State of Texas, and this District.

6. Upon information and belief, Microsoft conducts business within the State of Texas and in this district and has designated Corporation Service Company d/b/a CSC − Lawyers Incorporating Service Company, 211 E. 7th Street, STE 620, Austin, Texas 78701-3218, as its agent for service of process in this district.

7. On information and belief, Microsoft has been registered to do business within the State of Texas under Texas Secretary of State File Number 0010404606 since about March 1987.

8. On information and belief, Microsoft maintains one or more of its data centers in this district in furtherance of infringing acts in this district. For example, Microsoft maintains data

centers in this district, located at: 5150 Rogers Road, San Antonio, Texas 78251; 3823 Wiseman Boulevard, San Antonio, Texas 78251; and 5200 Rogers Road, San Antonio, Texas 78251.

9. On information and belief, Microsoft has operated data centers supporting Microsoft products and services within the State of Texas, and within this district, since at least 2008. Microsoft is building at least three additional data centers in this district, including two data centers located at: 3545 Wiseman Boulevard, San Antonio, Texas 78251, and another data center located at 15000 Block Lambda Drive, San Antonio, Texas 78245. Upon information and belief, Microsoft's data centers, including those in this district, include computer hardware (*e.g.*, memory and processors) that store and execute software that performs some of the actions that infringe on the patents in the lawsuit. On information and belief, Microsoft has employed, is employing, and is offering to employ individuals in this district in furtherance of infringing acts in this district. On information and belief, these employees have direct personal knowledge about the accused products and Microsoft's infringing activities. For example, Justin Savill, Chief Architect at Microsoft, purporting according to his LinkedIn profile to be working at Microsoft in Austin, Texas and is responsible for architecting Microsoft's Azure platform.

10. On information and belief, the below contains a non-exhaustive listing of several Microsoft employees located in Texas that are key fact witnesses and have knowledge that is relevant to the claims (and anticipated defenses) in this action:

- Steve Bruns, Principal Software Development Manager at Microsoft who leads a crew of engineers in the Microsoft Commercial Software Engineering organization;
- Jordan Olshevski, Principal Software Engineer at Microsoft for Azure Kubernetes Service ("AKS");
- Shay Patel, Senior Engineering Architect at Microsoft for Azure Data;

- Ola Sowemimo, Senior Software Engineer at Microsoft;
- Candice Ripley, Senior Software Engineer at Microsoft;
- Oliver King, Software Engineer II at Microsoft for AKS;
- Varun Ojha, Software Engineer II at Microsoft;
- Jorge Rangel, Software Engineer II at Microsoft;
- Pavan Poladi, Software Engineer at Microsoft;
- Michael Muruthi, Software Engiener at Microsoft
- Whitney Lampkin, Site Reliability Engineer at Microsoft for AKS;
- Adeboye Famurewa, Azure Engineer at Microsoft for AKS; Mr. Famurewa designs, deploys and manages AKS;
- Karan Thakur, AKS Engineer at Microsoft;
- Yamaan Qaddoura, Technical Advisor at Microsoft who leads a team of engineers as a technical advisor and people manager for the AKS team; and
- Joseph Amedeo, Software Engineer II at Microsoft, who is developing a usage processing pipeline to track usage for Azure users;

11. On information and belief, Microsoft has operated permanent office facilities within the State of Texas, and within this district, since at least 2000. Microsoft operates offices in Austin, Texas for the purpose of selling, promoting, maintaining, and providing support for a suite of products, including the accused products.

12. On information and belief, Microsoft maintains a "Corporate Sales Offices" in Austin, Texas at the following address: 10900 Stonelake Boulevard, Suite 225 Austin, TX, 78759; and Microsoft maintains a "Corporate Sales Office" in San Antonio, Texas at the following address: Concord Park II 401 East Sonterra Boulevard, Suite 300, San Antonio, TX, 78258.

13. On information and belief, one or more of the Accused Products are used, offered for sale, and sold in this district, including by Microsoft and by "Microsoft-certified resellers" (e.g., Heart of Texas Network Consultants, located at 703 Willow Grove Rd., Waco, Texas 76712).

14. On information and belief, Microsoft operated at least ten physical stores throughout Texas, some of which were in this district, from 2009 until they were all closed in 2020. During that time period, Microsoft had physical stores that sold Microsoft's products at least at the following addresses: (a) 3309 Esperanza Crossing, Austin, TX 78758 and (b) 15900 La Cantera Parkway The Shops at La Cantera San Antonio, TX 78256.

15. Edge Networking's causes of action arise directly from Microsoft's business contacts and other activities in the State of Texas and this District.

16. Microsoft has derived substantial revenues from its infringing acts within the State of Texas and this District.

17. Venue is proper in the Western District of Texas pursuant to 28 U.S.C. §§ 1391 and 1400(b) because Microsoft maintains regular and established places of business in this district and has committed acts of infringement within this district giving rise to this action.

18. Microsoft has committed acts of infringement in this District and does business in this District, including making sales and/or providing service and support for customers and/or end-users in this District. Microsoft purposefully and voluntarily sold one or more infringing products with the expectation they would be purchased in this District. These infringing products have been and continue to be purchased in this District and with ongoing service by servers and support technicians located in this District. Thus, Microsoft has committed acts of infringement within the United States, the State of Texas, and this District.

19. Furthermore, Microsoft maintains corporate sales offices in this district, which, on information and belief, provide sales and support for the infringing products.

PATENTS-IN-SUIT

20. On June 16, 2020, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 10,686,871 (the "871 Patent") entitled "Distributed Software Defined Networking." A true and correct copy of the 871 Patent is attached hereto as **Exhibit 1**.

21. On January 12, 2021, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 10,893,095 (the "095 Patent") entitled "Distributed Software Defined Networking." A true and correct copy of the 871 Patent is attached hereto as **Exhibit 2**.

22. On July 4, 2023, the United States Patent and Trademark Office duly and legally issued U.S. Patent No. 11,695,823 (the "823 Patent") entitled "Distributed Software Defined Networking." A true and correct copy of the 871 Patent is attached hereto as **Exhibit 3**.

23. Edge Networking is the sole and exclusive owner of all right, title, and interest to and in, or is the exclusive licensee with the right to sue for the 871, 095, and 823 Patents, and holds the exclusive right to take all actions necessary to enforce its rights to the 871, 095, and 823 Patents, including the filing of this patent infringement lawsuit. Edge Networking also has the right to recover all damages for past, present, and future infringement of the 871, 095, and 823 Patents and to seek injunctive relief as appropriate under the law.

