

**IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION**

Sandpiper CDN, LLC,

Plaintiff,

v.

Microsoft Corporation

Defendant.

Civil Case No. 2:25-cv-00664-JRG

MICROSOFT'S INVALIDITY CONTENTIONS

Pursuant to the Rules of Practice for Patent Cases before the Eastern District of Texas, Microsoft Corp. (“Microsoft” or “Defendant”) provides these Invalidity Contentions to Sandpiper CDN, LLC (“Sandpiper” or “Plaintiff”) for the claims (collectively, “Asserted Claims”) identified as asserted in Plaintiff Sandpiper’s Disclosure of Asserted Claims and Infringement Contentions served on September 16, 2025 (“Infringement Contentions”).

Defendant’s Invalidity Contentions include the accompanying claim charts concerning U.S. Patent Nos. 8,478,903 (the “’903 Patent”); 8,924,466 (the “’466 Patent”); 9,456,053 (the “’053 Patent”); 9,722,883 (the “’883 Patent”); 9,762,692 (the “’692 Patent”); 10,701,173 (the “’173 Patent”) (collectively, the “Asserted Patents”)

I. INTRODUCTION

Plaintiff asserts the following patents and claims:

Patent	Asserted Claims	Sandpiper’s Alleged Priority Date	Chart Prefix
’903 Patent	28, 29, 30, 31, 32, 34, 36, 37, 38, and 39	February 10, 1998	A
’466 Patent	1–5 and 7–11	February 14, 2002	B
’053 Patent	27, 28, 29, 31, 33–35, 37–42, 44–50	December 14, 2011	C
’883 Patent	1, 2, 5–9, 11, 14, 15, 17–21, and 23	December 13, 2012	D
’692 Patent	1, 2, 4, 5, 7, 8, 15, 17, and 18	April 4, 2008	E
’173 Patent	1–7	December 15, 2014	F

The claims identified above are referenced as “the Asserted Claims” unless reference is made specifically with respect to particularly identified one or more of the Asserted Patents and/or one or more of the Asserted Claims. By identifying Sandpiper’s alleged priority date for the Asserted Claims of each patent above, Microsoft does not concede that Sandpiper’s claims of priority are correct. Further, the citation of prior art herein and the accompanying exhibits are not

intended to reflect Defendant's claim construction contentions, which will be disclosed in due course in accordance with the Scheduling Order (Dkt. No. 32). Instead, Microsoft's Invalidity Contentions as set forth herein and in the accompanying exhibits may reflect Plaintiff's apparent proposed claim constructions based on its Infringement Contentions.

As further detailed in and supported by these Invalidity Contentions, Defendant contends that each of the Asserted Claims is invalid under at least 35 U.S.C. §§ 101, 102, 103, and/or 112.¹ Defendant reserves the right to prove the invalidity of the Asserted Claims on bases other than those required to be disclosed in these disclosures pursuant to P.R. 3-3.

II. RESERVATION OF RIGHTS

Based on this disclosure of the Asserted Claims, Microsoft does not provide any contentions regarding any claims not asserted in Sandpiper's Infringement Contentions. To the extent that the Court permits Sandpiper to assert additional claims against Microsoft to amend its infringement contentions and/or assert one or more claims other than the Asserted Claims, Microsoft reserves the right to further modify, amend, or supplement these invalidity contentions accordingly—for example, to show the invalidity and/or ineligibility of any such newly Asserted Claims.

This Court's Rules of Practice for Patent Cases before the Eastern District of Texas requires that these invalidity contentions be served after Sandpiper's Infringement Contentions, which must be "[a] chart identifying where specifically in each alleged item of prior art each element of each asserted claim is found, including for each element that such party contends is governed by 35 U.S.C. § 112 ¶ 6, the identity of the structure(s), act(s), or material(s) in each item of prior art that

¹ References to Title 35 of the United States Code are to statutes prior to amendment under the America Invents Act ("AIA"), to the extent that the Asserted Patents have effective filing dates prior to applicable AIA effective dates, and to the current statutory provisions with respect to Asserted Patents having effective filing dates after the applicable AIA effective dates.

performs the claimed function.” P.R. 3-3(c). Sandpiper’s infringement contentions fail to put Microsoft on notice of Sandpiper’s theories for why Microsoft’s products are purportedly infringing, or how Sandpiper interprets the elements of its asserted patent claims as covering Microsoft’s products.

These Invalidity Contentions do not constitute an admission that any accused products infringe the Asserted Patents. In other words, these Invalidity Contentions should not be deemed to be related to the non-infringement positions that Microsoft may advance in this action. In particular, Microsoft’s Invalidity Contentions are made in addition to, and/or in the alternative to, Microsoft’s non-infringement positions, and should not be interpreted to rely upon, or in any way affect, the non-infringement arguments Microsoft makes in this case.

Microsoft provides these disclosures consistent with the schedule currently in place, but do so without waiving any right to receive from Sandpiper such full and complete specific infringement disclosures as should have been provided within Sandpiper’s Infringement Contentions. Microsoft’s compliance with the current schedule should not be viewed as a waiver of any right to seek relief regarding the deficiencies in Sandpiper’s Infringement Contentions or to further amend or supplement its Invalidity Contentions once Sandpiper complies with the Court’s requirements.

Microsoft further reserves the right to revise these Invalidity Contentions in view of the Court’s construction of terms and phrases recited in one or more of the Asserted Claims, additional information obtained during discovery, additional infringement theories put forth by Plaintiff during fact and/or expert discovery, any findings as to the priority date(s) of the Asserted Claims, and/or positions that Plaintiff, its fact witnesses, or its expert witness(es) may take concerning claim construction, infringement, and/or invalidity issues, as permitted by the Court. Microsoft

further reserves the right to supplement their accompanying document productions should they later discover additional prior art documents, information, testimony, prior art systems and related documentation, and/or software or hardware code, including information provided by third parties after the date of service of these Invalidity Contentions.

Additionally, Plaintiff has not identified its contentions regarding the level of ordinary skill in the art for any of the Asserted Patents. Thus, Microsoft has based its Invalidity Contentions on the level of skill in the art apparent from the face of the asserted patents. For the avoidance of doubt, Microsoft contends that the asserted claims are invalid for the reasons set forth in their Invalidity Contentions under any plausible definition of the level of ordinary skill for each of the Asserted Patents. Nevertheless, Microsoft reserves the right to revise these Invalidity Contentions in response to any position taken by Plaintiff on the level of ordinary skill in the art applicable to a given Asserted Patent.

Microsoft may further rely on inventor admissions concerning the scope or state of the prior art relevant to the Asserted Claims, the patent prosecution histories of the Asserted Patents, related patents and/or patent applications, any deposition or trial testimony of the named inventors on the Asserted Patents, and the papers filed and any evidence produced or submitted by Plaintiff in connection with these cases or other related litigation. Microsoft reserves the right to contend that one or more of the Asserted Claims are invalid under 35 U.S.C. § 102(f) in the event Microsoft obtains evidence that one or more of the named inventors did not invent the subject matter in the Asserted Claims.

Prior art not included in these Invalidity Contentions, whether known or not known to Microsoft, may become relevant. In particular, Microsoft is currently unaware of the extent, if any, to which Plaintiff will contend that limitations of the Asserted Claims are not disclosed in the

prior art identified in these Invalidity Contentions. Accordingly, Microsoft reserves the right to identify other references that would disclose the allegedly missing limitation(s) of the claimed method, device, or system.

The references identified in these Invalidity Contentions, which include the attached claim charts, may disclose the elements of the Asserted Claims explicitly and/or inherently, render those elements alone or in combination obvious to those of ordinary skill in the art even absent an explicit or inherent disclosure, and/or they may be relied upon to show the state of the art in the relevant time frame. References identified in these Invalidity Contentions, as well as the “References Cited” on the face of the Asserted Patents and the patents cited within the body of the Asserted Patents, may be used to illustrate, but not limit the scope of, the state of the art to which the Asserted Patents pertain (i.e., at a time prior to the date of alleged inventions of the Asserted Claims of the Asserted Patents). Moreover, Microsoft reserves the ability to rely on later-identified sources of information, including but not limited to witness testimony and other discovery, to establish the state of the art in the relevant time frame pertaining to the Asserted Patents.

Because discovery is ongoing, Microsoft anticipates that additional prior art and invalidity bases may be found. Microsoft’s investigation and analysis of the prior art is continuing, and thus Microsoft reserves the right to supplement, amend, and/or revise the information provided herein as Microsoft conducts further investigation and/or analysis, including identifying, charting, and relying on additional references.

Additionally, in view of likely third-party discovery that will be taken, Microsoft reserves the right to present additional items of prior art under 35 U.S.C. §§ 102(a), (b), (e), and/or (g) and/or § 103 located during discovery or further investigation, and to assert contentions of invalidity under 35 U.S.C. §§ 102(c), (d), or (f). For example, Microsoft expects to issue

subpoenas to third parties believed to have knowledge, documents, and/or other evidence concerning invalidity of one or more of the Asserted Claims.

In addition to the positions and prior art identified in these Invalidity Contentions (including the accompanying claim charts), Microsoft also incorporates by reference all invalidity contentions and subject matter eligibility contentions, prior art, and invalidity claim charts (including, without limitation, all anticipation positions, obviousness positions [including all prior art combinations and motivations to combine], indefiniteness positions, written description positions, and non-enablement positions) concerning one or more of the Asserted Patents, as disclosed at any time. This includes without limitation disclosures in previous or related litigation, in United States Patent & Trademark Office (“USPTO”) proceedings, by the Plaintiff, by any other parties accused of patent infringement by the Plaintiff, or by the named inventors or any individuals associated with the prosecution of the Asserted Patents.

Specifically, for example, Microsoft identifies, as prior art upon which they may rely to show the invalidity of the Asserted Claims, the prior art references disclosed by parties in any other litigation involving one or more of the Asserted Patents, any patent related to any of the Asserted Patents, and/or any other patent allegedly assigned to Plaintiff claiming priority to the Asserted Patents (collectively, “Other Sandpiper Proceedings”). To the extent Microsoft has stipulated or stipulates in the future that it will not pursue certain invalidity grounds in these actions concerning patent claims subject to instituted *inter partes* review proceedings, Microsoft will abide by those stipulation(s), and nothing in these contentions should be read as suggesting otherwise. Rather, these contentions are intended to put Plaintiff on notice of the grounds of invalidity that Microsoft may assert in these actions, including because the conditions of any relevant stipulation are not

met (*e.g.*, if the corresponding *inter partes* review is not instituted) or because the grounds fall outside the scope of such stipulation(s).

Plaintiff has a present and ongoing duty to produce to Microsoft all relevant documents from any other Sandpiper Proceedings, including but not limited to all prior art, invalidity contentions, and expert reports on invalidity (among other relevant items).

Microsoft reserves the right to supplement or otherwise amend these Invalidity Contentions in response to any relevant discovery provided by third parties, Plaintiff, opening or rebuttal expert reports, fact or expert depositions, or in response to any claim construction ruling(s) issued by this Court (regardless of how and when such ruling is made). Microsoft also reserves the right to supplement or otherwise amend these Invalidity Contentions in response to any rebuttal evidence disclosed by Plaintiff or as otherwise may be necessary or appropriate under the circumstances.

III. BACKGROUND OF THE TECHNOLOGY

The references discussed in these Invalidity Contentions and the attached Exhibits may be relied upon to show the state of the art in the relevant time frame. These references are only exemplary and are not in any way intended to limit the scope of what a person of ordinary skill in the art (“POSITA”) would have understood at the relevant time period of the alleged inventions or the breadth of the state of the art to which the alleged inventions of the Asserted Patents relate.

A. CDN History

At least under Sandpiper’s apparent infringement theory, content delivery networks (“CDNs”) were well-known by February 10, 1998 (*i.e.*, the earliest alleged priority date of any of the Asserted Patents). The delivery of multimedia content over the internet grew throughout the 1990s. Accordingly, industry and academia had introduced architectures for replicating and caching web content across distributed servers to improve performance and handle load spikes.

1. Harvest

The Harvest project was an effort in the mid-1990s to make Internet information retrieval more scalable. Harvest introduced “a system that provides a scalable, customizable architecture for gathering, indexing, caching, replicating, and accessing Internet information.” *Bowman*, The Harvest Information Discovery and Access System at Abstract. In other words, Harvest provided a suite of tools that could collect information from diverse web sources, build searchable indexes, and cache objects to reduce network load. Its architecture was modular: Gatherers harvested content summaries from remote sites; Brokers built indexes of those summaries; a Replication service allowed mirroring of indexes; and importantly, an Object Cache stored retrieved documents for reuse. *Id.* at 6 (“To meet ever increasing demand on network links and information servers, Harvest include[d] a hierarchical Object Cache.”).

By early 1996, the Harvest Object Cache had been in active use for about two years at roughly 100 sites on the Internet. These included universities, research labs, and Internet service providers (“ISPs”) that installed Harvest to cache web objects. An April 1996 article by one of Harvest’s creators, Peter Danzig, reported that Harvest’s cache yielded an “order-of-magnitude performance improvement” over previous web proxies (specifically outperforming the CERN httpd cache).

https://www.usenix.org/legacy/publications/library/proceedings/sd96/full_papers/danzig-html/subsection3_3_1.html#:~:text=These%20figures%20show%20that%20the,minutes%20of%20the%20failed%20request. The cache could be configured either as a forward proxy (serving a group of users by caching external web content) or as a reverse proxy (HTTP accelerator) in front of an origin web server. In accelerator mode, Harvest would receive all requests and serve cacheable content itself, passing only cache-misses to the actual server on a different port. This technique could dramatically reduce load on origin servers without requiring any changes to clients

– an important capability for busy sites.

Another notable Harvest contribution was the definition of standard protocols for distributed caching and indexing. Harvest defined the Summary Object Interchange Format (SOIF) for exchanging index metadata between gatherers and brokers. And for caching, Harvest co-developed the Internet Cache Protocol (ICP) for communication among caches. ICP is a protocol allowing one cache to ask its neighbors (parents or siblings) if they have a particular URL, before retrieving it from the origin. This enables cooperative caching: a “miss” in one cache can be answered by a “hit” in another nearby cache, improving overall efficiency.

By 1997, ICP was published as RFC 2186/2187 by the IETF (authored by Duane Wessels and K. Claffy) as an informational specification, reflecting the fact that “several independent caching implementations” were already using ICP in practice. In summary, the Harvest project (1994–1996) established the state of the art for distributed web caching and indexing: it proved that hierarchical caches and distributed indexes could dramatically improve Internet performance, and it provided freely available software and protocols embodying these concepts. This was all well prior to 1998, demonstrating that a POSITA at the time was fully aware of these caching techniques.

2. Squid

Squid was an open-source web cache based on the Harvest object cache codebase after the Harvest project ran to completion.

<https://web.archive.org/web/20080119115602/http://www.squid-cache.org/Intro/>;

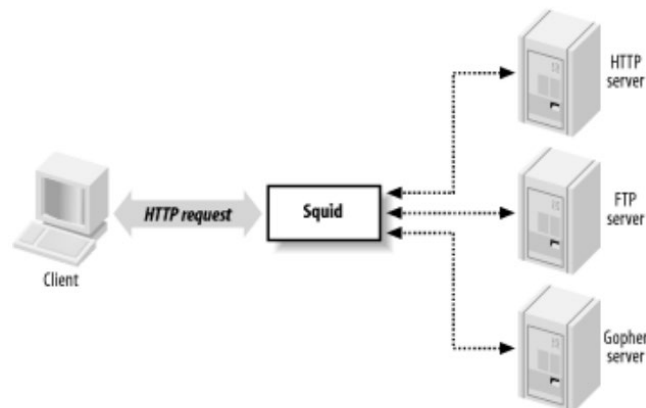
<https://www.squid-cache.org/Intro/>; Squid: The Definitive Guide By Duane Wessels (2004).

Developer Duane Wessels took the last version of Harvest’s cache, forked it, and renamed it “Squid” to avoid confusion with a separate commercial offshoot. Squid 1.0 was released to the public in July 1996 as a free software package. Squid retained Harvest’s fundamental architecture:

it is a caching and forwarding HTTP proxy that can serve cached content to clients, retrieve missing content from origin servers, and cooperate with other proxies in a hierarchy. Squid supports the ICP protocol for inter-cache queries, allowing it to be deployed in mesh topologies of sibling and parent caches (just as Harvest envisioned). Throughout 1996–1999, Squid was continuously improved by an active volunteer community and researchers (with NSF grants supporting its development). Notably, Squid added features like cache digests (a refinement on ICP to reduce query overhead) around 1997–98, further enhancing cooperative caching efficiency.

Internet Service Providers (“ISPs”), recognizing that they bore the brunt of the cost and burden of Internet traffic, also developed networks of servers to reduce traffic on their own networks. Many ISPs deployed systems called “web proxies” to distribute content to multiple users and thereby reduce traffic. Squid was widely deployed in ISP networks, universities, and large enterprises as a standard solution for web caching. As a cache, Squid stores recently retrieved web content for possible reuse later. Thereby, Squid would reduce bandwidth and improve response times by caching and reusing frequently-requested web pages. In practice, ISPs used Squid to cache web content at their network edges, thereby serving customers faster and cutting down on expensive upstream bandwidth usage.

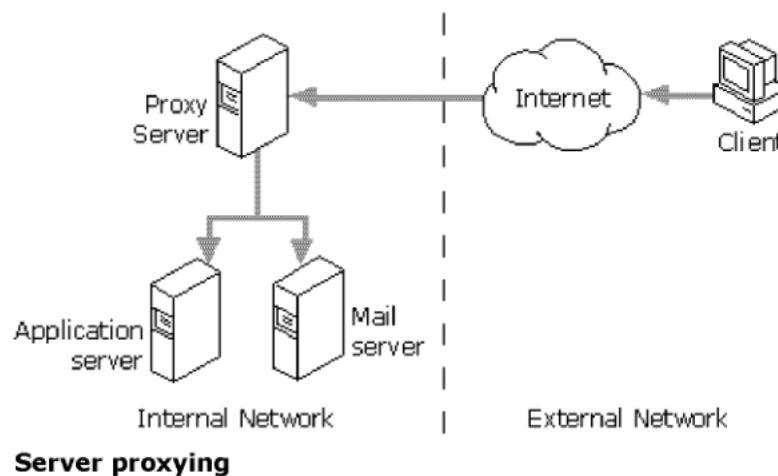
Figure 1-1. Squid sits between clients and servers



Squid: The Definitive Guide (Wessels) at 25.

3. Microsoft Proxy Server

Microsoft released its own Microsoft Proxy Server as early as 1996. Microsoft Proxy Server was designed to act as an intermediary between internal client computers and the Internet. In addition to routing requests, Microsoft Proxy Server provided a cache of frequently requested Internet sites, blocked access to specified sites, and provided secure access between internal networks and the Internet. Microsoft Proxy Server offered features including content caching (both distributed and hierarchical caching which reduced latency and optimized bandwidth usage), firewall integration, reverse proxy, SOCKS support, and Cache Array Routing Protocol (CARP). A diagram of Microsoft Proxy Server acting as a server proxy is shown below:



See Proxy Server Installation and Administration Guide at 13.

4. Akamai

Commercial CDNs also emerged such as Akamai. Akamai Technologies was founded in 1998 and deployed its own global CDN (“FreeFlow”). Akamai’s early network used a two-tier system: an “Akamai server” near the content source would relay streaming media to edge servers over dedicated lines, as described in public sources. These early CDNs could dramatically

improve web content delivery speed and reliability by caching content on hundreds of servers worldwide and redirecting users to the best node available.

B. CDN Concepts

1. Content Providers / Origin Servers

It was also well-understood that CDNs work in tandem with origin servers (sometimes called content providers' servers). The origin server is the primary source of content that could be replicated in a CDN. By the late 90s, any large website operated one or more origin servers, and CDNs were designed to fetch content from those origins and propagate it out to the edge. Prior art acknowledged that an "original web site" (origin server) could be mirrored by additional delivery sites to improve performance for users.

2. Content Servers

The use of distributed content servers was also well-understood by the late 1990s. Practitioners recognized that to achieve "near video on demand," one needed video servers capable of storing and streaming many movies simultaneously, distributed across the network (as noted in textbooks of the era). As noted above, by 1998 many Internet Service Providers (ISPs) were already deploying caching proxy servers within their networks to reduce bandwidth usage and latency for popular web content.

3. Caching and Caching Hierarchies

The concept of caching content was well known by content providers. There are many reasons for caching content: reducing origin server load, reducing latency, reducing bandwidth, and reducing the impact of traffic spikes.² Techniques for routing content from origins through intermediate nodes to end-users were also known. For example, hierarchical caching was a

² https://web.archive.org/web/20071011021519/http://www.deckle.co.za/squid-users-guide/Terminology_and_Technologies

common design: a cache hierarchy might include parent or relay proxies that fetch content from the origin and then supply it to child proxies or end-users. Academic papers in 1997 discussed hierarchical web caches and relays to reduce redundant traffic. By the end of the 90s, specialized routing mechanisms for CDNs were implemented. One known approach was to use DNS redirection or HTTP re-routing to direct a client's request to a cache node that either has the content or can fetch it. If the content was not on a given edge server, the request could be relayed to another node or up to the origin.

4. DNS

DNS (Domain Name System) was introduced in the early 1980s to replace centralized HOSTS.TXT file system, enabling scalable name resolution for a growing ARPANET and later the Internet. It was developed by the Internet Engineering Task Force (IETF) as an attempt to standardize a way to map pairings of logical host names to actual IP addresses on a TCP/IP network. Its hierarchical, distributed architecture allowed mapping human-readable names to IP addresses efficiently. The Harvest Cache, described above, relied on DNS resolution for hostname lookups. Squid. 1.0, described above, retained Harvest's architecture and added its own DNS resolution system. CDNs also began implementing DNS in the late 1990s to redirect user requests to optimal edge servers, including Akamai's DNS-based redirection. Microsoft also implemented DNS in products released in the 1990s. For example, Microsoft introduced a DNS server in Windows NT 3.51, which was later released into a product version with Windows NT Server 4.0 in 1996. For example, "[i]n addition to performing standard DNS, the DNS Server [provided with Windows NT Server version 4.0] c[ould] check a WINS names database for host names not resolved by DNS." *See Proxy Server Installation and Administration Guide* at 22. Proxy Server could also be used with DNS services. For example, Proxy Server documentation from 1997 explained that users could "use multiple Proxy Server computers on an internal network with DNS

services” and that “[e]ach entry indicate[d] an IP address to be searched for on the local network that provides DNS name resolution. The specified DNS server address is used to forward all domain name requests. In some cases, [users] c[ould] also have a secondary DNS server address to search for and use if the first DNS server [wa]s not available.” *Id.* at 24. Microsoft’s Internet Information Service, which was released with Windows NT Server 4.0, used DNS resolution to make its hosted services discoverable over the internet.

C. Growth of CDN

By the early 2000s, CDN technology had expanded dramatically. The dot-com era saw rapid adoption of CDNs for both static and streaming content. Several independent CDN providers were in operation such as Akamai (as discussed above), and many large enterprises without third-party CDNs were building their own content distribution solutions. By this time, the core concepts of edge caching, request routing (via DNS or HTTP redirects), and content replication were mainstream. A Cisco white paper published in 2000 outlined an “Enterprise CDN Solution,” underscoring that even private enterprises were implementing CDN techniques (edge caches, content management, load balancing) inside corporate networks. https://www.cisco.com/c/dam/global/tr_tr/assets/docs/Enterprise.pdf. In academic circles, research had moved to advanced topics like cooperative interconnection of CDNs.

Microsoft’s Windows Media Services (WMS) was a key enterprise streaming platform during the early 2000s, widely deployed for delivering audio and video content over corporate networks and the Internet. WMS supported both unicast and multicast streaming, enabling efficient delivery of live and on-demand media. Architecturally, WMS leveraged concepts described above: administrators could configure distributed streaming servers and use Network Load Balancing (NLB) to scale out clusters, ensuring high availability and fault tolerance. Microsoft documentation from this era emphasized policy-driven caching, plug-in modules for

custom logic, and integration with HTTP-based protocols, reflecting the convergence of traditional streaming servers and CDN technologies.