24. The Patents-in-Suit were co-invented by Pouya Taaghol. Mr. Taaghol has a rich history of being at the top of his field, including his history as the Chief Technology Officer ("CTO") of both Cisco's Smart Home Group and Intel's Mobile Wireless Group. In his roles as CTO at these major companies, he led the evolution of their technology, intellectual property strategies, ecosystem development, business acquisition, and divesture. Mr. Taaghol has also held

senior technical positions at Motorola and NEC. He is an inventor on over 71 granted and pending patents in wireless/networking/security/big data, and 150+ publications and contributions to standards bodies. Mr. Taaghol holds a Bachelor of Science in Electrical Engineering and Telecommunications and a Ph.D. in Mobile and Satellite Communications from the University of Surrey.

25. The Patents-in-Suit were co-invented by Vivek Ramanna.

FACTUAL ALLEGATIONS

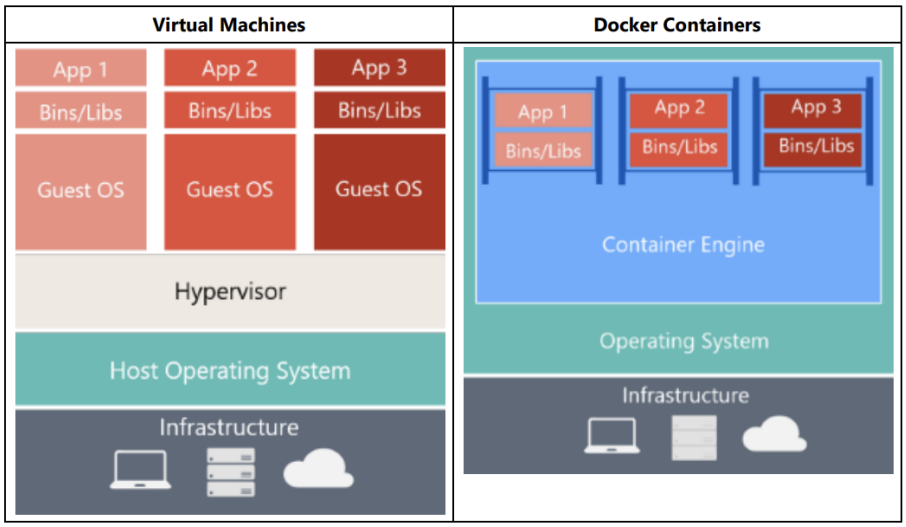
I. TECHNOLOGY BACKGROUND

26. Modern software is typically run in "the cloud." The cloud is a distributed collection of servers that host software and infrastructure, and it is accessed over the Internet. A plethora of software is run in the cloud, including but not limited to video games (on dedicated consoles, such as the Xbox, as well as computers), smartphone apps, office software (like Microsoft 365, and many web-based applications. Regarding web-based applications, like e-stores and cloud apps used on an internet browser, the pervasive use of cloud backends is more apparent.

27. Running software in the cloud requires servers to host the software. Traditionally, servers were physical machines dedicated to hosting a few software packages. But demand for these physical servers constantly fluctuated. Just like a gas station can only accommodate so many customers, a physical server can only accommodate so many users of software at any given time. When there are more users than the server can accommodate a queue of forms – and users/customers are left waiting for their turn. Conversely, during slow periods, there is underutilization of unused resources that costs money to maintain while not generating revenue.

28. Virtual machines offer one solution to the fluctuating demand problem. A virtual machine is essentially a digital copy of a physical machine running within a physical machine. Because virtual machines run within a physical machine, a host operating system is necessary to

give access to the infrastructure of the physical machine and a hypervisor is needed to abstract the hardware from that operating system, to be used by multiple operating systems at once. Additionally, being copies of physical machines, each virtual machine needs a its own operating system and all libraries necessary to run a given piece of software. *See* Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, page 4, Microsoft, Edition v7.0 (2023) ("For VMs, there are three base layers in the host server, from the bottom-up: infrastructure, Host Operating System and a Hypervisor and on top of all that each VM has its own OS and all necessary libraries."). During periods of high demand, the number of virtual machines could be increased, and then decreased during low demand. A representation of virtual machines is shown in the left side of the below figure:



Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, Figure 2-3, Microsoft, Edition v7.0 (2023)

29. However, virtual machines came with their own set of problems. As depicted in the above figure, virtual machines virtualize an entire operating system and run multiple instances of complete operating systems on a single host. Multiple operating systems running on a single host result in higher resource overhead. Additionally, the guest operating systems require additional

configuration and management to avoid conflicts with the host operating system because guest operating systems must interface with the host operating system to get to the physical machines resources. This is akin to attempting to increase the number of gas pumps at a gas station by connecting new pumps to an existing pump. For each new pump to operate, it first must turn the existing pump to which it is connected. If this fails, the new pump will not work. Additionally, one new pump may inadvertently request the entire pump to shut off, thereby stopping the flow of gas to the other pumps attached to the existing pump. Furthermore, because the newly added pumps do not have direct access to the storage, their output is limited by that of the existing pump to which they are attached. Likewise, while virtual machines allow for scaling to meet demand, they don't provide agility, portability, or efficient resource utilization.

30. To provide for more efficient resource utilization, agility, and portability, containers to were developed. "Containerization is an approach to software development in which an application or service, its dependencies, and its configuration (abstracted as deployment manifest files) are packaged together as a container image. The containerized application can be tested as a unit and deployed as a container image instance to the host operating system (OS)." Cesar de la Torre, .NET Microservices: Architecture for Containerized .NET Applications 1 (2023). Containers require fewer resources than virtual machines, thereby allowing them to have a higher density on a given physical machine. *Id.*

31. Though containers allowed better use of backend resources, they did not provide efficient scalability. Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 40, Microsoft, Edition v7.0 (2023) ("In production environments, you could have an Application Delivery Controller (ADC) like Azure Application Gateway between your microservices and the Internet… However, when you build large and complex microservice-based

applications (for example, when handling dozens of microservice types), and especially when the client apps are remote mobile apps or SPA web applications, that approach faces a few issues.").

32. The invention presented by the patents in this suit introduced a novel solution to the problem: a managed, distributed application architecture that allows for microservices to be deployed in special containers as a collection of services. Microsoft precisely utilizes this structure and infringes the patents as set forth herein.

II. INFRINGEMENT ALLEGATIONS

1. The Accused Products and Microsoft's Infringement

33. Through its own actions Microsoft has manufactured, marketed, sold, offered for sale, and exported from and imported into the United States products built on a managed distributed application architecture, such as, for example, games like Forza Horizon 5 (hereafter "Forza"), which is published by Microsoft and available on the Microsoft Xbox platform and well as on Windows, and Microsoft's market leading family of productivity software, Microsoft 365 (hereafter "365"), that directly and/or indirectly infringes (literally or via the doctrine of equivalents) the Patents-in-Suit.

34. Forza is just one example of an Xbox Games Studios game that infringes upon the Patents-in-Suit that, upon information and belief, infringes in the same way.

35. The primary building blocks of a managed distributed application architecture are a cloud infrastructure, divided into network devices and cloud devices, providing a platform for running containerized applications that communicate via a virtual fabric and are managed by a service. Such an architecture is utilized in both Forza and 365, as illustrated in the below figures:

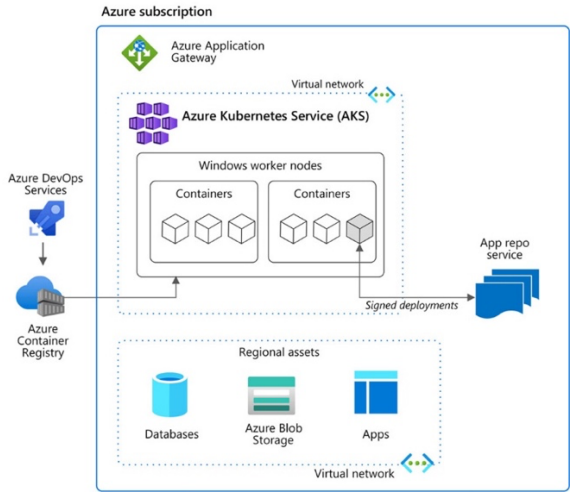


Figure 1 - Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS)

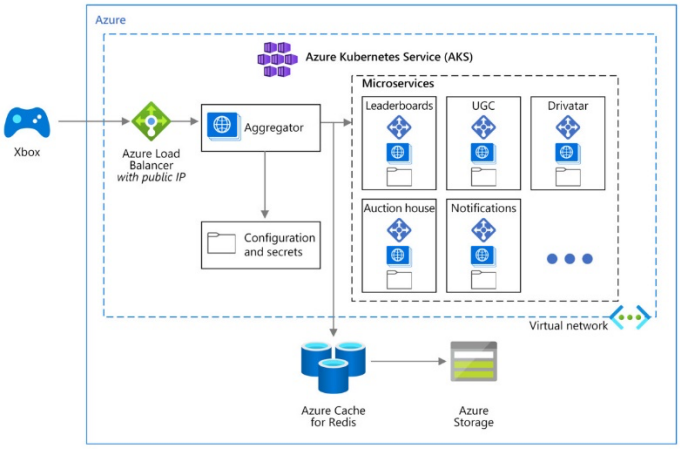


Figure 2 - Microsoft Customer Story-Forza Horizon 5 crosses the finish line fueled by Azure Kubernetes Service (AKS)

36. The basic building blocks of the managed distributed application architecture utilized by Forza and 365 are microservices deployed via containers. Forza and 365 each comprise microservices. Microsoft Customer Story-Forza Horizon 5 crosses the finish line fueled by Azure Kubernetes Service (AKS) ("Forza Horizon 5 is based on Windows and the .NET Framework and now features 17 core microservices that run in Windows containers hosted on AKS."); and Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on

Azure Kubernetes Service (AKS) ("Today [Microsoft 365 is] a mix of monolithic services and newer microservices."). "In a microservice-based architecture, the application is built as a collection of the services". Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 335, Microsoft, Edition v7.0 (2023). Containers are convenient for microservices, useful for monolithic applications, and are becoming the standard unit of deployment of server-based applications and/or services. *Id*. ("A Docker container is becoming the standard unit of deployment for any server-based application or service… Containers are convenient for microservices but can also be useful for monolithic applications based on the traditional .NET Framework, when using Windows Containers."). The primary purpose of containers is to package and run applications within the cloud. About Windows containers | Microsoft Learn ("Containers are a technology for packaging and running Windows and Linux applications across diverse environments on-premises and in the cloud."). Taking full advantage of the convenience and utility of the containers, Forza and 365 are deployed as Windows containers hosting microservices and monolithic applications. Microsoft Customer Story-Forza Horizon 5 crosses the finish line fueled by Azure Kubernetes Service (AKS) ("Forza Horizon 5 is based on Windows and the .NET Framework and now features 17 core microservices that run in Windows containers hosted on AKS."); and Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS) ("The architecture supporting Microsoft 365 runs highly scalable services in Windows containers hosted on AKS clusters.").

37. As shown in Figure 1-2, all the containers of Forza and 365 are within Azure Kubernetes Service (hereafter "AKS"). Kubernetes is a system for automating deployment, scaling, and management of containerized applications. Azure Kubernetes Service (AKS) simplifies deploying a managed Kubernetes cluster in Azure by offloading the operational

overhead to Azure. "Azure Kubernetes Service (AKS), a managed Kubernetes offering, further simplifies container-based application deployment and management." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn AKS, which is a managed service, utilizes Kubernetes to create a cluster architecture that provides a cloud infrastructure comprised of programmable devices powered by a sandboxing operating system and hosting containerized applications. The cluster architecture created by AKS is depicted below:

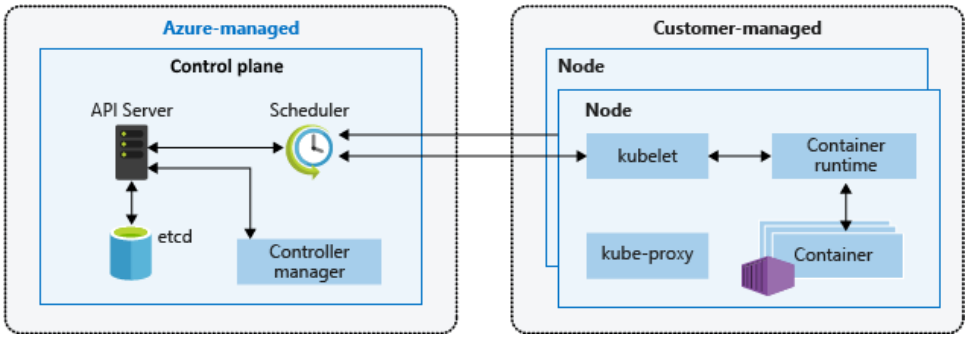


Figure 3 - Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn

38. As shown in Figure 3, the containers packing and running applications are placed with an AKS cluster. "An AKS cluster has at least one node, an Azure virtual machine (VM) that runs the Kubernetes node components and container runtime." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. "In contrast to containers, VMs run a complete operating system–including its own kernel". Containers vs. virtual machines | Microsoft Learn. Being virtual machines, the nodes provide the cloud infrastructure of the distributed application architecture and a base operating system.