By 2008, the widespread adoption of broadband and rich media content in the mid-2000s (e.g. the rise of YouTube and other video services after 2005) led to even greater reliance on CDNs to deliver large files and streaming video. The fundamental architecture of CDNs remained the same – a network of surrogate servers at the “edge” serving content on behalf of origins – but the scale was larger and new features had been added. Academic books and surveys around 2008 document the standard CDN architecture in detail, including the typical components: (i) a content distribution and replication system (to push or pull content from origin to edge caches), (ii) a request-routing system (to direct users to the optimal edge server, using DNS resolution or HTTP redirection), (iii) a delivery network of edge cache nodes, and (iv) an accounting/management system (to monitor performance and collect usage data). These references confirm that by 2008 the design of CDNs was well-understood and widely published. For example, Pathan et al., “Content Delivery Networks: State of the Art, Insights, and Imperatives,” Springer 2008, provides a comprehensive overview of CDN technology and notes that CDNs have “evolved to overcome inherent limitations of the Internet... improving user perceived QoS by replicating content” on servers near users. Notably, by this time streaming media delivery via CDNs had become common. Early streaming (RealNetworks, Windows Media, etc.) in the late '90s was often handled by specialized servers; by 2008, those functions were integrated into CDNs using either dedicated streaming nodes or progressive download techniques. The concept of “edge streaming”—serving video or audio streams from edge caches—was known in the art (many CDN providers offered streaming services by mid-2000s). In addition, peer-to-peer (P2P) content distribution had emerged as a complementary approach: systems like BitTorrent (introduced 2001) showed that

large-scale content delivery could be achieved by cooperating end-nodes. While P2P is distinct from traditional CDNs, hybrid models were being explored in this timeframe (e.g., research on integrating P2P with CDN to reduce infrastructure load). By 2008, multiple CDN providers and platforms were available to content publishers, and the choice of using a CDN was a routine engineering decision for heavy content websites.

The existence of established CDN companies, numerous CDN patents (from companies like Akamai, Cisco, IBM, Microsoft, Limelight, etc.), and textbooks all indicate that a POSITA in 2008 would view distributed content delivery, caching, and load balancing as well-established arts. Indeed, new challenges being addressed in 2008 – such as interconnecting CDNs or handling dynamic personalized content – only arose because basic CDN capabilities were considered baseline technology by then.

Entering the 2010s, Internet traffic was increasingly dominated by high-bandwidth content like video streaming, and CDN technology evolved to meet this demand. By 2011, global content providers (Netflix, for example) were pushing unprecedented volumes of data through CDNs. It was public knowledge that companies like Netflix and YouTube accounted for a huge fraction of Internet traffic, and they relied on content delivery networks to handle it. To cope with such scale, the state of the art included multi-tiered CDN architectures and content peering arrangements. Large content providers not only used third-party CDNs (Akamai, Limelight, etc.), but also began deploying their own content delivery infrastructure.

By December 15, 2014 (the latest priority date among the Asserted Patents), the content delivery landscape was even more advanced, but fundamentally continuous with the prior art of earlier years. The concept of a distributed network of content servers delivering web and streaming

content was decades old by 2014. All the major innovations claimed by the Asserted Patents were already foreshadowed by much earlier systems and publications.

D. Microservices

The mid-2000s saw the growth of a software engineering concept called “microservices.” In a “microservices” approach to software architecture, components (known as “microservices”) are built smaller and more modular, and interconnected using APIs. Service Fabric White Paper at 1. This approach allows individual components to be written in different languages, to be added and removed seamlessly, and to be easily scaled up and down. *Id.* A cloud application designed in a “monolithic” fashion might be unpredictably affected by outages whereas a microservice-based design isolates problems and allows them to be easily serviced. *Id.*

“Service Fabric” is Microsoft’s platform to support microservice applications in cloud settings. *Id.* Service Fabric is the culmination of over a decade and a half of design and development, with its design starting in the early 2000s. *Id.* Since 2007, “many critical production systems inside Microsoft have been running atop SF.” *Id.*

For example, as early as October 29, 2008, Microsoft released details on “Distributed Data Fabric” at the Professional Developers Conference 2008. <https://learn.microsoft.com/en-us/shows/pdc-pdc08/bb03>. In this presentation, Microsoft employees explained that the system they came up with for “building internet scale services,” and “large scale distributed system[s],” were powering something called “SQL Data Services” even then, in 2008. *Id.* The presenters explained the architecture, which consisted of several major subsystems leading to high availability and scalability. *Id.* These features included a Storage Unit, Consistency Unit, and Failover Unit (a.k.a. Partition). *Id.* The Microsoft presenters explained that they built a technology called “Distributed Fabric” that “solves the distributed parts of the problem.” *Id.* Distributed Fabric “orchestrates the SQL servers into a self-healing self-adjusting fabric” and ensures that

when failures occur, the system remains consistent, bringing in a replacement node if a node fails.

Id. Another application using this infrastructure was Microsoft’s “Velocity” caching solution, which used the same architecture as SQL Server to maintain cache availability.

IV. IDENTIFICATION OF PRIOR ART

Subject to Microsoft’s reservations of rights, Microsoft identifies each item of prior art that anticipates or renders obvious the Asserted Claims below. Microsoft incorporates by reference all prior art and invalidity theories from other litigations related to the Asserted Patents in documents produced at: MSFT_SACDN_000000001-MSFT_SACDN_000033175; SANDPIPER-M-00006023-SANDPIPER-M-00006168; SANDPIPER-M-00006822-SANDPIPER-M-00007128; SANDPIPER-M-00007129-SANDPIPER-M-0012961.

A. U.S. Patent No. 8,478,903

Invalidity claim charts identifying disclosures in the references identified in the tables below as to the Asserted Claims of the ’903 Patent are provided in attached Exhibits A1 through A8.

Table 1-A: Prior Art Patents for the ’903 Patent

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
4,495,570 A (“Kitajima”)	US	1985-01-22	1981-01-14
4,591,983 A (“Bennett”)	US	1986-05-27	1984-07-09
4,594,704 A (“Ollivier”)	US	1986-06-10	1983-11-23
4,726,017 A (“Krum”)	US	1988-02-16	1985-05-21
4,803,641 A (“Hardy”)	US	1989-02-07	1984-06-06
4,839,798 A (“Eguchi”)	US	1989-06-13	1984-11-07
4,874,784 A (“Clancey”)	US	1989-07-11	1987-07-13
4,920,432 A (“Eggers”)	US	1990-04-24	1988-01-12
4,922,417 A (“Churm”)	US	1990-05-01	1986-10-24
4,943,932 A (“Lark”)	US	1990-07-24	1986-11-25
4,949,187 A (“Cohen”)	US	1990-08-14	1988-12-16
4,949,248 (“Caro”)	US	1990-08-14	1988-07-15
5,029,232 A (“Nall”)	US	1990-08-14	1988-12-16
5,130,792 A (“Tindell”)	US	1992-07-14	1990-02-01

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
5,132,992 A (“Yurt”)	US	1992-07-21	1991-01-07
5,136,716 A (“Harvey”)	US	1992-08-04	1987-09-04
5,172,413 A (“Bradley”)	US	1992-12-15	1990-12-20
5,191,573 A (“Hair”)	US	1993-03-02	1988-06-13
2202572 (“Law”)	CA	1988-10-14	1997-04-14
0801487 A2 (“Blewett”)	EP	1997-10-15	1997-04-10
5,253,341 A (“Rozmanith”)	US	1993-10-12	1991-03-04
5,287,499 A (“Nemes”)	US	1994-02-15	1989-03-22
5,287,537 A (“Newmark”)	US	1994-02-15	1985-11-15
5,291,554 A (“Morales”)	US	1994-03-01	1992-05-28
5,341,477 A (“Pitken”)	US	1994-08-23	1989-02-24
5,371,532 A (“Gelman”)	US	1994-12-06	1992-05-15
2281793 A (“Wallis”)	GB	1995-03-15	1993-09-11
5,410,343 A (“Coddington”)	US	1995-04-25	1991-09-27
5,414,455 A (“Hooper”)	US	1995-05-09	1993-07-07
5,442,390 A (“Hooper”)	US	1995-08-15	1993-07-07
5,442,749 A (“Northcutt”)	US	1995-08-15	1991-08-22
WO 1996/41285 (“Kenner”)	WO	1996-12-19	1996-06-07
6,243,760 (“Armbruster”)	US	2001-06-05	1997-06-24
5,475,819 (“Miller”)	US	1995-12-12	1994-06-17
5,649,103 (“Datta”)	US	1997-07-15	1995-07-13

Table 2-A: Prior Art Publications for the '903 Patent

Title (Short Name)	Author	Publisher	Date of Publication
Andrew Tanenbaum, Computer Networks (3rd ed. 1996) (“Tanenbaum”)	Andrew Tanenbaum	PTR PH	Jan. 1, 1996
Tim Berners-Lee, The World-Wide Web, Communications of the ACM, Vol. 37, No. 8, August 1994 (“Berners-Lee”)	Tim Berners-Lee	Communications of the ACM	August 1994
Radhika Malpani, Making World Wide Web Caching Servers Cooperate, WWW4: Proceedings of the Fourth International Conference on World Wide Web, Pages 107-117, December 11, 1995 (“Malpani”)	Radhika Malpani	WWW4: Proceedings of the Fourth International Conference on World Wide Web	December 11, 1995
Fielding et al., Hypertext Transfer Protocol – HTTP/1.1,	R. Fielding et al.	Network Working Group	January 1997

Title (Short Name)	Author	Publisher	Date of Publication
RFC 2068, Jan. 1997, https://www.rfc-editor.org/rfc/rfc2068 ("HTTP/1.1")			
Barbara Tockey Zivkov et al., Disk Caching in Large Databases and Timeshared Systems, Proceedings Fifth International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 97), Haifa, Israel, 184– 95, January 1997 ("Zivkov")	Barbara Tockey Zivkov et al.	University of California, Berkeley	January 1997
Ronald J. Vetter et. al, Mosaic and the World-Wide Web, Computer, vol. 27, no. 10, pp. 49-57, Oct. 1994	Ronald J. Vetter et. al	Computer, vol. 27, no. 10	October 1994
Network Working Group, Request for Comments: 1034 ("RFC 1034")	P. Mockapetris	Network Working Group	November 1987
Network Working Group, Request for Comments: 2181 ("RFC 2181")	R. Elz et al.	Network Working Group	July 1997
The Cache System at the National Laboratory for Applied Network Research ("NLANR")	National Laboratory for Applied Network Research	National Laboratory for Applied Network Research	March 7, 1997
A Hierarchical Internet Object Cache ("Chankhunthod")	Anawat, Chankhunthod	USENIX 1996 TECHNICAL CONFERENCE, January.	1996
Fisher, Spinning the Web, Spring- Verlag New York, Inc. (1996) ("Fisher")	Fisher	Spinner-Verlag New York, Inc.	1996
Ricart, Apache Server, Sams.net Publishing (1996) ("Ricart")	Ricart	Sams.net Publishing	1996
S. Ramanathan and P. Rangan, "Architectures for Personalized Multimedia" IEEE Multimedia, 37-46 (Spring 1994)	S. Ramanathan and P. Rangan	IEEE Multimedia	May 31, 1994
Bradley, "Impact of the Internet and Online Services on Business Research Methods," Massachusetts Institute of Technology (June 1996)	Bradley	Massachusetts Institute of Technology	June 1, 1996
Vahdat et al., "WebOS: Software Support for Scalable Web Services," University of California – Berkeley	Vahdat et al	University of California – Berkeley	March 1, 1997

Title (Short Name)	Author	Publisher	Date of Publication
Baentsch et al., “Enhancing the Web’s Infrastructure from Caching to Replication,” IEEE Internet Computing	Baentsch et al	IEEE Internet Computing	April 30, 1997
Chawla and Bettati, “Replicating IP Services,” Technical Report 97-008, Texas A&M University (Sept. 1997)	Chawla and Bettati	Texas A&M University	September 1, 1997
Miller and Hollingsworth, “Using Content- Derived Names for Caching and Software Distribution,” University of Maryland	Miller and Hollingsworth	University of Maryland	October 15, 1998

Table 3-A: Prior Art Sales or Public Disclosures for the ’903 Patent

Name of System or Invention	Identity of the Relevant Person or Entity	Date of Sale / Offer for Sale / Public Use / Known by Others
Microsoft Proxy Server 2.0 Reference and System, 0797 Part No. 96218 (see also disk image MSPROXY2.0.ISO)	Microsoft	At least by 1997
IBM CDN systems, including IBM Web Object Management (“WOM”) & Network Dispatcher & other CDN systems	International Business Machines Corp.	At least by 1998
Harvest System	University of Colorado at Boulder (Internet Research Task Force Research Group); Peter Danzig	At least by 1995
Squid Cache	Squid Software Foundation; Duane Wessels	At least by 1996
Footprint CDN, developed and operated by Sandpiper Networks, Inc., Digital Island, Inc., Cable & Wireless PLC, Savvis Inc., Level 3 Communications, Century Link, Inc., and Lumen Technologies, Inc.	Sandpiper Networks, Inc., Digital Island, Inc., Cable & Wireless PLC, Savvis Inc., Level 3 Communications, Century Link, Inc., and Lumen Technologies, Inc.	At least by 1998
Cisco CDN products, including Cisco/Sightpath	Cisco Systems	At least by 1998

Name of System or Invention	Identity of the Relevant Person or Entity	Date of Sale / Offer for Sale / Public Use / Known by Others
products and Cisco/Precept IP/TV Content Manager		
InterVu CDN systems	InterVU, Inc.	At least by 1997
Microsoft Internet Information Server 4.0 (“IIS”)	Microsoft	At least by 1996
Akamai CDN	Akamai Technologies, Inc.	At least by 1988
Apache	The Apache Software Foundation	At least by 1995
Mirror-Image	Mirror Image Internet, Inc.	At least by 1997
CacheFlow	Broadcom, Inc.	At least by 1998
Resonate Central Dispatch	Resonate, Inc.	At least by 1998

Table 4-A: Charts for the ’903 Patent

Exhibit No.	Prior Art Reference/System
A1	Armbruster
A2	Kenner
A3	NLANR
A4	RFC1034
A5	Microsoft Proxy Server
A6	Microsoft IIS 4.0
A7	Squid Cache
A8	Wallis ’793

Microsoft incorporates by reference the invalidity claim charts and contentions served by Google in the *Sandpiper CDN, LLC v. Google LLC*, 2:24-cv-03951 (C.D. Cal.) matter and Comcast in the *Sandpiper CDN, LLC v. Comcast Cable Comms., LLC d/b/a Xfinity et al*, 2:24-cv-00886 (E.D. Tex.) matter.

For each of the prior art devices identified above, Microsoft has listed and produced documents as evidence of the relevant features and functionality to the extent that they are presently available and have been located. Microsoft has obtained or is in the process of attempting to obtain the identified devices and will make them available for inspection to the extent they are

obtained. To the extent that one or more documents (*e.g.*, user manual) may be used to describe aspects of a particular device, that device is a single reference for prior art purposes under 35 U.S.C. § 102. Some or all of the corroborating references may also separately qualify as prior art publications under 35 U.S.C. § 102 and may be used as invalidating references under 35 U.S.C. §§ 102 and/or 103.

Further, Microsoft is actively searching for information regarding at least the systems, devices, and/or inventions listed above in Table 3-A.

Discovery is ongoing, and Microsoft may serve third parties with document subpoenas. One or more of these devices, along with related documentation, may be invalidating, and Microsoft reserves the right to supplement these contentions accordingly. Microsoft continues to investigate each of the identified devices, including through potential discovery needed to be obtained via third party subpoenas, and reserve the right to supplement these contentions and accompanying claim charts after further investigation.

Microsoft's reference to a particular product, device, or software program in these contentions should be interpreted as a reference to the system itself as well as to any corresponding patents, publications, or product literature relating to the cited system. Upon information and belief, the systems were publicly disclosed, used, sold, or offered for sale in the United States before the alleged priority date of the Asserted Claims of the '903 Patent. Microsoft's investigation of such prior art systems is still ongoing and discovery has not yet been received from third parties who may have information concerning such prior art systems. Accordingly, subsequent discovery may reveal information that affects the disclosures and contentions herein. For example, subsequent discovery may provide additional information regarding whether or not any of the

third-party prior art systems anticipate or render obvious the Asserted Claims of the '903 Patent. As such, Microsoft reserves all rights to supplement their invalidity contentions.

Many of the inventive, research, design, and development activities concerning these systems and technologies occurred in the United States before the alleged priority date of the Asserted Claims of the '903 Patent. Microsoft has obtained, and is in the process of obtaining, additional information regarding the dates by which the cited products and services were publicly disclosed, used, sold, or offered for sale, the circumstances under which the research, design, and development activities were conducted, and the identities of the particular individuals involved in such activities through publicly available patents, publications, and product literature. The actual dates, circumstances, and identities of individuals will be the subject of third party discovery during this lawsuit, which Microsoft reserves the right to rely upon to corroborate the prior art status of the prior art identified herein. Microsoft reserves the right to modify, amend, or supplement these contentions if additional information becomes available during the course of this lawsuit. Furthermore, Microsoft will make all such devices, software programs, or other products in Microsoft's possession, custody, or control available for inspection by Plaintiff.

Consequently, Microsoft reserves the right to amend, modify, or supplement these Invalidity Contentions should additional information become available to them through discovery.

In addition, Microsoft identifies, responsive to Plaintiff's Infringement Contentions, the patents, publications, and/or systems listed in Tables 1-A, 2-A, and 3-A as evidence of the state of the art as it relates to the '903 Patent.

Each of these references show that the concepts of user authentication, proximity sensors, and location tracking are not novel and were well-known for many years before the alleged priority

date of the ‘903 Patent, as discussed in more detail in Section VI.B (obviousness). In light of the references discussed herein, the Asserted Claims of the ‘903 Patent cannot be valid.

Each prior art patent, publication, or product identified above was either effectively filed or issued (for patents), published (for publications) or known, used, offered for sale or sold (for products) before either the earliest claimed priority date of the ‘903 Patent to which it is applied for invalidity, and none appears to have been abandoned, suppressed, or concealed, so each such reference also constitutes evidence of prior invention pursuant to 35 U.S.C. § 102(g), if it is in the U.S. The persons or entities involved with each such invention include the named inventors on the above-identified patents, the authors listed on the above-identified publications, and the entities and individuals identified in connection with the above-identified products.

Because Microsoft has not yet completed discovery in this case, including taking depositions of the named inventors, and otherwise obtaining discovery from Plaintiff and third parties, Microsoft reserves the right to supplement these contentions with facts, documents, or other information learned at a later point through discovery or further investigation.

B. U.S. Patent No. 8,924,466

Invalidity claim charts identifying disclosures in the references identified in the tables below as to the Asserted Claims of the ‘466 Patent are provided in attached Exhibits B1 through B13.

Table 1-B: Prior Art Patents for the ‘466 Patent

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
4,495,570 (“Kitajima”)	US	1985-01-22	1981-01-14
4,591,983 (“Bennett”)	US	1986-05-27	1984-07-09
4,594,704 (“Ollivier”)	US	1986-06-10	1983-11-23
4,726,017 (“Krum”)	US	1988-02-16	1985-05-21
4,803,641 (“Hardy”)	US	1989-02-07	1984-06-06
4,839,798 (“Eguchi”)	US	1989-06-13	1984-11-07

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
4,847,784 (“Clancey”)	US	1989-07-11	1987-07-13
4,949,187 A (“Cohen”)	US	1990-08-14	1998-12-16
4,949,248 A (“Caro”)	US	1990-08-14	1988-07-15
5,029,232 A (“Nall”)	US	1991-07-02	1989-01-12
5,130,792 A (“Tindell”)	US	1992-07-14	1990-02-01
5,132,992 A (“Yurt”)	US	1992-07-21	1991-01-07
5,136,716 A (“Harvey”)	US	1992-08-04	1987-09-04
5,172,413 A (“Bradley”)	US	1992-12-15	1990-12-20
5,191,573 A (“Hair”)	US	1993-03-02	1988-06-13
5,253,275 A (“Yurt”)	US	1993-10-12	1991-01-07
5,253,341 A (“Rozmanith”)	US	1993-10-12	1991-03-04
5,287,499 A (“Nemes”)	US	1994-02-15	1989-03-22
5,291,554 A (“Morales”)	US	1994-03-01	1992-05-28
5,341,477 A (“Pitkin”)	US	1994-08-23	1989-02-24
5,371,532 A (“Gelman”)	US	1994-12-06	1992-05-15
5,410,343 A (“Coddington”)	US	1995-04-25	1991-09-27
5,414,455 A (“Hooper”)	US	1995-05-09	1993-07-07
5,442,749 A (“Northcutt”)	US	1995-08-15	1991-08-22
5,442,390 A (“Hooper”)	US	1995-08-15	1993-07-07
5,442,389 A (“Blahut”)	US	1995-08-15	1992-12-28
5,471,622 A (“Eadline”)	US	1995-11-28	1989-10-04
5,475,615 A (“Lin”)	US	1995-12-12	1993-12-23
5,508,732 A (“Bottomley”)	US	1996-04-16	1993-03-22
6,167,438 (“Yates”)	US	2000-12-26	1997-05-22
6,687,846 (“Adrangi”)	US	2004-02-03	2000-03-30
7,024,466 (“Outten”)	US	2006-04-04	2001-04-06
2002/0016835 (“Gamerman”)	US	2002-02-07	2001-05-24
2002/0198953 (“O’Rourke”)	US	2006-01-10	2001-06-26
2003/0112792 (“Cranor”)	US	2005-10-11	2001-12-14
6,370,620 (“Wu”)	US	2002-0-09	1999-01-28
2001/055878 (“Lahr”)	WO	2001-08-02	2001-01-29
6,785,704 (“McCanne”)	US	2000-07-03	2004-08-31

Table 2-B: Prior Art Publications for the ’466 Patent

Title (Short Name)	Author	Publisher	Date of Publication
Selecting an Internet content Delivery Service for Streaming Media (“Streaming Paper”)	Akamai	Akamai	April 2000

Title (Short Name)	Author	Publisher	Date of Publication
The Network Providers Business Case For Internet Content Delivery (“IRG Paper”)	Peter Christy	Internet Research Group	1999
Fast Internet Content Delivery with FreeFlow (“FreeFlow White Paper”)	Akamai	Akamai	November 1999
The Challenges of Delivering Content on the Internet (“Leighton Slides”)	Tom Leighton	Akamai	August 10, 2001
Internet Bottlenecks: the Case for Edge Delivery Services (“Bottlenecks Paper”)	Akamai	Akamai	2000
Akamai Edge Suite – Next-Generation Content Delivery (January 2001) (“EdgeSuite Product Sheet”)	Akamai	Akamai	January 2001
Turbo-Charging Dynamic Web Sites with Akamai EdgeSuite (“EdgeSuite White Paper”)	Akamai	Akamai	2001
RFC2187: Application of Internet Cache Protocol (ICP), version 2 (“RFC2187”)	Duane Wessels, Kimberly Claffy	Internet Engineering Task Force	September 1997

Table 3-B: Prior Art Sales or Public Disclosures for the ’466 Patent

Name of System or Invention	Identity of the Relevant Person or Entity	Date of Sale / Offer for Sale / Public Use / Known by Others
Akamai EdgeSuite	Akamai Technologies, Inc.	At least by August 20, 2001
Akamai FreeFlow	Akamai Technologies, Inc.	At least by June 2, 1999
Windows 2000 Server and Windows Media Services v4.1	Microsoft Corporation	December 15, 1999

Table 4-B: Charts for the ’466 Patent

Exhibit No.	Prior Art Reference/System
B1	Cranor
B2	Gamerman
B3	McCanne
B4	O’Rourke
B5	Outten
B6	RFC2187
B7	Adurangi
B8	Lahr
B9	Wu

Exhibit No.	Prior Art Reference/System
B10	Yates
B11	Akamai FreeFlow
B12	Akamai EdgeSuite
B13	Windows 2000 Server and Windows Media Services v4.1

For each of the prior art devices identified above, Microsoft has listed and produced documents as evidence of the relevant features and functionality to the extent that they are presently available and have been located. Microsoft has obtained or is in the process of attempting to obtain the identified devices and will make them available for inspection to the extent they are obtained. To the extent that one or more documents (*e.g.*, user manual) may be used to describe aspects of a particular device, that device is a single reference for prior art purposes under 35 U.S.C. § 102. Some or all of the corroborating references may also separately qualify as prior art publications under 35 U.S.C. § 102 and may be used as invalidating references under 35 U.S.C. §§ 102 and/or 103.

Further, Microsoft is actively searching for information regarding at least the systems, devices, and/or inventions listed above in Table 3-B.

Discovery is ongoing, and Microsoft may serve third parties with document subpoenas. One or more of these devices, along with related documentation, may be invalidating, and Microsoft reserves the right to supplement these contentions accordingly. Microsoft continues to investigate each of the identified devices, including through potential discovery needed to be obtained via third party subpoenas, and reserve the right to supplement these contentions and accompanying claim charts after further investigation.