39. In addition to providing the cloud infrastructure, nodes provide the platform for running containerized applications. "An AKS cluster has at least one node, an Azure virtual machine (VM) that runs the Kubernetes node components and container runtime." Azure

Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn

The container runtime "allows containerized applications to run and interact with additional resources, such as the virtual network or storage." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. For Windows Server node pools, such as those used by games like Forza and 365, "containers can be used in preview for the container runtime, but Docker is still the default container runtime." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. When run, the containerized application "shares the host operation system's kernel [but] the container doesn't get unfettered access to it. Instead, the container gets an isolated – and in some cases virtualized – view of the system." About Windows containers | Microsoft Learn. Furthermore, "any changes affect only the container and are discarded when it stops." About Windows containers | Microsoft Learn. A container, therefore, is "an isolated, lightweight package for running an application on the host operating system." About Windows containers | Microsoft Learn. Nodes are virtual machines with an OS and container runtime running isolated containerized applications; further, nodes are programmable devices powered by a sandboxing operating system facilitating the deployment of a plurality of network applications.

40. The containers utilized by games like Forza and 365 are not just randomly distributed within the cloud infrastructure. Rather, they are divided amongst network devices and cloud devices and interconnected via a virtual fabric, as shown in the below figure:

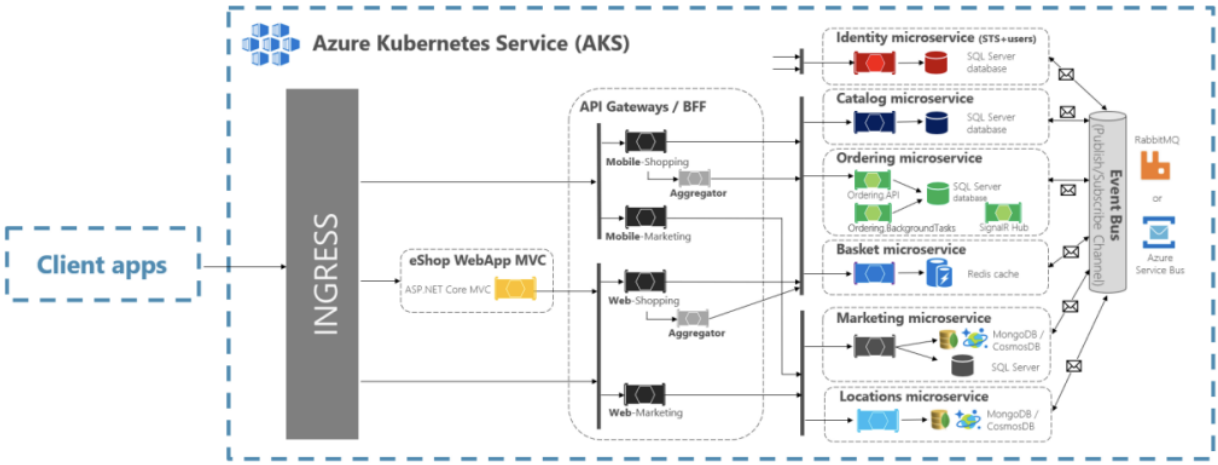


Figure 4 - Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, Figure 6-41, Microsoft, Edition v7.0 (2023)

41. As depicted in Figure 4, the architecture comprises containers divided amongst an API gateway and various microservices. A component of an API gateway is one or more aggregators. An aggregator is also shown in Figure 2 as a component of the architecture of Forza. The Forza "the front door for all client requests is the aggregator service, a single autoscaling deployment that runs multiple pods in AKS." Microsoft Customer Story-Forza Horizon 5 crosses the finish line fueled by Azure Kubernetes Service (AKS). Accordingly, Forza utilizes an API gateway comprised of multiple containers.

42. Though not shown in Figure 1, the managed distributed application architecture utilized for 365 also includes an API gateway. "Microsoft 365 represents a huge engineering effort." Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS). 365 is the result of multiple development teams writing millions of lines of code. Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS). ("Microsoft 365 represents the work of multiple development teams, writing millions of lines of code over the years."). Over the

years 365 has evolved into a mix of monolithic services and microservices uniting diverse products and services. Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS). ("The architecture uniting these products and services came together more than a decade ago. Today it's a mix of monolithic services and newer microservices."). Microsoft 365 is thus a very large and complex application.

43. Being so large and complex, 365 must include an API gateway acting as an intermediate level or tier of indirection between Azure Application Gateway (depicted in Figure 1) and the containers supporting 365's monolithic applications and microservices. Failing to have an API gateway would be acting against Microsoft's own advice. *See* Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 40, Microsoft, Edition v7.0 (2023) ("In production environments, you could have an Application Delivery Controller (ADC) like Azure Application Gateway between your microservices and the Internet… However, when you build large and complex microservice-based applications (for example, when handling dozens of microservice types), and especially when the client apps are remote mobile apps or SPA web applications, that approach faces a few issues."). Accordingly, on information and belief, 365 includes an API gateway.

44. Client applications, such as web apps, mobile apps, and desktop apps, connect to the API gateway and their requests are forwarded to the various containers. Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 43, Microsoft, Edition v7.0 (2023) ("Apps connect to a single endpoint, the API Gateway, that's configured to forward requests to individual microservices."); and Microsoft Customer Story-Forza Horizon 5 crosses the finish line fueled by Azure Kubernetes Service (AKS) ("The front door for all client requests is the aggregator service... The aggregator service interacts with the other microservices running

in containers."). In addition to forwarding requests, the API gateway also receives results from the various containers and then sends them back to the client app. Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 45, Microsoft, Edition v7.0 (2023) ("With this approach, the client app sends a single request to the API Gateway that dispatches several requests to the internal microservices and then aggregates the results and sends everything back to the client app."). Accordingly, the containers within the API gateway utilized by Microsoft's video games, like Forza, and 365 are running applications that accept requests form client apps, forward them to the appropriate microservice container, receive the results, and return the results to the client app. The API gateway is thus a programmable network device hosting a plurality of applications.

45. The backend containers, the microservices outside the API gateway, receive and process the requests. Accordingly, nodes running backend containers are cloud devices hosting a plurality of second network applications.

46. The communication between the containers of the API gateway and the backend containers provides a virtual fabric forming a distributed application.