Microsoft's reference to a particular product, device, or software program in these contentions should be interpreted as a reference to the system itself as well as to any corresponding patents, publications, or product literature relating to the cited system. Upon information and

belief, the systems were publicly disclosed, used, sold, or offered for sale in the United States before the alleged priority date of the Asserted Claims of the '466 Patent. Microsoft's investigation of such prior art systems is still ongoing and discovery has not yet been received from third parties who may have information concerning such prior art systems. Accordingly, subsequent discovery may reveal information that affects the disclosures and contentions herein. For example, subsequent discovery may provide additional information regarding whether or not any of the third-party prior art systems anticipate or render obvious the Asserted Claims of the '466 Patent. As such, Microsoft reserves all rights to supplement their invalidity contentions.

Many of the inventive, research, design, and development activities concerning these systems and technologies occurred in the United States before the alleged priority date of the Asserted Claims of the '466 Patent. Microsoft has obtained, and is in the process of obtaining, additional information regarding the dates by which the cited products and services were publicly disclosed, used, sold, or offered for sale, the circumstances under which the research, design, and development activities were conducted, and the identities of the particular individuals involved in such activities through publicly available patents, publications, and product literature. The actual dates, circumstances, and identities of individuals will be the subject of third party discovery during this lawsuit, which Microsoft reserves the right to rely upon to corroborate the prior art status of the prior art identified herein. Microsoft reserves the right to modify, amend, or supplement these contentions if additional information becomes available during the course of this lawsuit. Furthermore, Microsoft will make all such devices, software programs, or other products in Microsoft's possession, custody, or control available for inspection by Plaintiff.

Consequently, Microsoft reserves the right to amend, modify, or supplement these Invalidity Contentions should additional information become available to them through discovery.

In addition, Microsoft identifies, responsive to Plaintiff's Infringement Contentions, the patents, publications, and/or systems listed in Tables 1-B, 2-B, and 3-B as evidence of the state of the art as it relates to the '466 Patent.

Each of these references show that the concepts of user authentication, proximity sensors, and location tracking are not novel and were well-known for many years before the alleged priority date of the '466 Patent, as discussed in more detail in Section VI.B (obviousness). In light of the references discussed herein, the Asserted Claims of the '466 Patent cannot be valid.

Each prior art patent, publication, or product identified above was either effectively filed or issued (for patents), published (for publications) or known, used, offered for sale or sold (for products) before either the earliest claimed priority date of the '466 Patent to which it is applied for invalidity, and none appears to have been abandoned, suppressed, or concealed, so each such reference also constitutes evidence of prior invention pursuant to 35 U.S.C. § 102(g), if it is in the U.S. The persons or entities involved with each such invention include the named inventors on the above-identified patents, the authors listed on the above-identified publications, and the entities and individuals identified in connection with the above-identified products.

Because Microsoft has not yet completed discovery in this case, including taking depositions of the named inventors, and otherwise obtaining discovery from Plaintiff and third parties, Microsoft reserves the right to supplement these contentions with facts, documents, or other information learned at a later point through discovery or further investigation.

C. U.S. Patent No. 9,456,053

Invalidity claim charts identifying disclosures in the references identified in the tables below as to the Asserted Claims of the '053 Patent are provided in attached Exhibits C1 through C10.

Table 1-C: Prior Art Patents for the '053 Patent

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
5,511,208 A (“Boyles”)	US	1996-04-23	1993-03-23
5,805,837 A (“Hoover”)	US	1998-09-08	1996-03-21
5,870,559 A (“Leshem”)	US	1999-02-09	1996-10-15
5,951,694 A (“Choquier”)	US	1999-09-14	1995-06-07
6,173,322 B1 (“Hu”)	US	2001-01-09	1997-06-05
6,185,598 B1 (“Farber”)	US	2001-02-06	1998-02-10
6,212,178 B1 (“Beck”)	US	2001-04-03	1998-09-11
6,226,694 B1 (“Constant”)	US	2001-05-01	1998-04-29
6,279,032 B1 (“Short”)	US	2001-08-21	1997-11-03
6,484,143 B1 (“Swildens”)	US	2002-11-19	1999-11-22
6,571,261 B1 (“Wang-Knop”)	US	2003-05-27	2000-07-13
6,587,928 B1 (“Periyannan”)	US	2003-07-01	2000-02-28
6,654,807 B2 (“Farber”)	US	2004-06-29	2000-03-03
6,757,708 B1 (“Craig”)	US	2004-08-19	2003-02-13
6,965,930 B1 (“Arrowood”)	US	2005-11-15	2000-10-20
7,010,578 B1 * (“Lewin”)	US	2006-03-07	2000-09-21
7,054,935 B2 (“Farber”)	US	2006-05-30	1998-02-10
7,062,556 B1 (“Chen”)	US	2006-06-13	1999-11-22
7,076,608 B2 (“Arunagirinathan”)	US	2006-07-11	2003-12-02
7,136,649 B2 (“Ying”)	US	2006-11-14	2002-08-23
7,240,100 (“Wein”)	US	2007-07-03	2001-04-16
7,320,085 B2 (“Bain”)	US	2008-01-15	2004-03-09
7,370,102 B1 (“Chu”)	US	2008-05-06	1998-12-15
7,395,346 B2 (“Pinder”)	US	2008-07-01	2003-04-22
7,461,206 B2 (“Bhanoo”)	US	2008-12-02	2006-08-21
7,512,702 B1 (“Srivastava”)	US	2009-03-31	2002-03-19
7,512,707 B1 (“Manapradada”)	US	2009-03-31	2005-11-03
7,523,181 B2 (“Swildens”)	US	2009-04-21	1999-11-22
7,797,426 B1 (“Lyon”)	US	2010-09-14	2008-06-27
7,822,871 B2 (“Stolorz”)	US	2010-10-26	2001-09-28
8,423,662 (“Weihl”)	US	2013-04-16	2004-04-28
2000/13456 (“Crosbie”)	WO	2000-03-09	1999-08-31
2010/0257258 (“Liu”)	US	2010-10-07	2010-05-25
2007/0156845 (“Devanneaux”)	US	2006-12-29	2007-07-05

Table 2-C: Prior Art Publications for the '053 Patent

Title (Short Name)	Author	Publisher	Date of Publication
<i>The Akamai Network: A Platform for High-Performance Internet Applications ("Nygren")</i>	Erik Nygren et al.	ACM SIGOPS Operating Systems Review	July 2010

Table 3-C: Prior Art Sales or Public Disclosures for the '053 Patent

Name of System or Invention	Identity of the Relevant Person or Entity	Date of Sale / Offer for Sale / Public Use / Known by Others
Akamai CDN's Configuration Management System	Akamai Technologies	As early as 2005
Microsoft Velocity Cache	Microsoft	As early as September 2009
Microsoft System Center Configuration Manager	Microsoft	As early as 2007

Table 4-C: Charts for the '053 Patent

Exhibit No.	Prior Art Reference/System
C1	Lewin
C2	Crosbie
C3	Nygren
C4	Weihl
C5	Wein
C6	Akamai CDN's Configuration Management System
C7	Microsoft Velocity
C8	Microsoft System Center Configuration Manager
C9	Devanneaux
C10	Liu

For each of the prior art devices identified above, Microsoft has listed and produced documents as evidence of the relevant features and functionality to the extent that they are presently available and have been located. Microsoft has obtained or is in the process of attempting to obtain the identified devices and will make them available for inspection to the extent they are obtained. To the extent that one or more documents (*e.g.*, user manual) may be used to describe aspects of a particular device, that device is a single reference for prior art purposes under 35 U.S.C. § 102. Some or all of the corroborating references may also separately qualify as prior art publications under 35 U.S.C. § 102 and may be used as invalidating references under 35 U.S.C. §§ 102 and/or 103.

Further, Microsoft is actively searching for information regarding at least the systems, devices, and/or inventions listed above in Table 3-C.

Discovery is ongoing, and Microsoft may serve third parties with document subpoenas. One or more of these devices, along with related documentation, may be invalidating, and Microsoft reserves the right to supplement these contentions accordingly. Microsoft continues to investigate each of the identified devices, including through potential discovery needed to be obtained via third party subpoenas, and reserve the right to supplement these contentions and accompanying claim charts after further investigation.

Microsoft's reference to a particular product, device, or software program in these contentions should be interpreted as a reference to the system itself as well as to any corresponding patents, publications, or product literature relating to the cited system. Upon information and belief, the systems were publicly disclosed, used, sold, or offered for sale in the United States before the alleged priority date of the Asserted Claims of the '053 Patent. Microsoft's investigation of such prior art systems is still ongoing and discovery has not yet been received from third parties who may have information concerning such prior art systems. Accordingly, subsequent discovery may reveal information that affects the disclosures and contentions herein. For example, subsequent discovery may provide additional information regarding whether or not any of the third-party prior art systems anticipate or render obvious the Asserted Claims of the '053 Patent. As such, Microsoft reserves all rights to supplement their invalidity contentions.

Many of the inventive, research, design, and development activities concerning these systems and technologies occurred in the United States before the alleged priority date of the Asserted Claims of the '053 Patent. Microsoft has obtained, and is in the process of obtaining, additional information regarding the dates by which the cited products and services were publicly

disclosed, used, sold, or offered for sale, the circumstances under which the research, design, and development activities were conducted, and the identities of the particular individuals involved in such activities through publicly available patents, publications, and product literature. The actual dates, circumstances, and identities of individuals will be the subject of third party discovery during this lawsuit, which Microsoft reserves the right to rely upon to corroborate the prior art status of the prior art identified herein. Microsoft reserves the right to modify, amend, or supplement these contentions if additional information becomes available during the course of this lawsuit. Furthermore, Microsoft will make all such devices, software programs, or other products in Microsoft's possession, custody, or control available for inspection by Plaintiff.

Consequently, Microsoft reserves the right to amend, modify, or supplement these Invalidity Contentions should additional information become available to them through discovery.

In addition, Microsoft identifies, responsive to Plaintiff's Infringement Contentions, the patents, publications, and/or systems listed in Tables 1-C, 2-C, and 3-C as evidence of the state of the art as it relates to the '053 Patent.

Each of these references show that the concepts of user authentication, proximity sensors, and location tracking are not novel and were well-known for many years before the alleged priority date of the '053 Patent, as discussed in more detail in Section VI.B (obviousness). In light of the references discussed herein, the Asserted Claims of the '053 Patent cannot be valid.

Each prior art patent, publication, or product identified above was either effectively filed or issued (for patents), published (for publications) or known, used, offered for sale or sold (for products) before either the earliest claimed priority date of the '053 Patent to which it is applied for invalidity, and none appears to have been abandoned, suppressed, or concealed, so each such reference also constitutes evidence of prior invention pursuant to 35 U.S.C. § 102(g), if it is in the

U.S. The persons or entities involved with each such invention include the named inventors on the above-identified patents, the authors listed on the above-identified publications, and the entities and individuals identified in connection with the above-identified products.

Because Microsoft has not yet completed discovery in this case, including taking depositions of the named inventors, and otherwise obtaining discovery from Plaintiff and third parties, Microsoft reserves the right to supplement these contentions with facts, documents, or other information learned at a later point through discovery or further investigation.

D. U.S. Patent No. 9,722,883

Invalidity claim charts identifying disclosures in the references identified in the tables below as to the Asserted Claims of the '883 Patent are provided in attached Exhibits D1 through D9.

Table 1-D: Prior Art Patents for the '883 Patent

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
5,511,208 A ("Boyles")	US	1996-04-23	1993-03-23
5,805,837 A ("Hoover")	US	1998-09-08	1996-03-21
5,870,559 A ("Lesham")	US	1999-02-09	1996-10-15
5,951,694 A ("Choquier")	US	1999-09-14	1995-06-07
5,987,506 A ("Carter")	US	1999-11-16	1996-11-22
6,173,322 B1 ("Hu")	US	2001-01-09	1997-06-05
6,185,598 B1 ("Farber")	US	2001-02-06	1998-02-10
6,212,178 B1 ("Beck")	US	2001-04-03	1998-09-11
6,226,694 B1 ("Constant")	US	2001-05-01	1998-04-29
6,279,032 B1 ("Short")	US	2001-08-21	1997-11-03
6,484,143 B1 ("Swildens")	US	2002-11-19	1999-11-22
6,571,261 B1 ("Wang-Knop")	US	2003-05-27	2000-07-13
6,577,597 B1 ("Natarajan")	US	2003-06-10	1999-06-29
6,587,928 B1 ("Periyannan")	US	2003-07-01	2000-02-28
6,757,708 B1 ("Craig")	US	2004-06-29	2000-03-03
6,965,930 B1 ("Arrowood")	US	2005-11-15	2000-10-20

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
6,981,105 B2 (“Bourne”)	US	2005-12-27	1999-07-22
7,010,578 B1 (“Lewin”)	US	2006-03-07	2000-09-21
7,054,935 B2 (“Farber”)	US	2006-05-30	1998-02-10
7,062,556 B1 (“Chen”)	US	2006-06-13	1999-11-22
7,076,608 B2 (“Arunagirinathan”)	US	2006-07-11	2003-12-02
7,136,649 B2 (“Ying”)	US	2006-11-14	2002-08-23
2427490A (“Brunet”)	GB	2006-12-27	2005-06-22
U.S. Publication No. 2007/0086430 A1 (“Kemp”)	US	2007-04-19	2005-10-14
7,370,102 B1 (“Chu”)	US	2008-05-06	1998-12-15
7,395,346 B2 (“Pinder”)	US	2008-07-01	2003-04-22
7,461,206 B2 (“Bhanoo”)	US	2008-12-02	2006-08-21
6,341,311 (“Smith”)	US	2002-01-22	1998-05-29
6,112,279 (“Wang”)	US	2000-08-29	1998-03-31
2003/0014538 (“Roadknight”)	US	2009-01-13	2001-01-24
6,772,225 (“Jennings”)	US	2004-08-03	1999-09-30

Table 2-D: Prior Art Publications for the ’883 Patent

Title (Short Name)	Author	Publisher	Date of Publication
<i>IBM Load Balancer for IPv4 Administration, Version 7.0 (First Edition) (“IBM Load Balancer”)</i>	IBM	IBM	June 2008

Table 3-D: Prior Art Sales or Public Disclosures for the '883 Patent

Name of System or Invention	Identity of the Relevant Person or Entity	Date of Sale / Offer for Sale / Public Use / Known by Others
Lync Server 2013	Microsoft Corporation	At least as early as October 11, 2012
Windows Fabric	Microsoft Corporation	At least as early as October 11, 2012
Velocity Cache	Microsoft Corporation	At least as early as September 1, 2009
IBM Load Balancer	IBM	At least as early as June 2008

Table 4-D: Charts for the '883 Patent

Exhibit No.	Prior Art Reference/System
D1	IBM Load Balancer
D2	Jennings
D3	Kemp '430
D4	Roadknight
D5	Smith
D6	Wang
D7	Lync Server 2013
D8	Windows Fabric
D9	Velocity Cache

For each of the prior art devices identified above, Microsoft has listed and produced documents as evidence of the relevant features and functionality to the extent that they are presently available and have been located. Microsoft has obtained or is in the process of attempting to obtain the identified devices and will make them available for inspection to the extent they are obtained. To the extent that one or more documents (*e.g.*, user manual) may be used to describe aspects of a particular device, that device is a single reference for prior art purposes under 35 U.S.C. § 102. Some or all of the corroborating references may also separately qualify as prior art publications under 35 U.S.C. § 102 and may be used as invalidating references under 35 U.S.C. §§ 102 and/or 103.

Further, Microsoft is actively searching for information regarding at least the systems, devices, and/or inventions listed above in Table 3-D.

Discovery is ongoing, and Microsoft may serve third parties with document subpoenas. One or more of these devices, along with related documentation, may be invalidating, and Microsoft reserves the right to supplement these contentions accordingly. Microsoft continues to investigate each of the identified devices, including through potential discovery needed to be obtained via third party subpoenas, and reserve the right to supplement these contentions and accompanying claim charts after further investigation.

Microsoft's reference to a particular product, device, or software program in these contentions should be interpreted as a reference to the system itself as well as to any corresponding patents, publications, or product literature relating to the cited system. Upon information and belief, the systems were publicly disclosed, used, sold, or offered for sale in the United States before the alleged priority date of the Asserted Claims of the '883 Patent. Microsoft's investigation of such prior art systems is still ongoing and discovery has not yet been received from third parties who may have information concerning such prior art systems. Accordingly, subsequent discovery may reveal information that affects the disclosures and contentions herein. For example, subsequent discovery may provide additional information regarding whether or not any of the third-party prior art systems anticipate or render obvious the Asserted Claims of the '883 Patent. As such, Microsoft reserves all rights to supplement their invalidity contentions.

Many of the inventive, research, design, and development activities concerning these systems and technologies occurred in the United States before the alleged priority date of the Asserted Claims of the '883 Patent. Microsoft has obtained, and is in the process of obtaining, additional information regarding the dates by which the cited products and services were publicly

disclosed, used, sold, or offered for sale, the circumstances under which the research, design, and development activities were conducted, and the identities of the particular individuals involved in such activities through publicly available patents, publications, and product literature. The actual dates, circumstances, and identities of individuals will be the subject of third party discovery during this lawsuit, which Microsoft reserves the right to rely upon to corroborate the prior art status of the prior art identified herein. Microsoft reserves the right to modify, amend, or supplement these contentions if additional information becomes available during the course of this lawsuit. Furthermore, Microsoft will make all such devices, software programs, or other products in Microsoft's possession, custody, or control available for inspection by Plaintiff.

Consequently, Microsoft reserves the right to amend, modify, or supplement these Invalidity Contentions should additional information become available to them through discovery.

In addition, Microsoft identifies, responsive to Plaintiff's Infringement Contentions, the patents, publications, and/or systems listed in Tables 1-D, 2-D, and 3-D as evidence of the state of the art as it relates to the '883 Patent.

Each of these references show that the concepts of user authentication, proximity sensors, and location tracking are not novel and were well-known for many years before the alleged priority date of the '883 Patent, as discussed in more detail in Section VI.B (obviousness). In light of the references discussed herein, the Asserted Claims of the '883 Patent cannot be valid.

Each prior art patent, publication, or product identified above was either effectively filed or issued (for patents), published (for publications) or known, used, offered for sale or sold (for products) before either the earliest claimed priority date of the '883 Patent to which it is applied for invalidity, and none appears to have been abandoned, suppressed, or concealed, so each such reference also constitutes evidence of prior invention pursuant to 35 U.S.C. § 102(g), if it is in the

U.S. The persons or entities involved with each such invention include the named inventors on the above-identified patents, the authors listed on the above-identified publications, and the entities and individuals identified in connection with the above-identified products.

Because Microsoft has not yet completed discovery in this case, including taking depositions of the named inventors, and otherwise obtaining discovery from Plaintiff and third parties, Microsoft reserves the right to supplement these contentions with facts, documents, or other information learned at a later point through discovery or further investigation.

E. U.S. Patent No. 9,762,692

Invalidity claim charts identifying disclosures in the references identified in the tables below as to the Asserted Claims of the '692 Patent are provided in attached Exhibits E1 through E9.

Table 1-E: Prior Art Patents for the '692 Patent

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue/Publication	Filing Date
7,133,905 ("Dilley")	US	11/7/2006	4/9/2002
6,430,618 ("Karger")	US	8/6/2002	3/13/1998
2006/0015574 ("Seed")	US	1/19/2006	2/14/2002
2004/0221019 ("Swildens")	US	11/4/2004	4/29/2004
4,495,570 A ("Kitajima")	US	1985-01-22	1981-01-14
4,591,983 A ("Bennett")	US	1986-05-27	1984-07-09
4,594,704 A ("Ollivier")	US	1986-06-10	1983-11-23
4,726,017 A ("Krum")	US	1988-02-16	1985-05-21
4,803,641 A ("Hardy")	US	1989-02-07	1984-06-06
4,839,798 A ("Eguchi")	US	1989-06-13	1984-11-07
4,847,784 A ("Clancey")	US	1989-07-11	1987-07-13
4,920,432 A ("Eggers")	US	1990-04-24	1988-01-12
4,922,417 A ("Churm")	US	1990-05-01	1986-10-24
4,943,932 A ("Lark")	US	1990-07-24	1986-11-25
4,949,248 A ("Caro")	US	1990-08-14	1988-07-15
4,949,187 A ("Cohen")	US	1990-08-14	1988-12-16
5,029,232 A ("Nall")	US	1991-07-02	1989-01-12

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue/Publication	Filing Date
5,130,792 A (“Tindell”)	US	1992-07-14	1990-02-01
5,132,992 A (“Yurt”)	US	1992-07-21	1991-01-07
5,136,716 A (“Harvey”)	US	1992-08-04	1987-09-04
5,172,413 A (“Bradley”)	US	1992-12-15	1990-12-20
5,191,573 A (“Hair”)	US	1993-03-02	1988-06-13
5,253,341 A (“Rozmanith”)	US	1993-10-12	1991-03-04
2004/0003101 (“Roth”)	US	2004-01-01	2002-06-26
8,180,720 (“Kovacs”)	US	2012-05-15	2008-07-21

Table 2-E: Prior Art Publications for the '692 Patent

Title (Short Name)	Author	Publisher	Date of Publicati on
Andrew Tanenbaum, Computer Networks (3rd ed. 1996) (“Tanenbaum”)	Andrew Tanenbaum	PTR PH	Jan. 1, 1996
Tim Berners-Lee, The World-Wide Web, Communications of the ACM, Vol. 37, No. 8, August 1994 (“Berners-Lee”)	Tim Berners-Lee	Communi cations of the ACM	August 1994
Radhika Malpani, Making World Wide Web Caching Servers Cooperate, WWW4: Proceedings of the Fourth International Conference on World Wide Web, Pages 107-117, December 11, 1995 (“Malpani”)	Radhika Malpani	WWW4: Procedin gs of the Fourth Internatio nal Conferenc e on World Wide Web	Decembe r 11, 1995
Fielding et al., Hypertext Transfer Protocol – HTTP/1.1, RFC 2068, Jan. 1997, https://www.rfc-editor.org/rfc/rfc2068 (“HTTP/1.1”)	R. Fielding et al.	Network Working Group	January 1997
Barbara Tockey Zivkov et al., Disk Caching in Large Databases and Timeshared Systems, Proceedings Fifth International Symposium on Modeling, Analysis, and Simulation of Computer and	Barbara Tockey Zivkov et al.	University of California, Berkeley	January 1997

Title (Short Name)	Author	Publisher	Date of Publication
Telecommunication Systems (MASCOTS 97), Haifa, Israel, 184–95, January 1997 (“Zivkov”)			
Ronald J. Vetter et. al, Mosaic and the World-Wide Web, Computer, vol. 27, no. 10, pp. 49-57, Oct. 1994	Ronald J. Vetter et. al	Computer, vol. 27, no. 10	October 1994
Network Working Group, Request for Comments: 1034 (“RFC 1034”)	P. Mockapetris	Network Working Group	November 1987
Network Working Group, Request for Comments: 2181 (“RFC 2181”)	R. Elz et al.	Network Working Group	July 1997
The Cache System at the National Laboratory for Applied Network Research (“NLANR”)	National Laboratory for Applied Network Research	National Laboratory for Applied Network Research	March 7, 1997
A Hierarchical Internet Object Cache (“Chankhunthod”)	Anawat, Chankhunthod	USENIX 1996 TECHNICAL CONFERENCE, January.	1996
Fisher, Spinning the Web, Spring-Verlag New York, Inc. (1996) (“Fisher”)	Fisher	Spinner-Verlag New York, Inc.	1996
Ricart, Apache Server, Sams.net Publishing (1996) (“Ricart”)	Ricart	Sams.net Publishing	1996
S. Ramanathan and P. Rangan, “Architectures for Personalized Multimedia” IEEE Multimedia, 37-46 (Spring 1994)	S. Ramanathan and P. Rangan	IEEE Multimedia	May 31, 1994
Bradley, “Impact of the Internet and Online Services on Business Research Methods,” Massachusetts Institute of Technology (June 1996)	Bradley	Massachusetts Institute of Technology	June 1, 1996
Vahdat et al., “WebOS: Software Support for Scalable Web Services,” University of California – Berkeley	Vahdat et al	University of California	March 1, 1997

Title (Short Name)	Author	Publisher	Date of Publicati on
		— Berkeley	
Baentsch et al., “Enhancing the Web’s Infrastructure from Caching to Replication,” IEEE Internet Computing	Baentsch et al	IEEE Internet Computin g	April 30, 1997
Chawla and Bettati, “Replicating IP Services,” Technical Report 97-008, Texas A&M University (Sept. 1997)	Chawla and Bettati	Texas A&M University	Septembe r 1, 1997
Miller and Hollingsworth, “Using Content-Derived Names for Caching and Software Distribution,” University of Maryland	Miller and Hollingsworth	University of Maryland	October 15, 1998
Web Caching	Duane Wessels	O’REILL Y	2001
Lv, Z., Yang, et al., Multimedia Delivery Grid: A Novel Multimedia Delivery Scheme, In: Li, M., Sun, XH., Deng, Qn., Ni, J. (eds) GRID AND COOPERATIVE COMPUTING. GCC 2003. LECTURE NOTES IN COMPUTER SCIENCE, vol 3032. Springer, Berlin, Heidelberg (2004), https://doi.org/10.1007/978-3-540-24679-4_46	Berlin, Heidelberg	Springer	2004
https://web.archive.org/web/20060422163359/http://www.akamai.com/en/resources/pdf/brochures/S_B_esenterprise.pdf (“Akamai EdgeSuite Enterprise Edition Brochure”)	Akamai	Akamai	2006
Squid: The Definitive Guide	Duane Wessels	O’REILL Y	2004

Table 3-E: Prior Art Sales or Public Disclosures for the '692 Patent

Name of System or Invention	Identity of the Relevant Person or Entity	Date of Sale / Offer for Sale / Public Use / Known by Others
Server 2008 including Windows Media Services 9	Microsoft	April 25, 2007
Squid	Squid	January 2004
Akamai EdgeSuite	Akamai	At least by April 22, 2006

Table 4-E: Charts for the '692 Patent

Exhibit No.	Prior Art Reference/System
E1	U.S. Patent No. 7,133,905 ("Dilley")
E2	U.S. Patent No. 6,430,618 ("Karger")
E3	U.S. Pub. No. 2006/0015574 ("Seed")
E4	U.S. Pub. No. 2004/0221019 ("Swildens")
E5	Server 2008 including Windows Media Services 9
E6	Squid
E7	Akamai EdgeSuite
E8	U.S. Patent No. 8,180,720 ("Kovacs")
E9	U.S. Patent Pub. No. 2004/0003101 ("Roth")

Microsoft incorporates by reference the invalidity claim charts and contentions served by and Comcast in the *Sandpiper CDN, LLC v. Comcast Cable Comms., LLC d/b/a Xfinity et al*, 2:24-cv-00886 (E.D. Tex.) matter.

For each of the prior art devices identified above, Microsoft has listed and produced documents as evidence of the relevant features and functionality to the extent that they are presently available and have been located. Microsoft has obtained or is in the process of attempting to obtain the identified devices and will make them available for inspection to the extent they are obtained. To the extent that one or more documents (*e.g.*, user manual) may be used to describe aspects of a particular device, that device is a single reference for prior art purposes under 35 U.S.C. § 102. Some or all of the corroborating references may also separately qualify as prior art

publications under 35 U.S.C. § 102 and may be used as invalidating references under 35 U.S.C. §§ 102 and/or 103.

Further, Microsoft is actively searching for information regarding at least the systems, devices, and/or inventions listed above in Table 3-E.

Discovery is ongoing, and Microsoft may serve third parties with document subpoenas. One or more of these devices, along with related documentation, may be invalidating, and Microsoft reserves the right to supplement these contentions accordingly. Microsoft continues to investigate each of the identified devices, including through potential discovery needed to be obtained via third party subpoenas, and reserve the right to supplement these contentions and accompanying claim charts after further investigation.

Microsoft's reference to a particular product, device, or software program in these contentions should be interpreted as a reference to the system itself as well as to any corresponding patents, publications, or product literature relating to the cited system. Upon information and belief, the systems were publicly disclosed, used, sold, or offered for sale in the United States before the alleged priority date of the Asserted Claims of the '692 Patent. Microsoft's investigation of such prior art systems is still ongoing and discovery has not yet been received from third parties who may have information concerning such prior art systems. Accordingly, subsequent discovery may reveal information that affects the disclosures and contentions herein. For example, subsequent discovery may provide additional information regarding whether or not any of the third-party prior art systems anticipate or render obvious the Asserted Claims of the '692 Patent. As such, Microsoft reserves all rights to supplement their invalidity contentions.

Many of the inventive, research, design, and development activities concerning these systems and technologies occurred in the United States before the alleged priority date of the

Asserted Claims of the '692 Patent. Microsoft has obtained, and is in the process of obtaining, additional information regarding the dates by which the cited products and services were publicly disclosed, used, sold, or offered for sale, the circumstances under which the research, design, and development activities were conducted, and the identities of the particular individuals involved in such activities through publicly available patents, publications, and product literature. The actual dates, circumstances, and identities of individuals will be the subject of third party discovery during this lawsuit, which Microsoft reserves the right to rely upon to corroborate the prior art status of the prior art identified herein. Microsoft reserves the right to modify, amend, or supplement these contentions if additional information becomes available during the course of this lawsuit. Furthermore, Microsoft will make all such devices, software programs, or other products in Microsoft's possession, custody, or control available for inspection by Plaintiff.

Consequently, Microsoft reserves the right to amend, modify, or supplement these Invalidity Contentions should additional information become available to them through discovery.

In addition, Microsoft identifies, responsive to Plaintiff's Infringement Contentions, the patents, publications, and/or systems listed in Tables 1-E, 2-E, and 3-E as evidence of the state of the art as it relates to the '692 Patent.

Each of these references show that the concepts of user authentication, proximity sensors, and location tracking are not novel and were well-known for many years before the alleged priority date of the '692 Patent, as discussed in more detail in Section VI.B (obviousness). In light of the references discussed herein, the Asserted Claims of the '692 Patent cannot be valid.

Each prior art patent, publication, or product identified above was either effectively filed or issued (for patents), published (for publications) or known, used, offered for sale or sold (for products) before either the earliest claimed priority date of the '692 Patent to which it is applied

for invalidity, and none appears to have been abandoned, suppressed, or concealed, so each such reference also constitutes evidence of prior invention pursuant to 35 U.S.C. § 102(g), if it is in the U.S. The persons or entities involved with each such invention include the named inventors on the above-identified patents, the authors listed on the above-identified publications, and the entities and individuals identified in connection with the above-identified products.

Because Microsoft has not yet completed discovery in this case, including taking depositions of the named inventors, and otherwise obtaining discovery from Plaintiff and third parties, Microsoft reserves the right to supplement these contentions with facts, documents, or other information learned at a later point through discovery or further investigation.

F. U.S. Patent No. 10,701,173

Invalidity claim charts identifying disclosures in the references identified in the tables below as to the Asserted Claims of the '173 Patent are provided in attached Exhibits F1 through F15.

Table 1-F: Prior Art Patents for the '173 Patent

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
7,912,921 ("O'Rourke")	US	2011-03-22	2005-11-22
2000/181843 A	JP	2000-06-30	1998-12-18
2002/0198953	US	2002-12-26	2001-06-26
2003/0188021 A1 ("Challenger")	US	2003-10-02	2001-12-19
2003/0188009 A1 ("Agarwalla")	US	2003-10-02	2001-12-19
6,965,968 B1 ("Touboul")	US	2005-11-15	2003-02-27
2008/0065273 A1 ("Gerrity")	US	2008-03-13	2006-08-28
7,587,515 B2 ("Agarwalla")	US	2009-09-08	2001-12-19
7,730,154 B2 ("Agarwalla")	US	2010-06-01	2001-12-19
7,987,239 B2 * ("Agarwalla")	US	2011-07-26	2001-12-19
8,032,586 B2 * ("Challenger")	US	2011-10-04	2001-12-19
2012/0209942 A1 ("Zehavi")	WO	2012-08-16	2008-10-28
2013/0254343 A1 ("Stevens")	WO	2013-09-26	2012-03-22
2014/0129779 A1 * ("Lipstone")	US	2014-05-08	2012-11-06
2014/0215802A1 * ("Gonzalez")	US	2014-08-07	2008-11-14
8,812,695 B2 * ("Luna")	US	2014-08-19	2012-04-09

Patent No. (Primary Inventor/Short Name)	Country of Origin	Date of Issue	Filing Date
2014/0245359 A1 * (“De Foy”)	US	2014-08-28	2011-06-01
2014/0365683 A1 * (“Bergman”)	US	2014-12-11	2013-06-07
8,984,166 B2 * (“Graham-Cumming”)	US	2015-03-17	2012-04-05
2015/0334094 A1 * (“Suresh”)	US	2015-11-19	2014-05-15
2015/0347248 A1 * (“Ara jo”)	US	2015-12-03	2014-05-30
2016/0173634 A1 (“Newton”)	US	2016-06-16	2014-12-15
2018/0129613 A1 * (“Faldu”)	US	2018-05-10	2016-11-10
2008/0229025 (“Plamondon”)	US	2014-04-15	2007-03-12
2011/0060812 (“Middleton”)	US	2011-03-10	2010-07-14
2008/0201331 (“Eriksen”) A1 US8065275B2	US	2008-08-21 2011-11-12	2007-02-15 2007-02-15
2003/0187917 A1 (“Cohen”) US8650266 B2	US	2003-10-02 2014-02-11	2002-04-12 2002-04-12
2004/0003101 (“Roth”)	US	2004-01-01	2002-06-26
2002/0010798 (“Ben-Shaul”)	US	2002-01-24	2001-02-16
2014/0280493 (“Ortiz Rodriguez”)	US	2014-09-18	2013-02-24
2012/0096106 (“Blumofe”)	US	2012-04-19	2011-06-20
6,965,968 (“Touboul”)	US	2005-11-15	2003-02-27
2012/0005251 (“Xie”)	US	2012-01-05	2010-12-28

Table 2-F: Prior Art Publications for the ’173 Patent

Title (Short Name)	Author	Publisher	Date of Publication
Andrew Tanenbaum, Computer Networks (3rd ed. 1996) (“Tanenbaum”)	Andrew Tanenbaum	PTR PH	Jan. 1, 1996
Tim Berners-Lee, The World-Wide Web, Communications of the ACM, Vol. 37, No. 8, August 1994 (“Berners-Lee”)	Tim Berners-Lee	Communications of the ACM	August 1994
Radhika Malpani, Making World Wide Web Caching Servers Cooperate, WWW4: Proceedings of the Fourth International Conference on World Wide Web, Pages 107-117, December 11, 1995 (“Malpani”)	Radhika Malpani	WWW4: Proceedings of the Fourth International Conference on World Wide Web	December 11, 1995
Fielding et al., Hypertext Transfer Protocol – HTTP/1.1, RFC 2068, Jan. 1997, https://www.rfc-editor.org/rfc/rfc2068 (“HTTP/1.1”)	R. Fielding et al.	Network Working Group	January 1997

Title (Short Name)	Author	Publisher	Date of Publication
Barbara Tockey Zivkov et al., Disk Caching in Large Databases and Timeshared Systems, Proceedings Fifth International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS 97), Haifa, Israel, 184–95, January 1997 (“Zivkov”)	Barbara Tockey Zivkov et al.	University of California, Berkeley	January 1997
Ronald J. Vetter et. al, Mosaic and the World-Wide Web, Computer, vol. 27, no. 10, pp. 49-57, Oct. 1994	Ronald J. Vetter et. al	Computer, vol. 27, no. 10	October 1994
Network Working Group, Request for Comments: 1034 (“RFC 1034”)	P. Mockapetris	Network Working Group	November 1987
Network Working Group, Request for Comments: 2181 (“RFC 2181”)	R. Elz et al.	Network Working Group	July 1997
The Cache System at the National Laboratory for Applied Network Research (“NLNR”)	National Laboratory for Applied Network Research	National Laboratory for Applied Network Research	March 7, 1997
A Hierarchical Internet Object Cache (“Chankhunthod”)	Anawat, Chankhunthod	USENIX 1996 TECHNICAL CONFERENCE, January.	1996
Fisher, Spinning the Web, Springer-Verlag New York, Inc. (1996) (“Fisher”)	Fisher	Spinner-Verlag New York, Inc.	1996
Ricart, Apache Server, Sams.net Publishing (1996) (“Ricart”)	Ricart	Sams.net Publishing	1996
S. Ramanathan and P. Rangan, “Architectures for Personalized Multimedia” IEEE Multimedia, 37-46 (Spring 1994)	S. Ramanathan and P. Rangan	IEEE Multimedia	May 31, 1994
Bradley, “Impact of the Internet and Online Services on Business Research Methods,” Massachusetts Institute of Technology (June 1996)	Bradley	Massachusetts Institute of Technology	June 1, 1996
Vahdat et al., “WebOS: Software Support for Scalable Web Services,” University of California – Berkeley	Vahdat et al	University of California – Berkeley	March 1, 1997
Baentsch et al., “Enhancing the Web’s Infrastructure from Caching to	Baentsch et al	IEEE Internet Computing	April 30, 1997

Title (Short Name)	Author	Publisher	Date of Publication
Replication,” IEEE Internet Computing			
Chawla and Bettati, “Replicating IP Services,” Technical Report 97-008, Texas A&M University (Sept. 1997)	Chawla and Bettati	Texas A&M University	September 1, 1997
Miller and Hollingsworth, “Using Content- Derived Names for Caching and Software Distribution,” University of Maryland	Miller and Hollingsworth	University of Maryland	October 15, 1998

Table 3-F: Prior Art Sales or Public Disclosures for the ’173 Patent

Name of System or Invention	Identity of the Relevant Person or Entity	Date of Sale / Offer for Sale / Public Use / Known by Others
Server 2008 including Windows Media Services 9	Microsoft	April 25, 2007
Squid	Squid	January 2004
Akamai EdgeSuite	Akamai	At least by April 22, 2006

Table 4-F: Charts for the ’173 Patent

Exhibit No.	Prior Art Reference/System
F1	O’Rourke
F2	Eriksen
F3	Ben-Shaul
F4	Ortiz Rodriguez
F5	Blumofe
F6	Middleton
F7	Plamondon
F8	HTTP/1.1
F9	Cohen
F10	Touboul
F11	Xie
F12	Server 2008 including Windows Media Services 9

Exhibit No.	Prior Art Reference/System
F13	Squid
F14	Akamai EdgeSuite (“Akamai”)
F15	Roth

For each of the prior art devices identified above, Microsoft has listed and produced documents as evidence of the relevant features and functionality to the extent that they are presently available and have been located. Microsoft has obtained or is in the process of attempting to obtain the identified devices and will make them available for inspection to the extent they are obtained. To the extent that one or more documents (*e.g.*, user manual) may be used to describe aspects of a particular device, that device is a single reference for prior art purposes under 35 U.S.C. § 102. Some or all of the corroborating references may also separately qualify as prior art publications under 35 U.S.C. § 102 and may be used as invalidating references under 35 U.S.C. §§ 102 and/or 103.

Further, Microsoft is actively searching for information regarding at least the systems, devices, and/or inventions listed above in Table 3-F.

Discovery is ongoing, and Microsoft may serve third parties with document subpoenas. One or more of these devices, along with related documentation, may be invalidating, and Microsoft reserves the right to supplement these contentions accordingly. Microsoft continues to investigate each of the identified devices, including through potential discovery needed to be obtained via third party subpoenas, and reserve the right to supplement these contentions and accompanying claim charts after further investigation.

Microsoft’s reference to a particular product, device, or software program in these contentions should be interpreted as a reference to the system itself as well as to any corresponding patents, publications, or product literature relating to the cited system. Upon information and belief, the systems were publicly disclosed, used, sold, or offered for sale in the United States

before the alleged priority date of the Asserted Claims of the '173 Patent. Microsoft's investigation of such prior art systems is still ongoing and discovery has not yet been received from third parties who may have information concerning such prior art systems. Accordingly, subsequent discovery may reveal information that affects the disclosures and contentions herein. For example, subsequent discovery may provide additional information regarding whether or not any of the third-party prior art systems anticipate or render obvious the Asserted Claims of the '173 Patent. As such, Microsoft reserves all rights to supplement their invalidity contentions.

Many of the inventive, research, design, and development activities concerning these systems and technologies occurred in the United States before the alleged priority date of the Asserted Claims of the '173 Patent. Microsoft has obtained, and is in the process of obtaining, additional information regarding the dates by which the cited products and services were publicly disclosed, used, sold, or offered for sale, the circumstances under which the research, design, and development activities were conducted, and the identities of the particular individuals involved in such activities through publicly available patents, publications, and product literature. The actual dates, circumstances, and identities of individuals will be the subject of third party discovery during this lawsuit, which Microsoft reserves the right to rely upon to corroborate the prior art status of the prior art identified herein. Microsoft reserves the right to modify, amend, or supplement these contentions if additional information becomes available during the course of this lawsuit. Furthermore, Microsoft will make all such devices, software programs, or other products in Microsoft's possession, custody, or control available for inspection by Plaintiff.

Consequently, Microsoft reserves the right to amend, modify, or supplement these Invalidity Contentions should additional information become available to them through discovery.

In addition, Microsoft identifies, responsive to Plaintiff's Infringement Contentions, the patents, publications, and/or systems listed in Tables 1-F, 2-F, and 3-F as evidence of the state of the art as it relates to the '173 Patent.

Each of these references show that the concepts of user authentication, proximity sensors, and location tracking are not novel and were well-known for many years before the alleged priority date of the '173 Patent, as discussed in more detail in Section VI.B (obviousness). In light of the references discussed herein, the Asserted Claims of the '173 Patent cannot be valid.

Because Microsoft has not yet completed discovery in this case, including taking depositions of the named inventors, and otherwise obtaining discovery from Plaintiff and third parties, Microsoft reserves the right to supplement these contentions with facts, documents, or other information learned at a later point through discovery or further investigation.

V. PRIORITY DATE

A. U.S. Patent No. 8,478,903

In its Infringement Contentions, Sandpiper argues that the asserted claims of the '903 Patent are entitled to a priority date of at least February 10, 1998.

B. U.S. Patent No. 8,924,466

In its Infringement Contentions, Sandpiper argues that the asserted claims of the '466 Patent are entitled to a priority date of at least February 14, 2002.

C. U.S. Patent No. 9,456,053

In its Infringement Contentions, Sandpiper argues that the asserted claims of the '053 Patent are entitled to a priority date of at least December 14, 2011.

D. U.S. Patent No. 9,722,883

In its Infringement Contentions, Sandpiper argues that the asserted claims of the '883 Patent are entitled to a priority date of at least December 13, 2012.

A. U.S. Patent No. 9,762,692

In its Infringement Contentions, Sandpiper argues that the asserted claims of the '692 Patent are entitled to a priority date of at least April 4, 2008.

B. U.S. Patent No. 10,701,173

In its Infringement Contentions, Sandpiper argues that the asserted claims of the '173 Patent are entitled to a priority date of at least December 15, 2014.

VI. INVALIDITY UNDER 35 U.S.C. §§ 102 AND 103

Microsoft contends that certain prior art references below anticipate one or more Asserted Claims and that to the extent the identified prior art references do not anticipate the Asserted Claims, those claims are invalid as obvious under 35 U.S.C. §103. Each anticipatory prior art reference, either alone or in combination with other prior art, also renders the Asserted Claims invalid as obvious. In particular, each anticipatory prior art reference may be combined with (1) information generally known to persons skilled in the art at the time of the alleged invention, and/or (2) any of the other identified prior art references. To the extent that Plaintiff contends that any of the anticipatory prior art fails to disclose one or more limitations of the Asserted Claims, Microsoft contends that any difference between the reference and the corresponding patent claims would have been obvious to one of ordinary skill in the art. Thus, all anticipation charts should be interpreted as both reflecting anticipation by the reference as well as invalidity due to single reference obviousness, to the extent that Plaintiff contends that any limitation is missing.

A. Anticipation

As stated above, Microsoft incorporates by reference all other invalidity contentions related to the Asserted Patents served on or otherwise provided to Plaintiff, whether past or future. Prior art references anticipating the Asserted Claims are provided below. The prior art listed below anticipates the Asserted Claims under the proper construction of the claims and/or under Plaintiff's

apparent interpretation of the claims reflected by Plaintiff's Complaint and Infringement Contentions.

1. U.S. Patent No. 8,478,903

The Asserted Claims of the '903 Patent are anticipated and/or rendered obvious by prior art. Microsoft identifies the prior art references that anticipate or render obvious the '903 Patent Asserted Claims in the claim charts of Exhibits A1 through A8 (collectively "Appendix A") which are hereby incorporated by reference as if fully set forth herein. The claim charts of Appendix A show how these prior art references teach or suggest each and every element of the '903 Patent Asserted Claims. For each reference or combination of references suggested by each chart and/or herein, Microsoft indicates whether the prior art renders the claim anticipated and/or obvious.

In addition to contending that the '903 Patent Asserted Claims are invalid in view of the prior art references cited in the claim charts of Appendix A, Microsoft further contends that the '903 Patent Asserted Claims are invalid as anticipated and/or obvious under U.S.C. §§ 102(a), 102(b), and/or 103 in view of public knowledge and uses and/or offers for sale of products and services related to the subject matter of the cited references. As discovery is ongoing, Microsoft continues to investigate these items and to reserve the right to amend or supplement these contentions to include additional information or documents regarding such products and/or systems. In addition to contending that the '903 Patent Asserted Claims are invalid in view of the explicit disclosures of the prior art references cited in the claim charts of Appendix A, Microsoft further contends that the knowledge of skill evidenced by those references render those claims invalid. Microsoft further contends that, even if a prior art reference does not expressly disclose all limitations as arranged or combined in the same manner as an asserted claim, a skilled artisan reading the reference would at once envisage the claimed arrangement or combination. *See, e.g., Kennametal Inc. v. Ingersoll Cutting Tool Co.*, 780 F.3d 1376 (Fed. Cir. 2015).

Microsoft's reference to a particular system, software program, device, or product in the claim charts of Appendix A should be interpreted alternatively as both a reference to the product itself and to any corresponding patents, publications, or product literature cited in Appendix A that relates to the cited system, software program, device, or product. In addition, Microsoft may rely on other documents or things that have not yet been located to support its contentions regarding such prior art system, software program, device, or product referenced in the charts.

Microsoft incorporates by reference, as if set forth fully herein, all prior art cited during the prosecution of the '903 Patent and all prior art disclosed during previous litigation proceedings.

Microsoft further identifies and hereby incorporates by reference as if set forth fully herein the prior art references and invalidity contentions as described in any Other Sandpiper Proceedings wherein invalidity contentions have been, or will be, provided regarding the '903 Patent, its foreign counterparts, or any parent or child patent of the '903 Patent. Microsoft reserves the right to use any and all portions of the publication, related publications, commercial embodiments of the publication, and other evidence that is discovered in these lawsuits to demonstrate and/or evidence the components, functionality, and capabilities of the devices and systems disclosed in the references charted.

Where Microsoft identifies a particular figure in a prior art reference, the identification should be understood to encompass the caption and description of the figure, as well as any text relating to the figure in addition to the figure itself. Similarly, where an identified portion of text refers to a figure, embodiment, or other material, the identification should be understood to include the referenced figure, embodiment, or other material as well. Microsoft's citations to specific statements, figures, or disclosures in a prior art reference are sufficient to put Plaintiff on notice where each element of each asserted claim is found in the prior art. However, it should be

recognized that a person of ordinary skill in the art would generally read a prior art reference as a whole and in the context of other publications, literature, and general knowledge in the field. To understand and interpret any specific statement or disclosure in a prior art reference, a person of ordinary skill in the art would rely upon other information including other publications and general scientific or engineering knowledge. Microsoft therefore reserves the right to rely upon other unidentified portions of the prior art references and on other publications and expert testimony to provide context and to aid understanding and interpretation of the identified portions.

Microsoft also reserves the right to rely upon other portions of the prior art references, other publications, and the testimony of experts to establish that the alleged inventions would have been obvious to a person of ordinary skill in the art, including the basis of modifying or combining certain cited references. To the extent any limitation is deemed not to be exactly met by an item of prior art, then any purported differences are such that the claimed subject matter as a whole would have been obvious to one skilled in the art at the time of the alleged invention, in view of the state of the art and knowledge of those skilled in the art. To the extent that an element of an Asserted Claim is not anticipated, the claim is rendered obvious by combination with one or more other prior art references identified in Appendix A.

2. U.S. Patent No. 8,924,466

The Asserted Claims of the '466 Patent are anticipated and/or rendered obvious by prior art. Microsoft identifies the prior art references that anticipate or render obvious the '466 Patent Asserted Claims in the claim charts of Exhibits B1 through B13 (collectively "Appendix B") which are hereby incorporated by reference as if fully set forth herein. The claim charts of Appendix B show how these prior art references teach or suggest each and every element of the '466 Patent Asserted Claims. For each reference or combination of references suggested by each chart and/or herein, Microsoft indicates whether the prior art renders the claim anticipated and/or obvious.