47. In addition to programmable network devices and programmable cloud devices, the AKS clusters of the managed distributed application architecture of games like Forza and 365 include an application management portal managing usage of distributed applications. When the AKS clusters for Forza and 365 were created, "a control plane [was] automatically created and configured." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. The control plane contains various components that control workloads and enable scaling of distributed applications, such as a kube-scheduler and kube-controller-manager. When applications are created or scaled, "the Scheduler determines what nodes can run the

workload and starts them." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. "The Controller Manager oversees a number of smaller controllers that perform actions such as replicating pods and handling node operations." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. "While you don't need to configure components (like a highly available etcd store) with this managed control plane, you can't access the control plane directly. Kubernetes control plane and node upgrades are orchestrated through the Azure CLI or Azure portal."). Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. The control plane within AKS is thus an application management portal managing usage of distributed applications.

48. The distributed application comprising Forza "features 17 core microservices that run in Windows Containers hosted on AKS." Microsoft Customer Story-Forza Horizon 5 crosses the finish line fueled by Azure Kubernetes Service (AKS). Likewise, "the architecture supporting Microsoft 365 runs highly scalable services in Windows containers hosted on AKS clusters." Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS). Accordingly, both Forza and 365 utilize Windows containers, for which "Docker is still the default container runtime." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. "When using Docker, a developer creates an app or service and packages it and its dependencies into a container image." Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 7, Microsoft, Edition v7.0 (2023). A container image is "package with all the dependencies and information needed to create a container." *Id.* "Containers are a technology for packaging and running Windows and Linux applications." About Windows containers | Microsoft Learn. As an

image creates a container, which runs a portion of a distributed application, an image is a portion of the distributed application, and a collection of images forms a distributed application. As such, "to run the app or service, the app's image is instantiated to create a container." Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 7, Microsoft, Edition v7.0 (2023). "Developers should store images in a registry, which acts as a library of images and is needed when deploying to production orchestrators." *Id.* Accordingly both Forza and 365 include a container registry, such as Azure Container Registry (ACR), depicted in Figure 1, which "is a private registry for container images." Kubernetes on Azure tutorial - Create an Azure Container Registry and build images - Azure Kubernetes Service | Microsoft Learn. "Orchestrators enable you to manage their images, containers, and hosts through a command-line interface (CLI) or a graphical UI." Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 7, Microsoft, Edition v7.0 (2023). In AKS the control plane "provides the core Kubernetes services and orchestration of application workloads" and accessed via the Azure CLI or Azure portal. Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. "To run the app or service, the app's image is instantiated to create a container." Cesar de la Torre, et al., .NET Microservices: Architecture for Containerized .NET Applications, 7, Microsoft, Edition v7.0 (2023). As previously noted, the AKS control plane, accessible via Azure CLI, is an application management portal managing usage of distributed applications. In Azure CLI, the "kubectl apply" command is used to deploy a container from an image. Kubernetes on Azure tutorial - Deploy an application to Azure Kubernetes Service (AKS) - Azure Kubernetes Service | Microsoft Learn. As both Forza and 365 utilize AKS and Windows Containers, and include a container registry, they each include an application repository storing distributed applications that is coupled to an application management portal.

49. To ensure the images within the repository are trustworthy, Microsoft recommends using "Notary V2 to attach signatures to your images to ensure deployments are coming from a trusted location." Concepts - Security in Azure Kubernetes Services (AKS) - Azure Kubernetes Service | Microsoft Learn. There is no reason to believe Microsoft does not follow its own advice. When implementing Notary, "[a]rtifacts are signed (notation sign) with private keys, and validated with public keys (notation verify)." notaryproject/README.md at main · JeyJeyGao/notaryproject (github.com). The "notations verify" command utilized to verify a signature may be added to any container runtime. notaryproject/README.md at main · JeyJeyGao/notaryproject (github.com) ("**Note:** the notation verify command is shown external to the docker client to demonstrate validation, which may be added to any container runtime, including containerd."). For Windows Node pools, such as those used by games like Forza and 365, "containerd can be used in preview for the container runtime, but Docker is still the default container runtime." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. In fact, Figure 1 depicts the use of "signed deployments" within 365 at the container level, which would be within the runtime. As each the network device comprising the nodes forming the API gateway and the cloud device comprising the nodes running the backend containers each have a container runtime, in both Forza and 365 the network and cloud device verify the authenticity of upgrades.

50. In combination or the alternative, signatures could be verified at the control plane or by the runtimes of the nodes at the direction of the control plane. Notary V2 enables a scenario in which a "system builds the artifacts, signs them, and pushes to a registry. The production system pulls the artifacts, verifies the signatures and runs them." specifications/requirements/scenarios.md at main · notaryproject/specifications (github.com). In this scenario, "The orchestrator verifies the

artifacts are signed by a set of specifically trusted keys. Unsigned artifacts, or artifacts signed by non-trusted keys are rejected." specifications/requirements/scenarios.md at main · notaryproject/specifications (github.com). Within AKS, the control plane "provides the core Kubernetes services and orchestration of application workloads." Azure Kubernetes Services (AKS) Core Basic Concepts - Azure Kubernetes Service | Microsoft Learn. The control plane is thus the orchestrator in AKS which would verify the signature on a container image. As noted above, the control plane within AKS is an application management. Accordingly, in both Forza and 365 the application management portal of the control plane verifies the authenticity of upgrades based on security keys.

51. One of the advantages of utilizing containers and AKS clusters is the availability of blue-green deployments. *See* Blue-Green Deployment in Azure Container Apps | Microsoft Learn; and Blue-green deployment of AKS clusters - Azure Architecture Center | Microsoft Learn. "In a blue-green deployment, two identical environments, referred to as "blue" and "green," are set up. One environment (blue) is running the current application version and one environment (green) is running the new application version." Blue-Green Deployment in Azure Container Apps | Microsoft Learn. "This allows you to test the blue version in production while only exposing users to the green, stable version. Once tested, the blue version gradually replaces the green version." Kubernetes Deployment – Strategies & Tools | Microsoft Azure. Another method managing upgrades with substantially no interruption to operation is canary deployment in which a portion of users test the new upgrades. When utilizing this strategy, the managed distributed application architecture will "run one ReplicaSet of the new version along with the current version and then, after a specified period of time without errors, scale up the new version as you remove the old version." Kubernetes Deployment – Strategies & Tools | Microsoft Azure. A modification

of canary deployment is utilized by 365 to manage new releases. "Using deployment rings, teams have an automated, consistent process that speeds up releases." Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS). When utilizing ring deployment will "gradually deploy and validate changes to your extension in production, while limiting the affect on your users." Progressively expose your releases using deployment rings - Azure DevOps | Microsoft Learn. To facilitate minimum disruption, "users get divided into three groups in production: Canaries: voluntarily test features as soon as they're available. Early adopters: voluntarily preview releases, considered more refined than the canary bits. Users: consume the products, after they've passed through canaries and early adopters." Progressively expose your releases using deployment rings - Azure DevOps | Microsoft Learn. "The new release management workflow starts when application developers make a change to their code. Using Azure Pipelines or Docker, they package their services as a container and push the image to Azure Container Registry." Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS). "The new build triggers a continuous deployment trigger, which automatically starts the Canaries environment deployment… Only the Canaries are affected by the change." Progressively expose your releases using deployment rings - Azure DevOps | Microsoft Learn.