In addition to contending that the '466 Patent Asserted Claims are invalid in view of the prior art references cited in the claim charts of Appendix B, Microsoft further contends that the '466 Patent Asserted Claims are invalid as anticipated and/or obvious under U.S.C. §§ 102(a), 102(b), and/or 103 in view of public knowledge and uses and/or offers for sale of products and services related to the subject matter of the cited references. As discovery is ongoing, Microsoft continues to investigate these items and to reserve the right to amend or supplement these contentions to include additional information or documents regarding such products and/or systems. In addition to contending that the '466 Patent Asserted Claims are invalid in view of the explicit disclosures of the prior art references cited in the claim charts of Appendix B, Microsoft further contends that the knowledge of skill evidenced by those references render those claims invalid. Microsoft further contends that, even if a prior art reference does not expressly disclose all limitations as arranged or combined in the same manner as an asserted claim, a skilled artisan reading the reference would at once envisage the claimed arrangement or combination. *See, e.g., Kennametal Inc. v. Ingersoll Cutting Tool Co.*, 780 F.3d 1376 (Fed. Cir. 2015).

Microsoft's reference to a particular system, software program, device, or product in the claim charts of Appendix B should be interpreted alternatively as both a reference to the product itself and to any corresponding patents, publications, or product literature cited in Appendix B that relates to the cited system, software program, device, or product. In addition, Microsoft may rely on other documents or things that have not yet been located to support its contentions regarding such prior art system, software program, device, or product referenced in the charts.

Microsoft incorporates by reference, as if set forth fully herein, all prior art cited during the prosecution of the '466 Patent and all prior art disclosed during previous litigation proceedings.

Microsoft further identifies and hereby incorporates by reference as if set forth fully herein the prior art references and invalidity contentions as described in any Other Sandpiper Proceedings wherein invalidity contentions have been, or will be, provided regarding the '466 Patent, its foreign counterparts, or any parent or child patent of the '466 Patent. Microsoft reserves the right to use any and all portions of the publication, related publications, commercial embodiments of the publication, and other evidence that is discovered in these lawsuits to demonstrate and/or evidence the components, functionality, and capabilities of the devices and systems disclosed in the references charted.

Where Microsoft identifies a particular figure in a prior art reference, the identification should be understood to encompass the caption and description of the figure, as well as any text relating to the figure in addition to the figure itself. Similarly, where an identified portion of text refers to a figure, embodiment, or other material, the identification should be understood to include the referenced figure, embodiment, or other material as well. Microsoft's citations to specific statements, figures, or disclosures in a prior art reference are sufficient to put Plaintiff on notice where each element of each asserted claim is found in the prior art. However, it should be recognized that a person of ordinary skill in the art would generally read a prior art reference as a whole and in the context of other publications, literature, and general knowledge in the field. To understand and interpret any specific statement or disclosure in a prior art reference, a person of ordinary skill in the art would rely upon other information including other publications and general scientific or engineering knowledge. Microsoft therefore reserves the right to rely upon other unidentified portions of the prior art references and on other publications and expert testimony to provide context and to aid understanding and interpretation of the identified portions.

Microsoft also reserves the right to rely upon other portions of the prior art references, other publications, and the testimony of experts to establish that the alleged inventions would have been obvious to a person of ordinary skill in the art, including the basis of modifying or combining certain cited references. To the extent any limitation is deemed not to be exactly met by an item of prior art, then any purported differences are such that the claimed subject matter as a whole would have been obvious to one skilled in the art at the time of the alleged invention, in view of the state of the art and knowledge of those skilled in the art. To the extent that an element of an Asserted Claim is not anticipated, the claim is rendered obvious by combination with one or more other prior art references identified in Appendix B.

3. U.S. Patent No. 9,456,053

The Asserted Claims of the '053 Patent are anticipated and/or rendered obvious by prior art. Microsoft identifies the prior art references that anticipate or render obvious the '053 Patent Asserted Claims in the claim charts of Exhibits C1 through C10 (collectively "Appendix C") which are hereby incorporated by reference as if fully set forth herein. The claim charts of Appendix C show how these prior art references teach or suggest each and every element of the '053 Patent Asserted Claims. For each reference or combination of references suggested by each chart and/or herein, Microsoft indicates whether the prior art renders the claim anticipated and/or obvious.

In addition to contending that the '053 Patent Asserted Claims are invalid in view of the prior art references cited in the claim charts of Appendix C, Microsoft further contends that the '053 Patent Asserted Claims are invalid as anticipated and/or obvious under U.S.C. §§ 102(a), 102(b), and/or 103 in view of public knowledge and uses and/or offers for sale of products and services related to the subject matter of the cited references. As discovery is ongoing, Microsoft continues to investigate these items and to reserve the right to amend or supplement these contentions to include additional information or documents regarding such products and/or

systems. In addition to contending that the '053 Patent Asserted Claims are invalid in view of the explicit disclosures of the prior art references cited in the claim charts of Appendix C, Microsoft further contends that the knowledge of skill evidenced by those references render those claims invalid. Microsoft further contends that, even if a prior art reference does not expressly disclose all limitations as arranged or combined in the same manner as an asserted claim, a skilled artisan reading the reference would at once envisage the claimed arrangement or combination. *See, e.g., Kennametal Inc. v. Ingersoll Cutting Tool Co.*, 780 F.3d 1376 (Fed. Cir. 2015).

Microsoft's reference to a particular system, software program, device, or product in the claim charts of Appendix C should be interpreted alternatively as both a reference to the product itself and to any corresponding patents, publications, or product literature cited in Appendix C that relates to the cited system, software program, device, or product. In addition, Microsoft may rely on other documents or things that have not yet been located to support its contentions regarding such prior art system, software program, device, or product referenced in the charts.

Microsoft incorporates by reference, as if set forth fully herein, all prior art cited during the prosecution of the '053 Patent and all prior art disclosed during previous litigation proceedings.

Microsoft further identifies and hereby incorporates by reference as if set forth fully herein the prior art references and invalidity contentions as described in any Other Sandpiper Proceedings wherein invalidity contentions have been, or will be, provided regarding the '053 Patent, its foreign counterparts, or any parent or child patent of the '053 Patent. Microsoft reserves the right to use any and all portions of the publication, related publications, commercial embodiments of the publication, and other evidence that is discovered in these lawsuits to demonstrate and/or evidence the components, functionality, and capabilities of the devices and systems disclosed in the references charted.

Where Microsoft identifies a particular figure in a prior art reference, the identification should be understood to encompass the caption and description of the figure, as well as any text relating to the figure in addition to the figure itself. Similarly, where an identified portion of text refers to a figure, embodiment, or other material, the identification should be understood to include the referenced figure, embodiment, or other material as well. Microsoft's citations to specific statements, figures, or disclosures in a prior art reference are sufficient to put Plaintiff on notice where each element of each asserted claim is found in the prior art. However, it should be recognized that a person of ordinary skill in the art would generally read a prior art reference as a whole and in the context of other publications, literature, and general knowledge in the field. To understand and interpret any specific statement or disclosure in a prior art reference, a person of ordinary skill in the art would rely upon other information including other publications and general scientific or engineering knowledge. Microsoft therefore reserves the right to rely upon other unidentified portions of the prior art references and on other publications and expert testimony to provide context and to aid understanding and interpretation of the identified portions.

Microsoft also reserves the right to rely upon other portions of the prior art references, other publications, and the testimony of experts to establish that the alleged inventions would have been obvious to a person of ordinary skill in the art, including the basis of modifying or combining certain cited references. To the extent any limitation is deemed not to be exactly met by an item of prior art, then any purported differences are such that the claimed subject matter as a whole would have been obvious to one skilled in the art at the time of the alleged invention, in view of the state of the art and knowledge of those skilled in the art. To the extent that an element of an Asserted Claim is not anticipated, the claim is rendered obvious by combination with one or more other prior art references identified in Appendix C.

4. U.S. Patent No. 9,722,883

The Asserted Claims of the '883 Patent are anticipated and/or rendered obvious by prior art. Microsoft identifies the prior art references that anticipate or render obvious the '883 Patent Asserted Claims in the claim charts of Exhibits D1 through D9 (collectively "Appendix D") which are hereby incorporated by reference as if fully set forth herein. The claim charts of Appendix D show how these prior art references teach or suggest each and every element of the '883 Patent Asserted Claims. For each reference or combination of references suggested by each chart and/or herein, Microsoft indicates whether the prior art renders the claim anticipated and/or obvious.

In addition to contending that the '883 Patent Asserted Claims are invalid in view of the prior art references cited in the claim charts of Appendix D, Microsoft further contends that the '883 Patent Asserted Claims are invalid as anticipated and/or obvious under U.S.C. §§ 102(a), 102(b), and/or 103 in view of public knowledge and uses and/or offers for sale of products and services related to the subject matter of the cited references. As discovery is ongoing, Microsoft continues to investigate these items and to reserve the right to amend or supplement these contentions to include additional information or documents regarding such products and/or systems. In addition to contending that the '883 Patent Asserted Claims are invalid in view of the explicit disclosures of the prior art references cited in the claim charts of Appendix D, Microsoft further contends that the knowledge of skill evidenced by those references render those claims invalid. Microsoft further contends that, even if a prior art reference does not expressly disclose all limitations as arranged or combined in the same manner as an asserted claim, a skilled artisan reading the reference would at once envisage the claimed arrangement or combination. *See, e.g., Kennametal Inc. v. Ingersoll Cutting Tool Co.*, 780 F.3d 1376 (Fed. Cir. 2015).

Microsoft's reference to a particular system, software program, device, or product in the claim charts of Appendix D should be interpreted alternatively as both a reference to the product

itself and to any corresponding patents, publications, or product literature cited in Appendix D that relates to the cited system, software program, device, or product. In addition, Microsoft may rely on other documents or things that have not yet been located to support its contentions regarding such prior art system, software program, device, or product referenced in the charts.

Microsoft incorporates by reference, as if set forth fully herein, all prior art cited during the prosecution of the '883 Patent and all prior art disclosed during previous litigation proceedings.

Microsoft further identifies and hereby incorporates by reference as if set forth fully herein the prior art references and invalidity contentions as described in any Other Sandpiper Proceedings wherein invalidity contentions have been, or will be, provided regarding the '883 Patent, its foreign counterparts, or any parent or child patent of the '883 Patent. Microsoft reserves the right to use any and all portions of the publication, related publications, commercial embodiments of the publication, and other evidence that is discovered in these lawsuits to demonstrate and/or evidence the components, functionality, and capabilities of the devices and systems disclosed in the references charted.

Where Microsoft identifies a particular figure in a prior art reference, the identification should be understood to encompass the caption and description of the figure, as well as any text relating to the figure in addition to the figure itself. Similarly, where an identified portion of text refers to a figure, embodiment, or other material, the identification should be understood to include the referenced figure, embodiment, or other material as well. Microsoft's citations to specific statements, figures, or disclosures in a prior art reference are sufficient to put Plaintiff on notice where each element of each asserted claim is found in the prior art. However, it should be recognized that a person of ordinary skill in the art would generally read a prior art reference as a whole and in the context of other publications, literature, and general knowledge in the field. To

understand and interpret any specific statement or disclosure in a prior art reference, a person of ordinary skill in the art would rely upon other information including other publications and general scientific or engineering knowledge. Microsoft therefore reserves the right to rely upon other unidentified portions of the prior art references and on other publications and expert testimony to provide context and to aid understanding and interpretation of the identified portions.

Microsoft also reserves the right to rely upon other portions of the prior art references, other publications, and the testimony of experts to establish that the alleged inventions would have been obvious to a person of ordinary skill in the art, including the basis of modifying or combining certain cited references. To the extent any limitation is deemed not to be exactly met by an item of prior art, then any purported differences are such that the claimed subject matter as a whole would have been obvious to one skilled in the art at the time of the alleged invention, in view of the state of the art and knowledge of those skilled in the art. To the extent that an element of an Asserted Claim is not anticipated, the claim is rendered obvious by combination with one or more other prior art references identified in Appendix D.

5. U.S. Patent No. 9,762,692

The Asserted Claims of the '692 Patent are anticipated and/or rendered obvious by prior art. Microsoft identifies the prior art references that anticipate or render obvious the '692 Patent Asserted Claims in the claim charts of Exhibits E1 through E9 (collectively "Appendix E") which are hereby incorporated by reference as if fully set forth herein. The claim charts of Appendix E show how these prior art references teach or suggest each and every element of the '692 Patent Asserted Claims. For each reference or combination of references suggested by each chart and/or herein, Microsoft indicates whether the prior art renders the claim anticipated and/or obvious.

In addition to contending that the '692 Patent Asserted Claims are invalid in view of the prior art references cited in the claim charts of Appendix E, Microsoft further contends that the

'692 Patent Asserted Claims are invalid as anticipated and/or obvious under U.S.C. §§ 102(a), 102(b), and/or 103 in view of public knowledge and uses and/or offers for sale of products and services related to the subject matter of the cited references. As discovery is ongoing, Microsoft continues to investigate these items and to reserve the right to amend or supplement these contentions to include additional information or documents regarding such products and/or systems. In addition to contending that the '692 Patent Asserted Claims are invalid in view of the explicit disclosures of the prior art references cited in the claim charts of Appendix E, Microsoft further contends that the knowledge of skill evidenced by those references render those claims invalid. Microsoft further contends that, even if a prior art reference does not expressly disclose all limitations as arranged or combined in the same manner as an asserted claim, a skilled artisan reading the reference would at once envisage the claimed arrangement or combination. *See, e.g., Kennametal Inc. v. Ingersoll Cutting Tool Co.*, 780 F.3d 1376 (Fed. Cir. 2015).

Microsoft's reference to a particular system, software program, device, or product in the claim charts of Appendix E should be interpreted alternatively as both a reference to the product itself and to any corresponding patents, publications, or product literature cited in Appendix E that relates to the cited system, software program, device, or product. In addition, Microsoft may rely on other documents or things that have not yet been located to support its contentions regarding such prior art system, software program, device, or product referenced in the charts.

Microsoft incorporates by reference, as if set forth fully herein, all prior art cited during the prosecution of the '692 Patent and all prior art disclosed during previous litigation proceedings.

Microsoft further identifies and hereby incorporates by reference as if set forth fully herein the prior art references and invalidity contentions as described in any Other Sandpiper Proceedings wherein invalidity contentions have been, or will be, provided regarding the '692 Patent, its foreign

counterparts, or any parent or child patent of the '692 Patent. Microsoft reserves the right to use any and all portions of the publication, related publications, commercial embodiments of the publication, and other evidence that is discovered in these lawsuits to demonstrate and/or evidence the components, functionality, and capabilities of the devices and systems disclosed in the references charted.

Where Microsoft identifies a particular figure in a prior art reference, the identification should be understood to encompass the caption and description of the figure, as well as any text relating to the figure in addition to the figure itself. Similarly, where an identified portion of text refers to a figure, embodiment, or other material, the identification should be understood to include the referenced figure, embodiment, or other material as well. Microsoft's citations to specific statements, figures, or disclosures in a prior art reference are sufficient to put Plaintiff on notice where each element of each asserted claim is found in the prior art. However, it should be recognized that a person of ordinary skill in the art would generally read a prior art reference as a whole and in the context of other publications, literature, and general knowledge in the field. To understand and interpret any specific statement or disclosure in a prior art reference, a person of ordinary skill in the art would rely upon other information including other publications and general scientific or engineering knowledge. Microsoft therefore reserves the right to rely upon other unidentified portions of the prior art references and on other publications and expert testimony to provide context and to aid understanding and interpretation of the identified portions.

Microsoft also reserves the right to rely upon other portions of the prior art references, other publications, and the testimony of experts to establish that the alleged inventions would have been obvious to a person of ordinary skill in the art, including the basis of modifying or combining certain cited references. To the extent any limitation is deemed not to be exactly met by an item

of prior art, then any purported differences are such that the claimed subject matter as a whole would have been obvious to one skilled in the art at the time of the alleged invention, in view of the state of the art and knowledge of those skilled in the art. To the extent that an element of an Asserted Claim is not anticipated, the claim is rendered obvious by combination with one or more other prior art references identified in Appendix E.

6. U.S. Patent No. 10,701,173

The Asserted Claims of the '173 Patent are anticipated and/or rendered obvious by prior art. Microsoft identifies the prior art references that anticipate or render obvious the '173 Patent Asserted Claims in the claim charts of Exhibits F1 through F15 (collectively "Appendix F") which are hereby incorporated by reference as if fully set forth herein. The claim charts of Appendix F show how these prior art references teach or suggest each and every element of the '173 Patent Asserted Claims. For each reference or combination of references suggested by each chart and/or herein, Microsoft indicates whether the prior art renders the claim anticipated and/or obvious.

In addition to contending that the '173 Patent Asserted Claims are invalid in view of the prior art references cited in the claim charts of Appendix F, Microsoft further contends that the '173 Patent Asserted Claims are invalid as anticipated and/or obvious under U.S.C. §§ 102(a)(1), 102(a)(2), and/or 103 in view of public knowledge and uses and/or offers for sale of products and services related to the subject matter of the cited references. As discovery is ongoing, Microsoft continues to investigate these items and to reserve the right to amend or supplement these contentions to include additional information or documents regarding such products and/or systems. In addition to contending that the '173 Patent Asserted Claims are invalid in view of the explicit disclosures of the prior art references cited in the claim charts of Appendix F, Microsoft further contends that the knowledge of skill evidenced by those references render those claims invalid. Microsoft further contends that, even if a prior art reference does not expressly disclose

all limitations as arranged or combined in the same manner as an asserted claim, a skilled artisan reading the reference would at once envisage the claimed arrangement or combination. *See, e.g., Kennametal Inc. v. Ingersoll Cutting Tool Co.*, 780 F.3d 1376 (Fed. Cir. 2015).

Microsoft's reference to a particular system, software program, device, or product in the claim charts of Appendix F should be interpreted alternatively as both a reference to the product itself and to any corresponding patents, publications, or product literature cited in Appendix F that relates to the cited system, software program, device, or product. In addition, Microsoft may rely on other documents or things that have not yet been located to support its contentions regarding such prior art system, software program, device, or product referenced in the charts.

Microsoft incorporates by reference, as if set forth fully herein, all prior art cited during the prosecution of the '173 Patent and all prior art disclosed during previous litigation proceedings.

Microsoft further identifies and hereby incorporates by reference as if set forth fully herein the prior art references and invalidity contentions as described in any Other Sandpiper Proceedings wherein invalidity contentions have been, or will be, provided regarding the '173 Patent, its foreign counterparts, or any parent or child patent of the '173 Patent. Microsoft reserves the right to use any and all portions of the publication, related publications, commercial embodiments of the publication, and other evidence that is discovered in these lawsuits to demonstrate and/or evidence the components, functionality, and capabilities of the devices and systems disclosed in the references charted.

Where Microsoft identifies a particular figure in a prior art reference, the identification should be understood to encompass the caption and description of the figure, as well as any text relating to the figure in addition to the figure itself. Similarly, where an identified portion of text refers to a figure, embodiment, or other material, the identification should be understood to include

the referenced figure, embodiment, or other material as well. Microsoft's citations to specific statements, figures, or disclosures in a prior art reference are sufficient to put Plaintiff on notice where each element of each asserted claim is found in the prior art. However, it should be recognized that a person of ordinary skill in the art would generally read a prior art reference as a whole and in the context of other publications, literature, and general knowledge in the field. To understand and interpret any specific statement or disclosure in a prior art reference, a person of ordinary skill in the art would rely upon other information including other publications and general scientific or engineering knowledge. Microsoft therefore reserves the right to rely upon other unidentified portions of the prior art references and on other publications and expert testimony to provide context and to aid understanding and interpretation of the identified portions.

Microsoft also reserves the right to rely upon other portions of the prior art references, other publications, and the testimony of experts to establish that the alleged inventions would have been obvious to a person of ordinary skill in the art, including the basis of modifying or combining certain cited references. To the extent any limitation is deemed not to be exactly met by an item of prior art, then any purported differences are such that the claimed subject matter as a whole would have been obvious to one skilled in the art at the time of the alleged invention, in view of the state of the art and knowledge of those skilled in the art. To the extent that an element of an Asserted Claim is not anticipated, the claim is rendered obvious by combination with one or more other prior art references identified in Appendix F.

B. Obviousness

1. U.S. Patent No. 8,478,903

Microsoft identifies in Appendix A the prior art references that render obvious the Asserted Claims of the '903 Patent and include below exemplary combinations showing the obviousness of the '903 Patent Asserted Claims in view of the prior art. To the extent Plaintiff contends that an

element is not disclosed in any one of the anticipatory references described in Appendix A, the limitation would have been obvious in light of the disclosures within the reference and the knowledge of one of ordinary skill in the art at the time of the '903 Patent. Moreover, to the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix A, such reference may be combined with any other references listed in Appendix A for such element, thereby rendering the claims invalid for obviousness. To the extent a finder of fact determines that a limitation of any of the '903 Patent Asserted Claims is not disclosed by one of the references identified above, the claim is nevertheless unpatentable as obvious because they contain nothing that constitutes a patentable innovation. To the extent a finder of fact determines that a limitation of the '903 Patent Asserted Claims is not anticipated, it does not go beyond combining familiar elements according to known methods to achieve predictable results or does more than choose between clear alternatives known to those of ordinary skill in the art.

a. Obviousness

For at least the reasons described below, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix A with any other reference or references listed in Appendix A along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '903 Patent. For example, and without limitation, the Asserted Claims of the '903 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
Microsoft Proxy Server	• Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server)

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
Squid Cache	• Microsoft Proxy Server • Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
Microsoft Internet Information Server 4.0 (Windows NT Server)	• Microsoft Proxy Server • Squid Cache • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller)

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
WO 1996/41285 (“Kenner”)	• Microsoft Proxy Server • Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLNR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
U.S. 6,243,760 (“Armbruster”)	• Microsoft Proxy Server • Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLNR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
Network Working Group, Request for Comments: 1034 (“RFC 1034”)	• Microsoft Proxy Server • Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”)

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
Network Working Group, Request for Comments: 2181 (“RFC 2181”)	• Microsoft Proxy Server • Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
The Cache System at the National Laboratory for Applied Network Research (“NLANR”)	• Microsoft Proxy Server • Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
A Hierarchical Internet Object Cache (“Chankhunthod”)	• Microsoft Proxy Server • Squid Cache

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
U.S. 5,475,819 (Miller)	• Microsoft Proxy Server • Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,649,103 (Datta) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
U.S. 5,649,103 (Datta)	• Microsoft Proxy Server • Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”)

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • GB 2,281,793 (Wallis) • The knowledge of a person of ordinary skill in the art.
GB 2,281,793 (Wallis)	• Microsoft Proxy Server • Squid Cache • Microsoft Internet Information Server 4.0 (Windows NT Server) • WO 1996/41285 (“Kenner”) • U.S. 6,243,760 (“Armbruster”) • Network Working Group, Request for Comments: 1034 (“RFC 1034”) • Network Working Group, Request for Comments: 2181 (“RFC 2181”) • The Cache System at the National Laboratory for Applied Network Research (“NLANR”) • A Hierarchical Internet Object Cache (“Chankhunthod”) • U.S. 5,475,819 (Miller) • U.S. 5,649,103 (Datta) • The knowledge of a person of ordinary skill in the art.

b. General Motivations to Combine

Obviousness: For at least the reasons described in these contentions, it would have been obvious to one of ordinary skill in the art to combine any of a number of prior art references, including any combination of those prior art references identified in Appendix A along with the knowledge of one of ordinary skill in the art to meet the limitations of the ‘903 Patent Asserted Claims. As a result, the references identified herein render one or more Asserted Claims of the ‘903 Patent obvious when the references are read in combination with each other, and/or when read in view of the state of the art and knowledge of those skilled in the art. Each and every reference identified is also relevant to the state of the art at the time of the alleged invention. Any of the references disclosed above may be combined with one another to render obvious (and therefore invalid) each of Sandpiper’s Asserted Claims. Defendant may rely upon a subset of the

identified references or all of the references identified herein, including all references in Appendix A, for purposes of obviousness depending on the Court's claim construction, positions taken by Sandpiper during this litigation, and further investigation and discovery.

Reservation of Rights: Moreover, as mentioned above, Defendant has not yet completed their search or discovery concerning additional prior art. As such, Defendant's inclusion of exemplary combinations does not preclude them from identifying other invalidating combinations as appropriate, and Defendant reserves the right to identify additional specific combinations as well as to detail and explain such combinations.