52. Limiting the change to Canaries could be accomplished at either the Azure Application Gateway or the API gateway by directing the Canaries to different containers (i.e. ReplicaSet). In both instances new containers would be deployed by the AKS control panel operating as an orchestrator. Implementing an upgrade to games like Forza or 365 entails deploying new containers utilizing images for the upgrade that have been stored in the Azure Container Registry. "The automated CI/CD pipeline deploys the containers to test clusters on

AKS, where any issues can be detected and fixed. Only after a release passes all validation checks can it be pushed to the early ring and previewed before graduating to the production environments." Microsoft Customer Story-Modernizing Microsoft 365 with a move to Windows containers on Azure Kubernetes Service (AKS). As Canaries "voluntarily test features as soon as they're available", the first production environment would be the Canaries environment deployment. Progressively expose your releases using deployment rings - Azure DevOps | Microsoft Learn. As there will be running "one ReplicaSet of the new version along with the current version" (Kubernetes Deployment – Strategies & Tools | Microsoft Azure), the containers comprising the Canaries environment will coexist on the programmable cloud devices along with the old containers. For these containers to be utilized by Canaries the requests coming from Canaries must be forwarded to the Canary containers. Such forwarding could be accomplished by configuring Azure Application Gateway to forward the incoming requests from Canaries directly to the containers comprising the Canaries environment deployment. Alternatively, incoming requests from Canaries could be directed to new Canary containers within the API gateway on the programmable network device that would then forward them to the appropriate Canary containers.

53. As only Canaries are using containers from the upgrade, any disruptions or errors would only affect Canaries, thereby providing substantially no interruption to operation. Deploying new Canary containers for the upgrade would be accomplished using the "kubectl apply" command to deploy a container from an image for the upgrade from the Azure Container Registry. Kubernetes on Azure tutorial - Deploy an application to Azure Kubernetes Service (AKS) - Azure Kubernetes Service | Microsoft Learn. As previously noted, the AKS control plane, accessible via Azure CLI, is an application management portal. Accordingly, both Forza and 365 have an application management port capable of managing upgrades with substantially no

interruption to operation, by virtual of the Guest OS and container runtime of the API gateway nodes of the programmable network device and the backend node of the programmable cloud device providing a sandboxing operating system facilitating deployment isolated and coexisting containers.

54. As the foregoing shows, games like Forza and 365 each incorporate a managed distributed application architecture.

55. Edge Networking has at all times complied with the marking provisions of 35 U.S.C. § 287 with respect to the Patents-in-Suit. On information and belief, any prior assignees and licensees have also either complied with the marking provisions of 35 U.S.C. § 287, or else were excused from the obligation to mark for the reason that § 287 does not apply.

CLAIM 1

(Infringement of the 871 Patent)

56. Edge Networking repeats and realleges all the preceding paragraphs, as if set forth herein.

57. Edge Networking has not licensed or otherwise authorized Microsoft to make, use, offer for sale, sell, or important products that embody the 871 Patent.

58. Microsoft infringes at least claims 1 and 9 of the 871 Patent in violation of 35 U.S.C. § 271 with respect to the accused products. Edge Networking contends each limitation is met literally, and, to the extent a limitation is not met literally, is met under the doctrine of equivalents.

59. For example, Microsoft directly infringes at least claims 1 and 9 of the 871 Patent by making, using, selling access to, and/or offering to access to within the United States Forza and 365.

60. As described supra, the Forza and 365 are built upon a managed distributed application architecture comprising a programmable network device hosting a plurality of applications. Forza and 365 each have an API gateway comprising containerized applications that accept requests form client apps, forward them to the appropriate microservice container, receive the results, and return the results to the client app. By accepting and forwarding requests and receiving and returning results, the applications within the API gateway are processing data flows. The containerized applications of the API gateways are running on nodes, which are virtual machines. Accordingly, the API gateway in both Forza and 365 are thus programmable network devices hosting a plurality of network applications.

61. Additionally, Forza and 365 each comprise a programmable could device hosting a plurality of second network applications. As noted supra, both Forza and 365 contain backend containers running on nodes outside of the API gateway. These nodes are devices running a plurality of second network applications as containerized microservices.

62. A virtual fabric forms a distributed application via secure communication between the containerized applications of the API gateway nodes and containerized microservices of the backend nodes. The containers within the API gateway utilized by Forza and 365 are running applications that accept requests form client apps, forward them to the appropriate microservice container, receive the results, and return the results to the client app. Accordingly, the communication between the containers of the API gateway and the backend containers provides a virtual fabric forming a distributed application. As such, in both Forza and 365, at least one of the plurality of first network applications in the programmable network device of the API gateway and least one of the plurality of second network applications in the programmable cloud device backend nodes are in secure communication with each other to form a distributed application.

63. In both Forza and 365 a sandboxing operating system facilitates deployment of the API gateway applications and the backend microservices. The backend microservices and API gateway applications are containerized applications running on nodes. Each node is a virtual machine with an OS and a container runtime running the applications as isolated containerized applications. Accordingly, the programmable network device of the API gateway and the programmable cloud device having the backend nodes are each powered by a sandboxing operating system which facilitates deployment of a plurality of first and second network comprising applications of the API gateway and microservices within the backend nodes, as detailed supra.

64. As also noted supra, both Forza and 365 have an application management portal coupled to the programmable network device and programmable cloud device. When the AKS clusters were created for both Forza and 365 a control plane was automatically created and configured that is accessed through the Azure CLI or Azure portal. The control plane schedules nodes, replicates the pods, handles node operations, and otherwise manages applications. As API gateway of the programmable network device and the backend nodes of the programmable cloud device each comprise nodes running containerized applications wrapped in pods, for these operations to be carried out the control panel must be coupled to the network device of the API gateway and the cloud device of the backend nodes. Accordingly, both Forza and 365 have an application management portal coupled to the programmable network device and programmable cloud device. As further detailed above, the control plane enables blue-green deployments and thus is capable of managing upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device.