No More Than Ordinary Variations of Prior Art; No Unexpected Results: To the extent not anticipated, the '903 Patent Asserted Claims represent no more than the result of ordinary variations of the prior art. Defendant further believes that no showing of a specific motivation to combine prior art is required to combine the references disclosed above and in the attached charts, as each combination of art would have no unexpected results, and at most would simply represent a known alternative to one of ordinary skill in the art. *See KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 415-16 (2007) (rejecting the Federal Circuit's "rigid" application of the teaching, suggestion, or motivation to combine test, instead espousing an "expansive and flexible" approach); *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1058 (Fed. Cir. 2016); *Intercontinental Great Brands LLC v. Kellogg N. Am. Co.*, 869 F.3d 1336, 1344 (Fed. Cir. 2017). Indeed, the Supreme Court held that a person of ordinary skill in the art is "a person of ordinary creativity, not an automaton" and "in many cases a person of ordinary skill in the art will be able to fit the teachings of multiple patents together like pieces of a puzzle." *Id.* at 420-21. Nevertheless, in addition to the information contained elsewhere in these contentions, Defendant identifies motivation and reason to combine the cited art.

Predictable Results; Obvious to Try: One or more combinations of the prior art references identified in Appendix A would have been obvious because these references would have been combined using: known methods to yield predictable results; known techniques in the same way; a simple substitution of one known, equivalent element for another to obtain predictable results; and/or a teaching, suggestion, or motivation in the prior art generally. In addition, it would have been obvious to try combining the prior art references identified above because there were only a finite number of predictable solutions and/or because known work in one field of endeavor prompted variations based on predictable design incentives and/or market forces either in the same field or a different one. Further, the combinations of the prior art references identified in Appendix A would have been obvious because the combinations represent known potential options with a reasonable expectation of success.

Interrelated Teachings: Additional evidence that there would have been a motivation to combine the prior art references identified above includes the interrelated teachings of multiple prior art references; common authorship; the effects of demands known to the design community or present in the marketplace; the existence of a known problem for which there was an obvious solution encompassed by the '903 Patent Asserted Claims; the existence of a known need or problem in the field of the endeavor at the time of the alleged invention(s); and the background knowledge that would have been possessed by a person having ordinary skill in the art.

Summary of Motivations to Combine: Thus, the motivation to combine the teachings of the prior art references disclosed in Appendix A is found in the references themselves and also in: (1) the nature of the problem being solved; (2) the express, implied and inherent teachings of the prior art; (3) the knowledge of persons of ordinary skill in the art; (4) the predictable results obtained in combining the different elements of the prior art; (5) the predictable results obtained

in simple substitution of one known element for another; (6) the use of a known technique to improve similar devices, methods, or products in the same way; (7) the predictable results obtained in applying a known technique to a known device, method, or product ready for improvement; (8) the finite number of identified predictable solutions that had a reasonable expectation of success; and (9) known work in various technological fields that could be applied to the same or different technological fields based on design incentives or other market forces.

Same Area of Technology: Additionally, it would be obvious to one of ordinary skill in the art to consult and/or combine any of the prior art listed in Appendix A because all of these references relate to the same area of technology and/or are from analogous art. The '903 Patent Asserted Claims are directed to “[a] system in which a plurality of content providers provide multiple resources to multiple clients, wherein each content provider provides at least some resources via one or more content sources associated with that content provider.” '903 Patent at Abstract. The '903 Patent Asserted Claims merely unite old elements, well known in the field, with no change in their respective function or result. Given the interrelated teachings of the prior art, the effects of demands known to the design community or present in the marketplace, and the background knowledge possessed by a person having ordinary skill in the art, it would have been obvious for one of ordinary skill in the art to combine these familiar elements, disclosed and/or embodied in the prior art listed above to practice the '903 Patent Asserted Claims.

Same Technical Issues: A person of ordinary skill would have been motivated to combine the above prior art based on his/her knowledge, the nature of the problem to be solved, and the teachings of the prior art. The identified prior art addresses the same or similar technical issues and suggests the same or similar solutions to those issues. Moreover, some of the prior art refer to or discuss other prior art, illustrating the close technical relationship among the prior art. For

the '903 Patent, one of ordinary skill in the art would have been motivated to combine known prior art solutions described in these references relating to a method in a content delivery network that uses a shared repeater server to deliver content on behalf of an origin server, and to associate the repeater server with alias names and a table listing origin servers. For example, there was a need to address bandwidth growth needs and problems to help increase business productivity, and allow for increasing scalability and reliability. There was also a need for companies to improve network performance, including by reducing network resource consumption.

Demand and Market Forces: For example, market forces suggested a demand for increased security and versatility of user authentication systems, and thus one of skill in the art would have thought to combine or modify references that described known methods that one of skill in the art would have recognized as offering improvements to solutions of that time.

Known Elements; Obvious to Try: Moreover, many of the claim elements were already known as admitted by the applicants in the patent specification. These elements represented design choices available to a person of ordinary skill. When there is a design need or market pressure to solve a problem such as identified previously and/or described in the '903 Patent, and there are a finite number of identified, predictable solutions, a person of ordinary skill would be motivated to combine the known options that are within his or her technical grasp. For the '903 Patent, there are a finite number of approaches for expanding the capabilities of proximity systems and location tracking systems to run or interact with different applications. For example, as of the priority date of the '903 Patent, a POSITA would have been very familiar with basic elements of the '903 Patent's claims such as shared repeater servers, alias names, and tables listing origin servers having content located thereon. *See Appx. A.* Such elements were within the knowledge of a POSITA,

whether or not disclosed in any particular reference, and the inclusion of such elements in the '903 Patent's claims does not render them novel or non-obvious.

c. Exemplary Combinations

For at least the reasons described above, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix A with any other reference or references listed in Appendix A along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '903 Patent.

For example, and without limitation, the Asserted Claims of the '903 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

(1) Kenner in view of Armbruster

It would have been obvious to combine Kenner and Armbruster because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Kenner describes a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. Specifically, Kenner discloses a distributed digital video clip delivery system, whereby video clips, stored locally or at remote locations, can be requested by an authorized client user and retrieved at the user's terminal. *See, e.g.*, Kenner at Abstract. Kenner's system involves origin servers (source IMs), and repeaters (primary or regional IMs), where the repeaters have extended storage and retrieval units for storing resources requested by clients. *Id.* at 10:18-31, 31:19-21, 31:30-32, 32:14-17, 44:6-14. The resources (video clips) are uploaded to the IMs by content providers, and in Kenner's scalable system, many IMs can be shared by multiple such providers, servicing multiple users. *See, e.g., id.* at 5:21-26, 40:2-28,

54:12-16. The IMs include clip databases, identifying the clips stored in their extended SRUs by alias names called “video IDs.” *Id.* at 34:3-6, 37:20-22.

Like Kenner, Armbruster also teaches a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. *See, e.g.*, Armbruster at 4:27-35. Specifically, Armbruster discloses a Managed Distributed Caching (MDC) system, in which content providers can upload resources from origin servers to local or centralized caches (repeaters). *See id.*; *see also* Figs. 1-3. Clients that are authorized can request resources uploaded by content providers, and those requests are directed to specific repeaters via alias names associated with those repeaters, as well as the content provider associated with the requested resource, and a lookup table identifying repeaters by their respective aliases. *See, e.g., id.* at Fig. 3, 4:27-52, 6:5-40 (Scenarios 2 and 3). Armbruster’s system can also accommodate multiple users, multiple content providers, and multiple repeater servers. *Id.*

A POSITA would have been motivated to combine Armbruster’s system in which alias names are associated with specific repeaters (caches), with Kenner’s system in which alias names identify content provider and content, to expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems.

A POSITA would have been motivated to combine Kenner, which describes a CDN system in which alias names identify specific content providers and content (*see generally* Exhibit A-02-[Kenner]), with Armbruster, which describes a CDN architecture in which alias names are used to direct client request to specific shared repeaters (caches), based on a lookup table matching alias names to repeaters (*see, e.g.*, Exhibit A-01-[Armbruster], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b],

37[a][i]-[ii], 37[c]-[e]). A POSITA would have been motivated to add the functionality of Armbruster's alias names and router lookup tables to expand the capabilities of the combined system for numerous reasons, such as to permit content providers to store different types of content at the most appropriate repeaters, and direct client requests for such different types of content to those respective repeaters. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems.

(2) Kenner in view of RFC1034

It would have been obvious to combine Kenner and RFC1034 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Kenner describes a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. Specifically, Kenner discloses a distributed digital video clip delivery system, whereby video clips, stored locally or at remote locations, can be requested by an authorized client user and retrieved at the user's terminal. *See, e.g.*, Kenner at Abstract. Kenner's system involves origin servers (source IMs), and repeaters (primary or regional IMs), where the repeaters have extended storage and retrieval units for storing resources requested by clients. *Id.* at 10:18-31, 31:19-21, 31:30-32, 32:14-17, 44:6-14. The resources (video clips) are uploaded to the IMs by content providers, and in Kenner's scalable system, many IMs can be shared by multiple such providers, servicing multiple users. *See, e.g., id.* at 5:21-26, 40:2-28, 54:12-16. The IMs include clip databases, identifying the clips stored in their extended SRUs by alias names called "video IDs." *Id.* at 34:3-6, 37:20-22.

Like Kenner, RFC1034 also describes a system in which multiple users can request resources from multiple distributed caches. *See, e.g.*, RFC1034 at § 2.3 (“The domain system defines procedures for accessing the data and for referrals to other name servers. The domain system also defines procedures for caching retrieved data and for periodic refreshing of data defined by the system administrator.”) RFC1034 also describes the use of alias names for identifying specific name servers (i.e., through a system of aliases and canonical names (CNAMEs)), and use of a lookup table to direct requests to servers corresponding to those alias names. *Id.* at §§ 3.6.2, 5.2.2.

A POSITA would have been motivated to combine Kenner, which describes a CDN system in which alias names identify specific content providers and content (*see generally* Exhibit A-02[Kenner]), with RFC1034, in which specific servers can be associated with multiple aliases, and in which requests are directed to servers based on a lookup table matching them with specific aliases (*see, e.g.*, Exhibit A-04 [RFC1034], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[b]-[e]). A POSITA would have been motivated to add the functionality of RFC1034’s aliases and canonical names (CNAMEs) to expand the capabilities of the combined system for numerous reasons, such as, among other reasons, to enable customization of the domains that clients use and see. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems.

(3) Kenner in view of NLANR

It would have been obvious to combine Kenner and NLANR because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Kenner describes a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. Specifically, Kenner discloses a distributed digital video clip delivery system, whereby video clips, stored locally or at remote locations, can be requested by an authorized client user and retrieved at the user's terminal. *See, e.g.*, Kenner at Abstract. Kenner's system involves origin servers (source IMs), and repeaters (primary or regional IMs), where the repeaters have extended storage and retrieval units for storing resources requested by clients. *Id.* at 10:18-31, 31:19-21, 31:30-32, 32:14-17, 44:6-14. The resources (video clips) are uploaded to the IMs by content providers, and in Kenner's scalable system, many IMs can be shared by multiple such providers, servicing multiple users. *See, e.g., id.* at 5:21-26, 40:2-28, 54:12-16. The IMs include clip databases, identifying the clips stored in their extended SRUs by alias names called "video IDs." *Id.* at 34:3-6, 37:20-22.

Like Kenner, NLANR also describes a CDN in which multiple shared repeater servers could service requests from multiple users, for content uploaded by multiple content providers. *See* NLANR at 1 (describing project background as providing a "solution to the problem of scalable wide area information dissemination"); *id.* at 2 (describing "NLANR Cache Architecture" as including six large caches distributed geographically, configured to distribute loads so that each handles a subset of top-level-domains, and each handling resources from certain geographic regions); *see also id.* at 5 (NLANR Cache Configuration map, showing location of geographic caches and regional domains and objects intended to be directed to associated caches). NLANR also discloses a system in which requests are directed to specific repeaters (caches) via alias names (geographic domains) intended to be associated with a given repeater, as provided in a text-based lookup table. *See id.*; *see also, e.g., id.* at 6 (text file of exemplary access list). The NLANR system

is likewise configured to serve many users, requesting resources from many content providers, in which multiple content providers share each of the many repeaters. *Id.*

A POSITA would have been motivated to combine Kenner, which describes a CDN system in which alias names identify specific content providers and content (*see generally* Exhibit A-02 [Kenner]), with NLANR, which describes a CDN architecture in which requests are directed to shared repeaters based on alias names associated with appropriate geographic regions, based on a lookup table assigning certain names to specific repeaters (*see, e.g.,* Exhibit A-03 [NLANR], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[b]-[e]). A POSITA would have been motivated to add the functionality of NLANR's system to Kenner to expand the capabilities of the combined system, and particularly to allow for an even larger, scalable CDN that could more efficiently distribute content across an even larger client base. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems.

(4) Microsoft Proxy Server in view of Kenner

It would have been obvious to combine Microsoft Proxy Server and Kenner because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft Proxy Server routes requests and responses between the Internet and client computers, acting as a liaison between them. In addition to routing requests, Proxy Server provides a cache of frequently requested Internet sites, blocks access to specified sites, and provides secure access between an internal network and the Internet. Proxy Server allows users to set up distributed content caching among multiple Proxy Server computers, using array-based content caching and/or chain-based content caching. Proxy Server can also use reverse proxying and reverse

hosting to send requests to Web publishing servers connected behind the Proxy Server computer. *See, e.g.,* ProxyServer2.pdf at 1, 10, 12-13. Proxy Server also implements routing algorithms that include using an array membership table and computing hashes on the Proxy Server array member names. *See, e.g., id.* at 87-89.

Kenner describes a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. Specifically, Kenner discloses a distributed digital video clip delivery system, whereby video clips, stored locally or at remote locations, can be requested by an authorized client user and retrieved at the user's terminal. *See, e.g.,* Kenner at Abstract. Kenner's system involves origin servers (source IMs), and repeaters (primary or regional IMs), where the repeaters have extended storage and retrieval units for storing resources requested by clients. *Id.* at 10:18-31, 31:19-21, 31:30-32, 32:14-17, 44:6-14. The resources (video clips) are uploaded to the IMs by content providers, and in Kenner's scalable system, many IMs can be shared by multiple such providers, servicing multiple users. *See, e.g., id.* at 5:21-26, 40:2-28, 54:12-16. The IMs include clip databases, identifying the clips stored in their extended SRUs by alias names called "video IDs." *Id.* at 34:3-6, 37:20-22.

A POSITA would have been motivated to combine Microsoft Proxy Server's distributed caching, proxy chaining, and alias-based routing (*see, e.g.,* Exhibit A-05 [Microsoft Proxy Server], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with Kenner's system in which alias names identify content provider and content (*see, e.g.,* Exhibit A-02 [Kenner], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]) to expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(5) Microsoft Proxy Server in view of Armbruster

It would have been obvious to combine Microsoft Proxy Server and Armbruster because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft Proxy Server routes requests and responses between the Internet and client computers, acting as a liaison between them. In addition to routing requests, Proxy Server provides a cache of frequently requested Internet sites, blocks access to specified sites, and provides secure access between an internal network and the Internet. Proxy Server allows users to set up distributed content caching among multiple Proxy Server computers, using array-based content caching and/or chain-based content caching. Proxy Server can also use reverse proxying and reverse hosting to send requests to Web publishing servers connected behind the Proxy Server computer. *See, e.g.,* ProxyServer2.pdf at 1, 10, 12-13. Proxy Server also implements routing algorithms that include using an array membership table and computing hashes on the Proxy Server array member names. *See, e.g., id.* at 87-89.

Armbruster also teaches a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. *See, e.g.,* Armbruster at 4:27-35. Specifically, Armbruster discloses a Managed Distributed Caching (MDC) system, in which content providers can upload resources from origin servers to local or centralized caches (repeaters). *See id.; see also* Figs. 1-3. Clients that are authorized can request resources uploaded by content providers, and those requests are directed to specific repeaters via alias names associated with those repeaters, as well as the content provider associated with the requested resource, and a lookup table identifying repeaters by their respective aliases. *See, e.g., id.* at Fig. 3, 4:27-52, 6:5-40 (Scenarios 2 and 3).

Armbruster's system can also accommodate multiple users, multiple content providers, and multiple repeater servers. *Id.*

A POSITA would have been motivated to combine Microsoft Proxy Server's distributed caching, proxy chaining, and alias-based routing (*see generally* Exhibit A-05 [Microsoft Proxy Server], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with Armbruster's system in which alias names are associated with specific repeaters (caches) (*see, e.g.*, Exhibit A-01 [Armbruster], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]) to expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(6) Microsoft Proxy Server in view of RFC1034

It would have been obvious to combine Microsoft Proxy Server and RFC1034 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft Proxy Server routes requests and responses between the Internet and client computers, acting as a liaison between them. In addition to routing requests, Proxy Server provides a cache of frequently requested Internet sites, blocks access to specified sites, and provides secure access between an internal network and the Internet. Proxy Server allows users to set up distributed content caching among multiple Proxy Server computers, using array-based content caching and/or chain-based content caching. Proxy Server can also use reverse proxying and reverse hosting to send requests to Web publishing servers connected behind the Proxy Server computer. *See, e.g.*, ProxyServer2.pdf at 1, 10, 12-13. Proxy Server also implements routing algorithms that

include using an array membership table and computing hashes on the Proxy Server array member names. *See, e.g., id.* at 87-89.

RFC1034 also describes a system in which multiple users can request resources from multiple distributed caches. *See, e.g.,* RFC1034 at § 2.3 (“The domain system defines procedures for accessing the data and for referrals to other name servers. The domain system also defines procedures for caching retrieved data and for periodic refreshing of data defined by the system administrator.”) RFC1034 also describes the use of alias names for identifying specific name servers (i.e., through a system of aliases and canonical names (CNAMEs)), and use of a lookup table to direct requests to servers corresponding to those alias names. *Id.* at §§ 3.6.2, 5.2.2.

A POSITA would have been motivated to combine Microsoft Proxy Server’s distributed caching, proxy chaining, and alias-based routing (*see generally* Exhibit A-05 [Microsoft Proxy Server], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with the alias and CNAME system described by RFC1034, in which specific servers can be associated with multiple aliases (*see, e.g.,* Exhibit A-04 [RFC1034], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]) to expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(7) Microsoft Proxy Server in view of NLANR

It would have been obvious to combine Microsoft Proxy Server and NLANR because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft Proxy Server routes requests and responses between the Internet and client computers, acting as a liaison between them. In addition to routing requests, Proxy Server provides

a cache of frequently requested Internet sites, blocks access to specified sites, and provides secure access between an internal network and the Internet. Proxy Server allows users to set up distributed content caching among multiple Proxy Server computers, using array-based content caching and/or chain-based content caching. Proxy Server can also use reverse proxying and reverse hosting to send requests to Web publishing servers connected behind the Proxy Server computer. *See, e.g.,* ProxyServer2.pdf at 1, 10, 12-13. Proxy Server also implements routing algorithms that include using an array membership table and computing hashes on the Proxy Server array member names. *See, e.g., id.* at 87-89.

NLANR also describes a CDN in which multiple shared repeater servers could service requests from multiple users, for content uploaded by multiple content providers. *See* NLANR at 1 (describing project background as providing a “solution to the problem of scalable wide area information dissemination”); *id.* at 2 (describing “NLANR Cache Architecture” as including six large caches distributed geographically, configured to distribute loads so that each handles a subset of top-level-domains, and each handling resources from certain geographic regions); *see also id.* at 5 (NLANR Cache Configuration map, showing location of geographic caches and regional domains and objects intended to be directed to associated caches). NLANR also discloses a system in which requests are directed to specific repeaters (caches) via alias names (geographic domains) intended to be associated with a given repeater, as provided in a text-based lookup table. *See id.; see also, e.g., id.* at 6 (text file of exemplary access list). The NLANR system is likewise configured to serve many users, requesting resources from many content providers, in which multiple content providers share each of the many repeaters. *Id.*

A POSITA would have been motivated to combine Microsoft Proxy Server’s distributed caching, proxy chaining, and alias-based routing (*see generally* Exhibit A-05 [Microsoft Proxy

Server], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with the CDN architecture described in NLANR, in which requests are directed to shared repeaters based on alias names associated with appropriate geographic regions (*see, e.g.*, Exhibit A-03 [NLANR], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]) to expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems.

(8) Microsoft Proxy Server in view of Squid Cache

It would have been obvious to combine Microsoft Proxy Server and Squid Cache because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft Proxy Server routes requests and responses between the Internet and client computers, acting as a liaison between them. In addition to routing requests, Proxy Server provides a cache of frequently requested Internet sites, blocks access to specified sites, and provides secure access between an internal network and the Internet. Proxy Server allows users to set up distributed content caching among multiple Proxy Server computers, using array-based content caching and/or chain-based content caching. Proxy Server can also use reverse proxying and reverse hosting to send requests to Web publishing servers connected behind the Proxy Server computer. *See, e.g.*, ProxyServer2.pdf at 1, 10, 12-13. Proxy Server also implements routing algorithms that include using an array membership table and computing hashes on the Proxy Server array member names. *See, e.g., id.* at 87-89.

Squid is a proxy that acts as an intermediary in a web transaction and also a cache that stores recently retrieved web content for possible reuse. Squid can operate in cache accelerator

mode where it acts as an origin web server and fetches content from one or more origin servers. *See, e.g.*, Squid Users Guide - Cache Accelerator Mode at 1-2; Squid FAQ 4 at 2. Squid allows for acceleration of many origin servers using the Host header, which conserves IP addresses since only one is needed to handle an unlimited number of hosts. The client sends the desired hostname in the HTTP/1.1 Host header, and a redirector translates the specified hostname into the origin server.

A POSITA would have been motivated to combine Microsoft Proxy Server's distributed caching, proxy chaining, and alias-based routing (see generally Exhibit A-05 [Microsoft Proxy Server], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with Squid Cache, including its implementation in cache accelerator mode (*see, e.g.*, Exhibit A-07 [Squid Cache], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), and expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(9) Microsoft Proxy Server in view of Microsoft IIS

It would have been obvious to combine Microsoft Proxy Server and Microsoft IIS because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft Proxy Server routes requests and responses between the Internet and client computers, acting as a liaison between them. In addition to routing requests, Proxy Server provides a cache of frequently requested Internet sites, blocks access to specified sites, and provides secure access between an internal network and the Internet. Proxy Server allows users to set up distributed content caching among multiple Proxy Server computers, using array-based content caching

and/or chain-based content caching. Proxy Server can also use reverse proxying and reverse hosting to send requests to Web publishing servers connected behind the Proxy Server computer. *See, e.g., ProxyServer2.pdf* at 1, 10, 12-13. Proxy Server also implements routing algorithms that include using an array membership table and computing hashes on the Proxy Server array member names. *See, e.g., id.* at 87-89.

Microsoft IIS 4.0 (IIS) is a standards-based application server integrated with the Microsoft Windows NT Server operating system that was designed to support multiple Web server scenarios, ranging from simple web sites on a corporate Intranet to large ISP web hosting farms. Windows NT Server at 4-5. Amongst many other features, Microsoft IIS enables organizations to host multiple web sites using a single IP address through the use of host headers. *Id.* For the website to be accessible to users, registration of the host header name and associated IP address with a DNS server is needed. *Id.*

A POSITA would have been motivated to combine Microsoft Proxy Server's distributed caching, proxy chaining, and alias-based routing (*see generally* Exhibit A-05 [Microsoft Proxy Server], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with Microsoft IIS, including its use of host headers and integration with Windows DNS Server (*see, e.g.,* Exhibit A-06 [Microsoft IIS], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), and expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems.

(10) Squid Cache in view of Kenner

It would have been obvious to combine Squid Cache and Kenner because the combination does nothing more than combine prior art elements from similar systems according to known

methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Squid is a proxy that acts as an intermediary in a web transaction and also a cache that stores recently retrieved web content for possible reuse. Squid can operate in cache accelerator mode where it acts as an origin web server and fetches content from one or more origin servers. *See, e.g.*, Squid Users Guide - Cache Accelerator Mode at 1-2; Squid FAQ 4 at 2. Squid allows for acceleration of many origin servers using the Host header, which conserves IP addresses since only one is needed to handle an unlimited number of hosts. The client send the desired hostname in the HTTP/1.1 Host header, and a redirector translates the specified hostname into the origin server.

Kenner describes a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. Specifically, Kenner discloses a distributed digital video clip delivery system, whereby video clips, stored locally or at remote locations, can be requested by an authorized client user and retrieved at the user's terminal. *See, e.g.*, Kenner at Abstract. Kenner's system involves origin servers (source IMs), and repeaters (primary or regional IMs), where the repeaters have extended storage and retrieval units for storing resources requested by clients. *Id.* at 10:18-31, 31:19-21, 31:30-32, 32:14-17, 44:6-14. The resources (video clips) are uploaded to the IMs by content providers, and in Kenner's scalable system, many IMs can be shared by multiple such providers, servicing multiple users. *See, e.g., id.* at 5:21-26, 40:2-28, 54:12-16. The IMs include clip databases, identifying the clips stored in their extended SRUs by alias names called "video IDs." *Id.* at 34:3-6, 37:20-22.