65. For the reasons noted supra, the control plane of the application management portal utilized in both Forza and 365 is capable of utilizing Notary V2. When implementing Notary, artifacts are signed with private keys, and validated with public keys. The verification of signatures, as noted supra, may be performed by the container runtime running on each API gateway node of the network device and each backend node of the cloud device at the direction of the control plane or otherwise. Accordingly, in both Forza and 365 the network device comprising the API gateway nodes and the cloud device comprising the backend nodes respectively verify authenticity of upgrades to the network applications within the API gateway and the second network applications within the backend nodes of the cloud devices based on unique security keys associated with the first and second network applications.

66. As both Forza and 365 utilize AKS and Windows Containers, and include a container registry, they each include an application repository that is coupled to the control plane of the application management portal and that is capable of storing distributed applications for installation in the API gateway of the programmable network device and the backend nodes of the programmable cloud device.

67. In both Forza and 365, the control plane of the application management portal schedules nodes, replicates the pods, handles node operations, and otherwise manages applications. As such, both Forza and 365 have an application management portal managing provisioning, usage and de-provisioning of the distributed applications on the API gateway of the programmable network device and the backend nodes of the programmable cloud device.

68. Edge Networking Inc. has been injured and seeks damages to adequately compensate it for Microsoft's infringement of the 871 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

69. Upon information and belief, Microsoft will continue to infringe the 871 Patent unless permanently enjoined by the Court. Pursuant to 35 U.S.C. § 283, Edge Networking is entitled to a permanent injection against further infringement of the 871 Patent by Microsoft.

CLAIM 2

(Infringement of the 095 Patent)

70. Edge Networking repeats and realleges all the preceding paragraphs, as if set forth herein.

71. Edge Networking has not licensed or otherwise authorized Microsoft to make, use, offer for sale, sell, or important products that embody the 095 Patent.

72. Microsoft infringes at least claims 1 and 15 of the 095 Patent in violation of 35 U.S.C. § 271 with respect to the accused products. Edge Networking contends each limitation is met literally, and, to the extent a limitation is not met literally, is met under the doctrine of equivalents.

73. For example, Microsoft directly infringes at least claims 1 and 15 of the 905 Patent by making, using, selling access to, and/or offering to access to within the United States Forza and 365.

74. As described supra, the Forza and 365 are built upon a managed distributed application architecture comprising a programmable network device hosting a plurality of applications. Forza and 365 each have an API gateway comprising containerized applications that accept requests form client apps, forward them to the appropriate microservice container, receive the results, and return the results to the client app. By accepting and forwarding requests and receiving and returning results, the applications within the API gateway are processing data flows. The containerized applications of the API gateways are running on nodes, which are virtual

machines. Accordingly, the API gateway in both Forza and 365 are thus programmable network devices hosting a plurality of network applications.

75. Additionally, Forza and 365 each comprise a programmable could device hosting a plurality of second network applications. As noted supra, both Forza and 365 contain backend containers running on nodes outside of the API gateway. These nodes are devices running a plurality of second network applications as containerized microservices. Thus, Forza and 365 have programmable cloud devices having plurality of virtual machines wherein at least one of the virtual machines host a least one of a plurality of second network applications comprising containerized microservices.

76. A virtual fabric forms a distributed application via secure communication between the containerized applications of the API gateway nodes and containerized microservices of the backend nodes. The containers within the API gateway utilized by Forza and 365 are running applications that accept requests form client apps, forward them to the appropriate microservice container, receive the results, and return the results to the client app. Accordingly, the communication between the containers of the API gateway and the backend containers provides a virtual fabric forming a distributed application. As such, in both Forza and 365, at least one of the plurality of first network applications in the programmable network device of the API gateway and least one of the plurality of second network applications in the programmable cloud device backend nodes are in secure communication with each other to form a distributed application.

77. In both Forza and 365 a sandboxing operating system facilitates deployment of the API gateway applications and the backend microservices. The backend microservices and API gateway applications are containerized applications running on nodes. Each node is a virtual machine with an OS and a container runtime running the applications as isolated containerized

applications. Accordingly, the programmable network device of the API gateway and the programmable cloud device having the backend nodes are each powered by a sandboxing operating system which facilitates deployment of a plurality of first and second network comprising applications of the API gateway and microservices within the backend nodes, as detailed supra.

78. As also noted supra, both Forza and 365 have an application management portal coupled to the programmable network device and programmable cloud device. When the AKS clusters were created for both Forza and 365 a control plane was automatically created and configured that is accessed through the Azure CLI or Azure portal. The control plane schedules nodes, replicates the pods, handles node operations, and otherwise manages applications. As API gateway of the programmable network device and the backend nodes of the programmable cloud device each comprise nodes running containerized applications wrapped in pods, for these operations to be carried out the control panel must be coupled to the network device of the API gateway and the cloud device of the backend nodes. Accordingly, both Forza and 365 have an application management portal coupled to the programmable network device and programmable cloud device. As further detailed above, the control plane enables blue-green deployments and thus is capable of managing upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device by virtual of the Guest OS and container runtime of the API gateway nodes of the programmable network device and the backend node of the programmable cloud device providing a sandboxing operating system facilitating deployment isolated and coexisting containers.

79. For the reasons noted supra, the control plane of the application management portal utilized in both Forza and 365 is capable of utilizing Notary V2. When implementing Notary,

artifacts are signed with private keys, and validated with public keys. The verification of signatures may occur by the control plane of the application management portal. Accordingly, in both Forza and 365 the control plane of the application management portal verifies authenticity of upgrades to the network applications within the API gateway and the second network applications within the backend nodes of the cloud devices based on unique security keys associated with the first and second network applications.

80. As both Forza and 365 utilize AKS and Windows Containers, and include a container registry, they each include an application repository that is coupled to the control plane of the application management portal and that is capable of storing distributed applications for installation in the API gateway of the programmable network device and the backend nodes of the programmable cloud device. As the applications would have gone through deployment rings, they would have been tested for installation in the programmable network device and programmable cloud device.

81. In both Forza and 365, the control plane of the application management portal schedules nodes, replicates the pods, handles node operations, and otherwise manages applications. As such, both Forza and 365 have an application management portal managing provisioning, usage and de-provisioning of the distributed applications on the API gateway of the programmable network device and the backend nodes of the programmable cloud device.