A POSITA would have been motivated to combine Squid Cache, including its implementation in cache accelerator mode (*see generally* Exhibit A-07 [Squid Cache], cls. 28[a]-

[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with Kenner's system in which alias names identify content provider and content (*see, e.g.*, Exhibit A-02 [Kenner], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), to expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(11) Squid Cache in view of Armbruster

It would have been obvious to combine Squid Cache and Armbruster because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Squid is a proxy that acts as an intermediary in a web transaction and also a cache that stores recently retrieved web content for possible reuse. Squid can operate in cache accelerator mode where it acts as an origin web server and fetches content from one or more origin servers. *See, e.g.*, Squid Users Guide - Cache Accelerator Mode at 1-2; Squid FAQ 4 at 2. Squid allows for acceleration of many origin servers using the Host header, which conserves IP addresses since only one is needed to handle an unlimited number of hosts. The client send the desired hostname in the HTTP/1.1 Host header, and a redirector translates the specified hostname into the origin server.

Armbruster also teaches a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. *See, e.g.*, Armbruster at 4:27-35. Specifically, Armbruster discloses a Managed Distributed Caching (MDC) system, in which content providers can upload resources from origin servers to local or centralized caches (repeaters). *See id.*; *see also* Figs. 1-3. Clients that are authorized can request resources uploaded by content providers, and

those requests are directed to specific repeaters via alias names associated with those repeaters, as well as the content provider associated with the requested resource, and a lookup table identifying repeaters by their respective aliases. *See, e.g., id.* at Fig. 3, 4:27-52, 6:5-40 (Scenarios 2 and 3). Armbruster's system can also accommodate multiple users, multiple content providers, and multiple repeater servers. *Id.*

A POSITA would have been motivated to combine Squid Cache, including its implementation in cache accelerator mode (*see generally* Exhibit A-07 [Squid Cache], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with Armbruster's system in which alias names are associated with specific repeaters (caches) (*see, e.g.,* Exhibit A-01 [Armbruster], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), to expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(12) Squid Cache in view of RFC 1034

It would have been obvious to combine Squid Cache and RFC1034 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Squid is a proxy that acts as an intermediary in a web transaction and also a cache that stores recently retrieved web content for possible reuse. Squid can operate in cache accelerator mode where it acts as an origin web server and fetches content from one or more origin servers. *See, e.g.,* Squid Users Guide - Cache Accelerator Mode at 1-2; Squid FAQ 4 at 2. Squid allows for acceleration of many origin servers using the Host header, which conserves IP addresses since only one is needed to handle an unlimited number of hosts. The client send the desired hostname

in the HTTP/1.1 Host header, and a redirector translates the specified hostname into the origin server.

RFC1034 also describes a system in which multiple users can request resources from multiple distributed caches. *See, e.g.*, RFC1034 at § 2.3 (“The domain system defines procedures for accessing the data and for referrals to other name servers. The domain system also defines procedures for caching retrieved data and for periodic refreshing of data defined by the system administrator.”) RFC1034 also describes the use of alias names for identifying specific name servers (i.e., through a system of aliases and canonical names (CNAMEs)), and use of a lookup table to direct requests to servers corresponding to those alias names. *Id.* at §§ 3.6.2, 5.2.2.

A POSITA would have been motivated to combine Squid Cache, including its implementation in cache accelerator mode (*see generally* Exhibit A-07 [Squid Cache], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with the alias and CNAME system described by RFC1034, in which specific servers can be associated with multiple aliases (*see, e.g.*, Exhibit A-04 [RFC1034], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), to expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(13) Squid Cache in view of NLANR

It would have been obvious to combine Squid Cache and NLANR because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Squid is a proxy that acts as an intermediary in a web transaction and also a cache that stores recently retrieved web content for possible reuse. Squid can operate in cache accelerator

mode where it acts as an origin web server and fetches content from one or more origin servers. *See, e.g.,* Squid Users Guide - Cache Accelerator Mode at 1-2; Squid FAQ 4 at 2. Squid allows for acceleration of many origin servers using the Host header, which conserves IP addresses since only one is needed to handle an unlimited number of hosts. The client send the desired hostname in the HTTP/1.1 Host header, and a redirector translates the specified hostname into the origin server.

NLANR also describes a CDN in which multiple shared repeater servers could service requests from multiple users, for content uploaded by multiple content providers. *See* NLANR at 1 (describing project background as providing a “solution to the problem of scalable wide area information dissemination”); *id.* at 2 (describing “NLANR Cache Architecture” as including six large caches distributed geographically, configured to distribute loads so that each handles a subset of top-level-domains, and each handling resources from certain geographic regions); *see also id.* at 5 (NLANR Cache Configuration map, showing location of geographic caches and regional domains and objects intended to be directed to associated caches). NLANR also discloses a system in which requests are directed to specific repeaters (caches) via alias names (geographic domains) intended to be associated with a given repeater, as provided in a text-based lookup table. *See id.*; *see also, e.g., id.* at 6 (text file of exemplary access list). The NLANR system is likewise configured to serve many users, requesting resources from many content providers, in which multiple content providers share each of the many repeaters. *Id.*

A POSITA would have been motivated to combine Squid Cache, including its implementation in cache accelerator mode (*see generally* Exhibit A-07 [Squid Cache], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with the CDN architecture described in NLANR, in which requests are directed to shared repeaters based on alias names associated with

appropriate geographic regions (*see, e.g.*, Exhibit A-03 [NLANR], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), and expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(14) Microsoft IIS in view of Kenner

It would have been obvious to combine Microsoft IIS and Kenner because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft IIS 4.0 (IIS) is a standards-based application server integrated with the Microsoft Windows NT Server operating system that was designed to support multiple Web server scenarios, ranging from simple web sites on a corporate Intranet to large ISP web hosting farms. Windows NT Server at 4-5. Amongst many other features, Microsoft IIS enables organizations to host multiple web sites using a single IP address through the use of host headers. *Id.* For the website to be accessible to users, registration of the host header name and associated IP address with a DNS server is needed. *Id.*

Kenner describes a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. Specifically, Kenner discloses a distributed digital video clip delivery system, whereby video clips, stored locally or at remote locations, can be requested by an authorized client user and retrieved at the user's terminal. *See, e.g.*, Kenner at Abstract. Kenner's system involves origin servers (source IMs), and repeaters (primary or regional IMs), where the repeaters have extended storage and retrieval units for storing resources requested by clients. *Id.* at 10:18-31, 31:19-21, 31:30-32, 32:14-17, 44:6-14. The resources (video clips) are uploaded to the IMs by content providers, and in Kenner's scalable system, many IMs can be

shared by multiple such providers, servicing multiple users. *See, e.g., id.* at 5:21-26, 40:2-28, 54:12-16. The IMs include clip databases, identifying the clips stored in their extended SRUs by alias names called “video IDs.” *Id.* at 34:3-6, 37:20-22.

A POSITA would have been motivated to combine Microsoft IIS, including its use of host headers and integration with Windows DNS Server (*see generally* Exhibit A-06 [Microsoft IIS], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with Kenner’s system in which alias names identify content provider and content (*see, e.g.,* Exhibit A-02 [Kenner], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), to expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(15) Microsoft IIS in view of Ambruster

It would have been obvious to combine Microsoft IIS and Ambruster because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft IIS 4.0 (IIS) is a standards-based application server integrated with the Microsoft Windows NT Server operating system that was designed to support multiple Web server scenarios, ranging from simple web sites on a corporate Intranet to large ISP web hosting farms. Windows NT Server at 4-5. Amongst many other features, Microsoft IIS enables organizations to host multiple web sites using a single IP address through the use of host headers. *Id.* For the website to be accessible to users, registration of the host header name and associated IP address with a DNS server is needed. *Id.*

Armbruster also teaches a CDN in which multiple content providers can provide multiple resources via one or more shared repeater servers. *See, e.g.*, Armbruster at 4:27-35. Specifically, Armbruster discloses a Managed Distributed Caching (MDC) system, in which content providers can upload resources from origin servers to local or centralized caches (repeaters). *See id.*; *see also* Figs. 1-3. Clients that are authorized can request resources uploaded by content providers, and those requests are directed to specific repeaters via alias names associated with those repeaters, as well as the content provider associated with the requested resource, and a lookup table identifying repeaters by their respective aliases. *See, e.g., id.* at Fig. 3, 4:27-52, 6:5-40 (Scenarios 2 and 3). Armbruster's system can also accommodate multiple users, multiple content providers, and multiple repeater servers. *Id.*

A POSITA would have been motivated to combine Microsoft IIS, including its use of host headers and integration with Windows DNS Server (*see generally* Exhibit A-06 [Microsoft IIS], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with Armbruster's system in which alias names are associated with specific repeaters (caches) (*see, e.g.*, Exhibit A-01 [Armbruster], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), to expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(16) Microsoft IIS in view of RFC 1034

It would have been obvious to combine Microsoft IIS and RFC1034 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft IIS 4.0 (IIS) is a standards-based application server integrated with the Microsoft Windows NT Server operating system that was designed to support multiple Web server scenarios, ranging from simple web sites on a corporate Intranet to large ISP web hosting farms. Windows NT Server at 4-5. Amongst many other features, Microsoft IIS enables organizations to host multiple web sites using a single IP address through the use of host headers. *Id.* For the website to be accessible to users, registration of the host header name and associated IP address with a DNS server is needed. *Id.*

RFC1034 also describes a system in which multiple users can request resources from multiple distributed caches. *See, e.g.,* RFC1034 at § 2.3 (“The domain system defines procedures for accessing the data and for referrals to other name servers. The domain system also defines procedures for caching retrieved data and for periodic refreshing of data defined by the system administrator.”) RFC1034 also describes the use of alias names for identifying specific name servers (i.e., through a system of aliases and canonical names (CNAMEs)), and use of a lookup table to direct requests to servers corresponding to those alias names. *Id.* at §§ 3.6.2, 5.2.2.

A POSITA would have been motivated to combine Microsoft IIS, including its use of host headers and integration with Windows DNS Server (*see generally* Exhibit A-06 [Microsoft IIS], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with the alias and CNAME system described by RFC1034, in which specific servers can be associated with multiple aliases (*see, e.g.,* Exhibit A-04 [RFC1034], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), to expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

(17) Microsoft IIS in view of NLANR

It would have been obvious to combine Microsoft IIS and NLANR because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft IIS 4.0 (IIS) is a standards-based application server integrated with the Microsoft Windows NT Server operating system that was designed to support multiple Web server scenarios, ranging from simple web sites on a corporate Intranet to large ISP web hosting farms. Windows NT Server at 4-5. Amongst many other features, Microsoft IIS enables organizations to host multiple web sites using a single IP address through the use of host headers. *Id.* For the website to be accessible to users, registration of the host header name and associated IP address with a DNS server is needed. *Id.*

NLANR also describes a CDN in which multiple shared repeater servers could service requests from multiple users, for content uploaded by multiple content providers. *See* NLANR at 1 (describing project background as providing a “solution to the problem of scalable wide area information dissemination”); *id.* at 2 (describing “NLANR Cache Architecture” as including six large caches distributed geographically, configured to distribute loads so that each handles a subset of top-level-domains, and each handling resources from certain geographic regions); *see also id.* at 5 (NLANR Cache Configuration map, showing location of geographic caches and regional domains and objects intended to be directed to associated caches). NLANR also discloses a system in which requests are directed to specific repeaters (caches) via alias names (geographic domains) intended to be associated with a given repeater, as provided in a text-based lookup table. *See id.*; *see also, e.g., id.* at 6 (text file of exemplary access list). The NLANR system is likewise

configured to serve many users, requesting resources from many content providers, in which multiple content providers share each of the many repeaters. *Id.*

A POSITA would have been motivated to combine Microsoft IIS, including its use of host headers and integration with Windows DNS Server (*see generally* Exhibit A-06 [Microsoft IIS], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), with the CDN architecture described in NLANR, in which requests are directed to shared repeaters based on alias names associated with appropriate geographic regions (*see, e.g.*, Exhibit A-03 [NLANR], cls. 28[a]-[c], 29[a], 30[a], 31, 34[a]-[b], 37[a][i]-[ii], 37[c]-[e]), and expand the capabilities of the combined system. The POSA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible systems.

2. U.S. Patent No. 8,924,466

Microsoft identifies in Appendix B the prior art references that render obvious the Asserted Claims of the '466 Patent and include below exemplary combinations showing the obviousness of the '466 Patent Asserted Claims in view of the prior art. To the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix B, the limitation would have been obvious in light of the disclosures within the reference and the knowledge of one of ordinary skill in the art at the time of the '466 Patent. Moreover, to the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix B, such reference may be combined with any other references listed in Appendix B for such element, thereby rendering the claims invalid for obviousness. To the extent a finder of fact determines that a limitation of any of the '466 Patent Asserted Claims is not disclosed by one of the references identified above, the claim is nevertheless unpatentable as obvious because they contain nothing that constitutes a patentable innovation. To the extent a

finder of fact determines that a limitation of the '466 Patent Asserted Claims is not anticipated, it does not go beyond combining familiar elements according to known methods to achieve predictable results or does more than choose between clear alternatives known to those of ordinary skill in the art.

a. Obviousness

For at least the reasons described below, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix B with any other reference or references listed in Appendix B along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '466 Patent. For example, and without limitation, the Asserted Claims of the '466 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
Cranor	• Gamerman • McCanne • O'Rourke • Outten • RFC2187 • Adrangi • Lahr • Wu • Yates • Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
Gamerman	• Cranor • McCanne • O'Rourke • Outten • RFC2187 • Adrangi • Lahr

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Wu • Yates • Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
McCanne	• Cranor • Gamerman • O'Rourke • Outten • RFC2187 • Adrangi • Lahr • Wu • Yates • Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
O'Rourke	• Cranor • Gamerman • McCanne • Outten • RFC2187 • Adrangi • Lahr • Wu • Yates • Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
Outten	• Cranor • Gamerman • McCanne • O'Rourke • RFC2187 • Adrangi • Lahr • Wu • Yates

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
RFC2187	• Andragi • Cranor • Gamerman • McCanne • O'Rourke • Outten • Adrangi • Lahr • Wu • Yates • Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
Adrangi	• Cranor • Gamerman • McCanne • O'Rourke • Outten • RFC2187 • Lahr • Wu • Yates • Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
Lahr	• Cranor • Gamerman • McCanne • O'Rourke • Outten • RFC2187 • Adrangi • Wu • Yates • Akamai FreeFlow

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
Wu	• Cranor • Gamerman • McCanne • O'Rourke • Outten • RFC2187 • Adrangi • Lahr • Yates • Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
Yates	• Cranor • Gamerman • McCanne • O'Rourke • Outten • RFC2187 • Adrangi • Lahr • Wu • Akamai FreeFlow • Akamai EdgeSuite • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
Akamai FreeFlow	• Cranor • Gamerman • McCanne • O'Rourke • Outten • RFC2187 • Adrangi • Lahr • Wu • Yates • Akamai EdgeSuite • Windows Media Services v4.1

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
Akamai EdgeSuite	• The knowledge of a person of ordinary skill in the art. • Cranor • Gamerman • McCanne • O'Rourke • Outten • RFC2187 • Adrangi • Lahr • Wu • Yates • Akamai FreeFlow • Windows Media Services v4.1 • The knowledge of a person of ordinary skill in the art.
Windows Media Services v4.1	• Cranor • Gamerman • McCanne • O'Rourke • Outten • RFC2187 • Adrangi • Lahr • Wu • Yates • Akamai FreeFlow • Akamai EdgeSuite • The knowledge of a person of ordinary skill in the art.

b. General Motivations to Combine

Obviousness: For at least the reasons described in these contentions, it would have been obvious to one of ordinary skill in the art to combine any of a number of prior art references, including any combination of those prior art references identified in Appendix B along with the knowledge of one of ordinary skill in the art to meet the limitations of the '466 Patent Asserted Claims. As a result, the references identified herein render one or more Asserted Claims of the '466 Patent obvious when the references are read in combination with each other, and/or when

read in view of the state of the art and knowledge of those skilled in the art. Each and every reference identified is also relevant to the state of the art at the time of the alleged invention. Any of the references disclosed above may be combined with one another to render obvious (and therefore invalid) each of Sandpiper's Asserted Claims. Defendant may rely upon a subset of the identified references or all of the references identified herein, including all references in Appendix B, for purposes of obviousness depending on the Court's claim construction, positions taken by Sandpiper during this litigation, and further investigation and discovery.

Reservation of Rights: Moreover, as mentioned above, Defendant has not yet completed their search or discovery concerning additional prior art. As such, Defendant's inclusion of exemplary combinations does not preclude them from identifying other invalidating combinations as appropriate, and Defendant reserves the right to identify additional specific combinations as well as to detail and explain such combinations.

No More Than Ordinary Variations of Prior Art; No Unexpected Results: To the extent not anticipated, the '466 Patent Asserted Claims represent no more than the result of ordinary variations of the prior art. Defendant further believes that no showing of a specific motivation to combine prior art is required to combine the references disclosed above and in the attached charts, as each combination of art would have no unexpected results, and at most would simply represent a known alternative to one of ordinary skill in the art. *See KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 415-16 (2007) (rejecting the Federal Circuit's "rigid" application of the teaching, suggestion, or motivation to combine test, instead espousing an "expansive and flexible" approach); *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1058 (Fed. Cir. 2016); *Intercontinental Great Brands LLC v. Kellogg N. Am. Co.*, 869 F.3d 1336, 1344 (Fed. Cir. 2017). Indeed, the Supreme Court held that a person of ordinary skill in the art is "a person of ordinary

creativity, not an automaton” and “in many cases a person of ordinary skill in the art will be able to fit the teachings of multiple patents together like pieces of a puzzle.” *Id.* at 420-21. Nevertheless, in addition to the information contained elsewhere in these contentions, Defendant identifies motivation and reason to combine the cited art.

Predictable Results; Obvious to Try: One or more combinations of the prior art references identified in Appendix B would have been obvious because these references would have been combined using: known methods to yield predictable results; known techniques in the same way; a simple substitution of one known, equivalent element for another to obtain predictable results; and/or a teaching, suggestion, or motivation in the prior art generally. In addition, it would have been obvious to try combining the prior art references identified above because there were only a finite number of predictable solutions and/or because known work in one field of endeavor prompted variations based on predictable design incentives and/or market forces either in the same field or a different one. Further, the combinations of the prior art references identified in Appendix B would have been obvious because the combinations represent known potential options with a reasonable expectation of success.

Interrelated Teachings: Additional evidence that there would have been a motivation to combine the prior art references identified above includes the interrelated teachings of multiple prior art references; common authorship; the effects of demands known to the design community or present in the marketplace; the existence of a known problem for which there was an obvious solution encompassed by the ‘466 Patent Asserted Claims; the existence of a known need or problem in the field of the endeavor at the time of the alleged invention(s); and the background knowledge that would have been possessed by a person having ordinary skill in the art.

Summary of Motivations to Combine: Thus, the motivation to combine the teachings of the prior art references disclosed in Appendix B is found in the references themselves and also in: (1) the nature of the problem being solved; (2) the express, implied and inherent teachings of the prior art; (3) the knowledge of persons of ordinary skill in the art; (4) the predictable results obtained in combining the different elements of the prior art; (5) the predictable results obtained in simple substitution of one known element for another; (6) the use of a known technique to improve similar devices, methods, or products in the same way; (7) the predictable results obtained in applying a known technique to a known device, method, or product ready for improvement; (8) the finite number of identified predictable solutions that had a reasonable expectation of success; and (9) known work in various technological fields that could be applied to the same or different technological fields based on design incentives or other market forces.

Same Area of Technology: Additionally, it would be obvious to one of ordinary skill in the art to consult and/or combine any of the prior art listed in Appendix B because all of these references relate to the same area of technology and/or are from analogous art. The ‘466 Patent Asserted Claims are directed to “a content delivery method” that “directs a request by a client for an object to a first server in a content delivery network (CDN), regardless of whether the first server has the requested object. If the first server has the requested object, then it is served to the client from the first server; otherwise, the first server redirects the client request to a second server in the CDN to handle the request. The second server may be a peer server of the first server. The first and second servers may be edge servers in the CDN.” ‘466 Patent at Abstract. The ‘466 Patent Asserted Claims merely unite old elements, well known in the field, with no change in their respective function or result. Given the interrelated teachings of the prior art, the effects of demands known to the design community or present in the marketplace, and the background

knowledge possessed by a person having ordinary skill in the art, it would have been obvious for one of ordinary skill in the art to combine these familiar elements, disclosed and/or embodied in the prior art listed above to practice the ‘466 Patent Asserted Claims.

Same Technical Issues: A person of ordinary skill would have been motivated to combine the above prior art based on his/her knowledge, the nature of the problem to be solved, and the teachings of the prior art. The identified prior art addresses the same or similar technical issues and suggests the same or similar solutions to those issues. Moreover, some of the prior art refer to or discuss other prior art, illustrating the close technical relationship among the prior art. For the ‘466 Patent, one of ordinary skill in the art would have been motivated to combine known prior art solutions described in these references relating to a content delivery method that directs a request by a client to a first server, and then to additional servers up the hierarchy depending on which server has the requested object. The motivation lies in increasing scalability, reliability, and protecting against failure. It would have been further obvious to redirect a client to another server in the hierarchy. The motivation to combine comes from desires to increase efficiency and to ensure that clients can receive the object that they are requesting. A POSITA would have had a reasonable expectation of success because combining organization of client requests and hierarchies were commonly used in content delivery networks.

It would have been obvious to combine or modify each reference with another reference identified in Sections IV.B to render these claims invalid because each teaches complementary aspects of tiered content delivery. A person of ordinary skill in the art (“POSITA”) would have been motivated to combine these teachings to increase efficiency, scalability and reliability—well-recognized design goals in CDN systems. Combining known architectures and request organization with hierarchies and request redirection would have been a predictable use of prior

art elements according to their established functions, yielding no unexpected results. Market demand for efficient content delivery and widespread adoption of hierarchical content delivery networks would have provided further motivation. POSITA would have had a reasonable expectation of success because these techniques were routinely implemented in distributed systems at the time.

Demand and Market Forces: For example, market forces suggested a demand for increased security and versatility of user authentication systems, and thus one of skill in the art would have thought to combine or modify references that described known methods that one of skill in the art would have recognized as offering improvements to solutions of that time.

Known Elements; Obvious to Try: Moreover, many of the claim elements were already known as admitted by the applicants in the patent specification. These elements represented design choices available to a person of ordinary skill. When there is a design need or market pressure to solve a problem such as identified previously and/or described in the ‘466 Patent, and there are a finite number of identified, predictable solutions, a person of ordinary skill would be motivated to combine the known options that are within his or her technical grasp. For the ‘466 Patent, there are a finite number of approaches for expanding the capabilities of proximity systems and location tracking systems to run or interact with different applications. For example, as of the priority date of the ‘466 Patent, a POSITA would have been very familiar with basic elements of the ‘466 Patent’s claims such as hierarchical content delivery networks and redirects. *See* Appx. B. Such elements were within the knowledge of a POSITA, whether or not disclosed in any particular reference, and the inclusion of such elements in the ‘466 Patent’s claims does not render them novel or non-obvious.

c. Exemplary Combinations

For at least the reasons described above, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix B with any other reference or references listed in Appendix B along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '466 Patent.

For example, and without limitation, the Asserted Claims of the '466 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

(1) Adrangi in view of Lahr

It would have been obvious to combine Adrangi and Lahr because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Adrangi describes a content delivery method where a client request is directed to an edge POP, and if the content is not available, the client is redirected to another server in the content delivery network (CDN). Lahr is directed to a CDN with a tiered hierarchy of servers, including media serving systems (i.e., edge servers), regional data centers, and master data centers.

Adrangi describes a two-tiered system where a client is redirected from an edge server to a parent server if the content is not found at the edge. Lahr, with its “director” and multi-tiered architecture (e.g., edge, regional, master data centers), explicitly teaches the scenario where the parent server itself might not have the content and would need to direct the client to another server in the CDN. Lahr’s director will automatically route traffic to a master data center if a regional data center (i.e. the “second server” in the claim) is unavailable or does not have the requested content. Therefore, Lahr discloses further details of redirection within a multi-tiered CDN. *See, e.g., Exhibits B7, B8, cls. 1(f), 7(f), 10(f).* Additionally, Lahr reinforces and strengthens the

concept of dynamic server selection based on network conditions and server load, as its director considers network performance, CPU load, and application status when determining the optimal server. *See* B8, cls. 1(b), 7(b), 10(b).