82. Edge Networking has been injured and seeks damages to adequately compensate it for Microsoft's infringement of the 095 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

83. Upon information and belief, Microsoft will continue to infringe the 095 Patent unless permanently enjoined by the Court. Pursuant to 35 U.S.C. § 283, Edge Networking is entitled to a permanent injection against further infringement of the 095 Patent by Microsoft.

CLAIM 3

(Infringement of the 823 Patent)

84. Edge Networking repeats and realleges all the preceding paragraphs, as if set forth herein.

85. Edge Networking has not licensed or otherwise authorized Microsoft to make, use, offer for sale, sell, or important products that embody the 823 Patent.

86. Microsoft infringes at least claims 1 and 19 of the 823 Patent in violation of 35 U.S.C. § 271 with respect to the accused products. Edge Networking contends each limitation is met literally, and, to the extent a limitation is not met literally, is met under the doctrine of equivalents.

87. For example, Microsoft directly infringes at least claims 1 and 19 of the 823 Patent by making, using, selling access to, and/or offering to access to within the United States Forza and 365.

88. As described supra, Forza and 365 are built upon a managed distributed application architecture comprising a programmable network device hosting a plurality of applications. Forza and 365 each have an API gateway comprising containerized applications that accept requests form client apps, forward them to the appropriate microservice container, receive the results, and return the results to the client app. By accepting and forwarding requests and receiving and returning results, the applications within the API gateway are processing data flows. The containerized applications of the API gateways are running on nodes, which are virtual machines. Accordingly,

the API gateway in both Forza and 365 are programmable network devices adapted to host a plurality of network applications.

89. Additionally, Forza and 365 each comprise a programmable could device hosting a plurality of second network applications. As noted supra, both Forza and 365 contain backend containers running on nodes outside of the API gateway. These nodes are devices running a plurality of second network applications as containerized microservices. Thus, Forza and 365 have programmable cloud devices adapted to host a plurality of cloud applications.

90. A virtual fabric forms a distributed application via secure communication between the containerized applications of the API gateway nodes and containerized microservices of the backend nodes. The containers within the API gateway utilized by Forza and 365 are running applications that accept requests form client apps, forward them to the appropriate microservice container, receive the results, and return the results to the client app. Accordingly, the communication between the containers of the API gateway and the backend containers provides a virtual fabric forming a distributed application. As such, in both Forza and 365, a plurality of first network applications in the programmable network device of the API gateway and a of plurality of cloud applications in the programmable cloud device backend nodes are in secure communication with each other to form a distributed application.

91. As also noted supra, both Forza and 365 have an application management portal coupled to the programmable network device and programmable cloud device. When the AKS clusters were created for both Forza and 365 a control plane was automatically created and configured that is accessed through the Azure CLI or Azure portal. The control plane schedules nodes, replicates the pods, handles node operations, and otherwise manages applications. As API gateway of the programmable network device and the backend nodes of the programmable cloud

device each comprise nodes running containerized applications wrapped in pods, for these operations to be carried out the control panel must be coupled to the network device of the API gateway and the cloud device of the backend nodes. Accordingly, both Forza and 365 have an application management portal coupled to the programmable network device and programmable cloud device. As further detailed above, the control plane enables blue-green deployments and thus is capable of managing upgrades of the first and second network applications with substantially no interruption to operation of the programmable network device and programmable cloud device by virtual of the Guest OS and container runtime of the API gateway nodes of the programmable network device and the backend node of the programmable cloud device providing a sandboxing operating system facilitating deployment isolated and coexisting containers. By virtual of being to change operations for subset of user by deploying isolated containers, the network device applications and the cloud device application device form unified capabilities enabling a plurality of upper layer application programing interfaces to program the plurality of network device applications and the plurality of cloud application independent of network device hardware and cloud device hardware.

92. In both Forza and 365, the control plane of the application management portal schedules nodes, replicates the pods, handles node operations, and otherwise manages applications. As such, both Forza and 365 have an application management portal managing life cycles of the distributed applications on the API gateway of the programmable network device and the backend nodes of the programmable cloud device.

93. As both Forza and 365 utilize AKS and Windows Containers, and include a container registry, they each include an infrastructure application marketplace that is capable of providing distributed applications to the application management portal for installation in the API

gateway of the programmable network device and the backend nodes of the programmable cloud device. As the applications would have gone through deployment rings, they would have been tested for installation in the programmable network device and programmable cloud device.

94. Edge Networking. has been injured and seeks damages to adequately compensate it for Microsoft's infringement of the 823 Patent. Such damages should be no less than a reasonable royalty under 35 U.S.C. § 284.

95. Upon information and belief, Microsoft will continue to infringe the 823 Patent unless permanently enjoined by the Court. Pursuant to 35 U.S.C. § 283, Edge Networking is entitled to a permanent injection against further infringement of the 823 Patent by Microsoft.

DEMAND FOR JURY TRIAL

Plaintiff hereby requests a jury trial of all issues so triable.

PRAYER FOR RELIEF

WHEREFORE, Plaintiff prays for relief against Defendants as follows:

a. Entry of judgment declaring that Defendant infringes one or more claims of each of the Patents-in-Suit;

b. An order awarding damages sufficient to compensate Plaintiff for Defendant's infringement of the Patents-in-Suit, but in no event less than a reasonable royalty, including supplemental damages post-verdict, together with pre-judgment and post-judgment interest and costs;

c. Enhanced damages pursuant to 35 U.S.C. § 284;

d. Entry of judgment declaring that this case is exceptional and awarding Plaintiff its costs and reasonable attorney fees pursuant to 35 U.S.C. § 285;

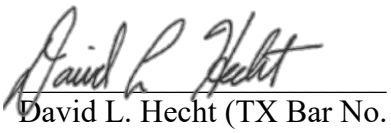
e. An accounting for acts of infringement;

f. Such other equitable relief which may be requested and to which the Plaintiff is entitled; and

g. Such other and further relief as the Court deems just and proper.

Dated: February 29, 2024

HECHT PARTNERS LLP



David L. Hecht (TX Bar No. 24136516)
111 Congress Avenue, Suite 500
Austin, TX 78701
Tel: 212-851-6821
Email: dhecht@hechtpartners.com

Maxim Price (*admission forthcoming*)
Yi Wen Wu (*admission forthcoming*)
125 Park Avenue, 25th Floor
New York, NY 10017
Tel: 212-851-6821
Email: mprice@hechtpartners.com
wwu@hechtpartners.com