A POSITA would have been motivated to combine Adrangi and Lahr to create a more robust and efficient CDN system. A POSITA would have recognized that Adrangi provides a foundation for initial request handling and redirection to a parent server, while Lahr offers a more sophisticated approach to multi-tiered redirection and dynamic server selection. *See* Adrangi, Abstract, 2:30–58, 9:26–35, 10:59–67, FIGs. 2 & 7 & 13; Lahr, Abstract, 3:26–4:3, 5:27–6:11, 20:26–21:2, 22:18–27, FIGs. 1 & 10 & 12. Specifically, a POSITA would combine Adrangi’s initial request handling and redirection to a parent server, Adrangi, 9:25–32, 10:50–58, 10:46–65, with Lahr’s multi-tiered redirection process managed by the “director,” Lahr, 9:27–29, 10:10–13, 19:27–20:2, 20:26–21:2, 21:10–16, 22:18–27, 23:6–16. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based on factors such as network load, server load, and user location. A POSITA would have recognized that this combination would have led to improved performance, scalability, and reliability of the CDN system.

(2) O’Rourke in view of Outten and Wu

It would have been obvious to combine O’Rourke with Outten and Wu because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

O’Rourke, Outten, and Wu all describe content delivery systems with layered servers and request forwarding. O’Rourke, Abstract, FIGs. 2-3; Outten, Abstract, FIGs. 1-2; Wu, 2:47-67, FIG.

1. Furthermore, the O’Rourke-Outten combination was used by the Examiner during prosecution for an early rejection (along with Cranor and Gamerman) further evidencing that a POSITA would have been motivated to combine the teachings of O’Rourke and Outten. Furthermore, as was shown by the Examiner, a POSITA would have also combined the teachings of O’Rourke, Outten, Cranor, and Gamerman.

O’Rourke discloses an environment (CDN) with an origin server and multiple cache servers. Each cache server is used to redistribute media content to local clients. Outten discloses a multiple layer architecture of servers, configured to efficiently deliver content, comprised of a plurality of edge servers and parent servers. To the extent that O’Rourke does not disclose or render obvious the claimed hierarchical CDN with parent servers, a POSITA would have been motivated, for example, to include Outten’s parent server as one of its cache servers. See, e.g., cls. 1[a], 1[e]. For example, O’Rourke describes cache servers that utilize policies to determine their function, such as serving content to other cache servers or to clients. O’Rourke, Abstract, ¶¶28-30. Outten provides that servers may act as a “parent” to edge servers. Outten, 2:64-3:14. Including Outten’s parent server would have enhanced O’Rourke’s system with an additional policy and functionality to better serve content to clients. Furthermore, a POSITA would have been motivated to incorporate Outten’s load balancing techniques into O’Rourke’s CDN to better accommodate clients and serve content in a more orderly manner. Outten, 15:56-67.

Further, O’Rourke describes that when the cache server handling a client request does not have the requested content available, the system’s cache policies determine whether the cache server will download the requested content or perform a redirect. If the system is configured to perform redirection, the cache server redirects the client to another server. Furthermore, O’Rourke describes that the cache server may redirect the client to another cache server or to an origin server.

O'Rourke, ¶¶30, 33, 37, 41, 5, FIG. 5. Outten describes that in its multiple layer architecture that includes edge servers and parent servers, an edge server stores less than the full content of its parent server. When an edge server does not have a content item requested by a user, the edge server requests the content item from its parent. Outten, 13:53-55, 14:20-24, 15:16-18. Thus, to the extent O'Rourke does not disclose or render obvious redirecting a client to a parent server, a POSITA would have been motivated to utilize Outten's method of directing a client to a parent server to obtain the requested content. Redirecting to the parent server would have been advantageous over redirecting to another edge server because of the greater likelihood of a "hit" at the parent server compared to another edge server. And redirecting to the parent server would have been advantageous over redirecting to an origin server, since the origin server could be overwhelmed if used as the receiver of client requests.

Additionally, Wu discloses that when a cache server needs to redirect a request from a client (e.g., a client web browser), due to CPU overloading of the cache server or other conditions, the cache server responds to the client with the IP address of a suggested sibling cache server. See, e.g., cls. 1(e), 7(e), 10(e). The client then redirects its request to sibling cache server. Wu, 5:51-67, 6:4-19, FIG. 8. A POSITA would have been motivated to include Wu's method of sending a response to the client to redirect the client to another server because doing so would free up the edge server for other operations and allow the client to maintain control when fetching the desired content.

Furthermore, considering the similarities in the CDN architectures of O'Rourke, Outten, and Wu, a POSITA would have been equally motivated to combine Outten's load balancing and network traffic management with O'Rourke's CDN to better manage redirection of the client to more available cache servers. Outten, 13:36-48, 15:56-67; see cls. 1[c], 4, 7[c], 10[c].

A POSITA would have had a reasonable expectation of success in combining the teachings of O'Rourke, Outten and Wu. The combination would have involved only routine engineering leading to predictable results. For example, adding policies to O'Rourke's servers to modify how they handle content requests was well known and well within the skill of a POSITA.

(3) Windows Media Services v4.1 in view of Adrangi

It would have been obvious to combine Windows Media Services v4.1 and Adrangi because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Adrangi describes a content delivery method where a client request is directed to an edge POP, and if the content is not available, the client is redirected to another server in the content delivery network (CDN). Windows Media Services v4.1 enables caching and proxying for use with a hierarchy of servers, including a CDN's origin servers.

A POSITA would have been motivated to combine Windows Media Services v4.1 and Adrangi to create a more robust and efficient CDN system. A POSITA would have recognized that Adrangi provides a foundation for initial request handling and redirection to a parent server, while Windows Media Services v4.1s offers a more sophisticated approach using caching and proxying. *See* Adrangi, Abstract, 2:30–58, 9:26–35, 10:59–67, FIGs. 2 & 7 & 13; *e.g.*, Windows Media Services v4.1, Hercules Release Slides at 2–3, 8–11, 28, 33–34. Specifically, a POSITA would combine Adrangi's initial request handling and redirection to a parent server, Adrangi, 9:25–32, 10:50–58, 10:46–65, with Windows Media Services v4.1s' caching and proxying functionality, *e.g.*, Windows Media Services v4.1, Hercules Release Slides at 2–3, 8–11, 28, 33–34. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers

of servers based on factors such as network load, server load, and user location, while using proxies and caches. A POSITA would have recognized that this combination would have led to improved performance, scalability, security, and reliability of the CDN system.

(4) Windows Media Services v4.1 in view of Lahr

It would have been obvious to combine Windows Media Services v4.1 and Lahr because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Windows Media Services v4.1 enables caching and proxying for use with a hierarchy of servers, including a CDN's origin servers. Lahr is directed to a CDN with a tiered hierarchy of servers, including media serving systems (i.e., edge servers), regional data centers, and master data centers.

A POSITA would have been motivated to combine Windows Media Services v4.1 and Lahr to create a more robust and efficient CDN system. A POSITA would have recognized that Windows Media Services v4.1s offers a initial request handling and redirection, as well as tiered servers, and Lahr offers a more sophisticated approach to multi-tiered redirection and dynamic server selection. *See, e.g.*, Windows Media Services v4.1, Hercules Release Slides at 2–3, 8–11, 28, 33–34; Lahr, Abstract, 3:26–4:3, 5:27–6:11, 20:26–21:2, 22:18–27, FIGs. 1 & 10 & 12. Specifically, a POSITA would combine Windows Media Services v4.1s' caching and proxying functionality, *e.g.*, Windows Media Services v4.1, Hercules Release Slides at 2–3, 8–11, 28, 33–34, with Lahr's multi-tiered redirection process managed by the "director," Lahr, 9:27–29, 10:10–13, 19:27–20:2, 20:26–21:2, 21:10–16, 22:18–27, 23:6–16. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based

on factors such as network load, server load, and user location. A POSITA would have recognized that this combination would have led to improved performance, scalability, and reliability of the CDN system.

(5) Windows Media Services v4.1 in view of O'Rourke

It would have been obvious to combine Windows Media Services v4.1 and O'Rourke because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Windows Media Services v4.1 enables caching and proxying for use with a hierarchy of servers, including a CDN's origin servers. O'Rourke discloses an environment (CDN) with an origin server and multiple cache servers. Each cache server is used to redistribute media content to local clients.

A POSITA would have been motivated to combine Windows Media Services v4.1 and O'Rourke to create a more robust and efficient CDN system. A POSITA would have recognized that Windows Media Services v4.1s offers a initial request handling and redirection, as well as tiered servers, whereas O'Rourke discloses cache servers that utilize policies to determine their function, such as serving content to other cache servers or to clients. *See, e.g.*, Windows Media Services v4.1, Hercules Release Slides at 2–3, 8–11, 28, 33–34; O'Rourke, Abstract, ¶¶28–30. Specifically, a POSITA would combine Windows Media Services v4.1s' caching and proxying functionality, *e.g.*, Windows Media Services v4.1, Hercules Release Slides at 2–3, 8–11, 28, 33–34, with O'Rourke's description of cache server handling client requests, where when the cache server does not have the requested content available, the system's cache policies determine whether the cache server will download the requested content or perform a redirect. If the system is configured to perform redirection, the cache server redirects the client to another server.

O'Rourke also describes that the cache server may redirect the client to another cache server or to an origin server. O'Rourke, ¶¶30, 33, 37, 41, 5, FIG. 5. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based on factors such as network load, server load, and user location. A POSITA would have recognized that this combination would have led to improved performance, scalability, and reliability of the CDN system.

A POSITA would have had a reasonable expectation of success in combining the teachings of Windows Media Services v4.1s and O'Rourke. The combination would have involved only routine engineering leading to predictable results.

(6) Akamai FreeFlow in view of Adrangi

It would have been obvious to combine Akamai FreeFlow and Adrangi because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Adrangi describes a content delivery method where a client request is directed to an edge POP, and if the content is not available, the client is redirected to another server in the content delivery network (CDN). Akamai FreeFlow describes a CDN with hierarchical servers including edge and origin servers, caching, and failure detection.

A POSITA would have been motivated to combine Akamai FreeFlow and Adrangi to create a more robust and efficient CDN system. A POSITA would have recognized that Adrangi provides a foundation for initial request handling and redirection to a parent server, while Akamai FreeFlow describes parent and edge servers as well as increased stability through failure and fault detection. *See* Adrangi, Abstract, 2:30–58, 9:26–35, 10:59–67, FIGs. 2 & 7 & 13; *e.g.*, Akamai

FreeFlow White Paper at 9. Specifically, a POSITA would combine Adrangi's initial request handling and redirection to a parent server, Adrangi, 9:25-32, 10:50-58, 10:46-65, with Akamai FreeFlow's parent and edge servers as well as increased stability through failure and fault detection, Akamai FreeFlow White Paper at 3-4. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based on factors such as network load, server load, and user location, while using proxies and caches. A POSITA would have recognized that this combination would have led to improved performance, scalability, security, and reliability of the CDN system.

(7) Akamai FreeFlow in view of Lahr

It would have been obvious to combine Akamai FreeFlow and Lahr because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Akamai FreeFlow describes a CDN with hierarchical servers including edge and origin servers, caching, and failure detection. Lahr is directed to a CDN with a tiered hierarchy of servers, including media serving systems (i.e., edge servers), regional data centers, and master data centers.

A POSITA would have been motivated to combine Akamai FreeFlow and Lahr to create a more robust and efficient CDN system. A POSITA would have recognized that Lahr offers a more sophisticated approach to multi-tiered redirection and dynamic server selection, while Akamai FreeFlow describes parent and edge servers as well as increased stability through failure and fault detection. Akamai FreeFlow White Paper at 9; Lahr, Abstract, 3:26-4:3, 5:27-6:11, 20:26-21:2, 22:18-27, FIGs. 1 & 10 & 12. Specifically, a POSITA would combine Akamai FreeFlow's parent and edge servers as well as increased stability through failure and fault detection, Akamai

FreeFlow White Paper at 3–4, with Lahr’s multi-tiered redirection process managed by the “director,” Lahr, 9:27–29, 10:10–13, 19:27–20:2, 20:26–21:2, 21:10–16, 22:18–27, 23:6–16. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based on factors such as network load, server load, and user location. A POSITA would have recognized that this combination would have led to improved performance, scalability, and reliability of the CDN system.

(8) Akamai FreeFlow in view of O’Rourke

It would have been obvious to combine Akamai FreeFlow and O’Rourke because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Akamai FreeFlow describes a CDN with hierarchical servers including edge and origin servers, caching, and failure detection. O’Rourke discloses an environment (CDN) with an origin server and multiple cache servers. Each cache server is used to redistribute media content to local clients.

A POSITA would have been motivated to combine Akamai FreeFlow and O’Rourke to create a more robust and efficient CDN system. A POSITA would have recognized that Akamai FreeFlow describes parent and edge servers as well as increased stability through failure and fault detection, whereas O’Rourke discloses cache servers that utilize policies to determine their function, such as serving content to other cache servers or to clients. *See, e.g.*, Akamai FreeFlow White Paper at 9; O’Rourke, Abstract, ¶¶28–30. Specifically, a POSITA would combine Akamai FreeFlow’s parent and edge servers as well as increased stability through failure and fault detection, Akamai FreeFlow White Paper at 3–4, with O’Rourke’s description of cache server

handling client requests, where when the cache server does not have the requested content available, the system's cache policies determine whether the cache server will download the requested content or perform a redirect. If the system is configured to perform redirection, the cache server redirects the client to another server. O'Rourke also describes that the cache server may redirect the client to another cache server or to an origin server. O'Rourke, ¶¶30, 33, 37, 41, 5, FIG. 5. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based on factors such as network load, server load, and user location. A POSITA would have recognized that this combination would have led to improved performance, scalability, and reliability of the CDN system.

A POSITA would have had a reasonable expectation of success in combining the teachings of Akamai FreeFlow and O'Rourke. The combination would have involved only routine engineering leading to predictable results.

(9) Akamai EdgeSuite in view of Adrangi

It would have been obvious to combine Akamai EdgeSuite and Adrangi because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Adrangi describes a content delivery method where a client request is directed to an edge POP, and if the content is not available, the client is redirected to another server in the content delivery network (CDN). Akamai EdgeSuite describes a CDN with hierarchical servers including edge and origin servers, caching, and failure detection.

A POSITA would have been motivated to combine Akamai EdgeSuite and Adrangi to create a more robust and efficient CDN system. A POSITA would have recognized that Adrangi

provides a foundation for initial request handling and redirection to a parent server, while Akamai EdgeSuite describes parent and edge servers as well as increased stability through failure and fault detection. *See* Adrangi, Abstract, 2:30–58, 9:26–35, 10:59–67, FIGs. 2 & 7 & 13; *e.g.*, Akamai EdgeSuite White Paper at 6. Specifically, a POSITA would combine Adrangi’s initial request handling and redirection to a parent server, Adrangi, 9:25-32, 10:50–58, 10:46–65, with Akamai EdgeSuite’s parent and edge servers as well as increased stability through failure and fault detection, Akamai EdgeSuite White Paper at 6. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based on factors such as network load, server load, and user location, while using proxies and caches. A POSITA would have recognized that this combination would have led to improved performance, scalability, security, and reliability of the CDN system.

(10) Akamai EdgeSuite in view of Lahr

It would have been obvious to combine Akamai FreeEdgeFlow and Lahr because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Akamai EdgeSuite describes a CDN with hierarchical servers including edge and origin servers, caching, and failure detection. Lahr is directed to a CDN with a tiered hierarchy of servers, including media serving systems (*i.e.*, edge servers), regional data centers, and master data centers.

A POSITA would have been motivated to combine Akamai EdgeSuite and Lahr to create a more robust and efficient CDN system. A POSITA would have recognized that Lahr offers a more sophisticated approach to multi-tiered redirection and dynamic server selection, while Akamai EdgeSuite describes parent and edge servers as well as increased stability through failure

and fault detection. Akamai EdgeSuite White Paper at 6; Lahr, Abstract, 3:26–4:3, 5:27–6:11, 20:26–21:2, 22:18–27, FIGs. 1 & 10 & 12. Specifically, a POSITA would combine Akamai EdgeSuite’s parent and edge servers as well as increased stability through failure and fault detection, Akamai EdgeSuite White Paper at 6, with Lahr’s multi-tiered redirection process managed by the “director,” Lahr, 9:27–29, 10:10–13, 19:27–20:2, 20:26–21:2, 21:10–16, 22:18–27, 23:6–16. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based on factors such as network load, server load, and user location. A POSITA would have recognized that this combination would have led to improved performance, scalability, and reliability of the CDN system.

(11) Akamai EdgeSuite in view of O’Rourke

It would have been obvious to combine Akamai EdgeSuite and O’Rourke because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Akamai EdgeSuite describes a CDN with hierarchical servers including edge and origin servers, caching, and failure detection. O’Rourke discloses an environment (CDN) with an origin server and multiple cache servers. Each cache server is used to redistribute media content to local clients.

A POSITA would have been motivated to combine Akamai EdgeSuite and O’Rourke to create a more robust and efficient CDN system. A POSITA would have recognized that Akamai EdgeSuite describes parent and edge servers as well as increased stability through failure and fault detection, whereas O’Rourke discloses cache servers that utilize policies to determine their function, such as serving content to other cache servers or to clients. *See, e.g.*, Akamai EdgeSuite

White Paper at 6; O'Rourke, Abstract, ¶¶28–30. Specifically, a POSITA would combine Akamai EdgeSuite's parent and edge servers as well as increased stability through failure and fault detection, Akamai EdgeSuite White Paper at 6, with O'Rourke's description of cache server handling client requests, where when the cache server does not have the requested content available, the system's cache policies determine whether the cache server will download the requested content or perform a redirect. If the system is configured to perform redirection, the cache server redirects the client to another server. O'Rourke also describes that the cache server may redirect the client to another cache server or to an origin server. O'Rourke, ¶¶30, 33, 37, 41, 5, FIG. 5. This would have resulted in a predictable system where the initial request is efficiently directed to an edge server, and if the content is not available, the request is intelligently routed through multiple tiers of servers based on factors such as network load, server load, and user location. A POSITA would have recognized that this combination would have led to improved performance, scalability, and reliability of the CDN system.

A POSITA would have had a reasonable expectation of success in combining the teachings of Akamai EdgeSuite and O'Rourke. The combination would have involved only routine engineering leading to predictable results.

3. U.S. Patent No. 9,456,053

Microsoft identifies in Appendix C the prior art references that render obvious the Asserted Claims of the '053 Patent and include below exemplary combinations showing the obviousness of the '053 Patent Asserted Claims in view of the prior art. To the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix C, the limitation would have been obvious in light of the disclosures within the reference and the knowledge of one of ordinary skill in the art at the time of the '053 Patent. Moreover, to the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references

described in Appendix C, such reference may be combined with any other references listed in Appendix C for such element, thereby rendering the claims invalid for obviousness. To the extent a finder of fact determines that a limitation of any of the '053 Patent Asserted Claims is not disclosed by one of the references identified above, the claim is nevertheless unpatentable as obvious because they contain nothing that constitutes a patentable innovation. To the extent a finder of fact determines that a limitation of the '053 Patent Asserted Claims is not anticipated, it does not go beyond combining familiar elements according to known methods to achieve predictable results or does more than choose between clear alternatives known to those of ordinary skill in the art.

a. Obviousness

For at least the reasons described below, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix C with any other reference or references listed in Appendix C along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '053 Patent. For example, and without limitation, the Asserted Claims of the '053 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
Akamai Configuration Management System	• Microsoft System Center Configuration Manager • Microsoft Velocity Cache • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> ("Nygren") • U.S. 7,010,578 ("Lewin") • U.S. 8,423,662 ("Weihl") • WO 2000/13456 ("Crosbie") • U.S. 7,240,100 ("Wein") • U.S. 2010/0257258 ("Liu") • U.S. 2007/0156845 ("Devanneaux") • The knowledge of a person of ordinary skill in the art.

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
Microsoft System Center Configuration Manager	• Akamai Configuration Management System • Microsoft Velocity Cache • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”) • U.S. 7,010,578 (“Lewin”) • U.S. 8,423,662 (“Weihl”) • WO 2000/13456 (“Crosbie”) • U.S. 7,240,100 (“Wein”) • U.S. 2010/0257258 (“Liu”) • U.S. 2007/0156845 (“Devanneaux”) • The knowledge of a person of ordinary skill in the art.
Microsoft Velocity Cache	• Akamai Configuration Management System • Microsoft System Center Configuration Manager • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”) • U.S. 7,010,578 (“Lewin”) • U.S. 8,423,662 (“Weihl”) • WO 2000/13456 (“Crosbie”) • U.S. 7,240,100 (“Wein”) • U.S. 2010/0257258 (“Liu”) • U.S. 2007/0156845 (“Devanneaux”) • The knowledge of a person of ordinary skill in the art.
<i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”)	• Akamai Configuration Management System • Microsoft System Center Configuration Manager • Microsoft Velocity Cache • U.S. 7,010,578 (“Lewin”) • U.S. 8,423,662 (“Weihl”) • WO 2000/13456 (“Crosbie”) • U.S. 7,240,100 (“Wein”) • U.S. 2010/0257258 (“Liu”) • U.S. 2007/0156845 (“Devanneaux”) • The knowledge of a person of ordinary skill in the art.
U.S. 7,010,578 (“Lewin”)	• Akamai Configuration Management System • Microsoft System Center Configuration Manager • Microsoft Velocity Cache • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”) • U.S. 8,423,662 (“Weihl”) • WO 2000/13456 (“Crosbie”) • U.S. 7,240,100 (“Wein”) • U.S. 2010/0257258 (“Liu”)

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• U.S. 2007/0156845 (“Devanneaux”) • The knowledge of a person of ordinary skill in the art.
U.S. 8,423,662 (“Weihl”)	• Akamai Configuration Management System • Microsoft System Center Configuration Manager • Microsoft Velocity Cache • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”) • U.S. 7,010,578 (“Lewin”) • WO 2000/13456 (“Crosbie”) • U.S. 7,240,100 (“Wein”) • U.S. 2010/0257258 (“Liu”) • U.S. 2007/0156845 (“Devanneaux”) • The knowledge of a person of ordinary skill in the art.
WO 2000/13456 (“Crosbie”)	• Akamai Configuration Management System • Microsoft System Center Configuration Manager • Microsoft Velocity Cache • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”) • U.S. 7,010,578 (“Lewin”) • U.S. 8,423,662 (“Weihl”) • U.S. 7,240,100 (“Wein”) • U.S. 2010/0257258 (“Liu”) • U.S. 2007/0156845 (“Devanneaux”) • The knowledge of a person of ordinary skill in the art.
U.S. 7,240,100 (“Wein”)	• Akamai Configuration Management System • Microsoft System Center Configuration Manager • Microsoft Velocity Cache • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”) • U.S. 7,010,578 (“Lewin”) • U.S. 8,423,662 (“Weihl”) • WO 2000/13456 (“Crosbie”) • U.S. 7,240,100 (“Wein”) • U.S. 2010/0257258 (“Liu”) • U.S. 2007/0156845 (“Devanneaux”) • The knowledge of a person of ordinary skill in the art.
U.S. 2010/0257258 (“Liu”)	• Akamai Configuration Management System • Microsoft System Center Configuration Manager • Microsoft Velocity Cache • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”)

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• U.S. 7,010,578 (“Lewin”) • U.S. 8,423,662 (“Weihl”) • WO 2000/13456 (“Crosbie”) • U.S. 7,240,100 (“Wein”) • U.S. 2007/0156845 (“Devanneaux”) • The knowledge of a person of ordinary skill in the art.
U.S. 2007/0156845 (“Devanneaux”)	• Akamai Configuration Management System • Microsoft System Center Configuration Manager • Microsoft Velocity Cache • <i>The Akamai Network: A Platform for High-Performance Internet Applications</i> (“Nygren”) • U.S. 7,010,578 (“Lewin”) • U.S. 8,423,662 (“Weihl”) • WO 2000/13456 (“Crosbie”) • U.S. 7,240,100 (“Wein”) • U.S. 2010/0257258 (“Liu”) • The knowledge of a person of ordinary skill in the art.

b. General Motivations to Combine

Obviousness: For at least the reasons described in these contentions, it would have been obvious to one of ordinary skill in the art to combine any of a number of prior art references, including any combination of those prior art references identified in Appendix C along with the knowledge of one of ordinary skill in the art to meet the limitations of the ‘053 Patent Asserted Claims. As a result, the references identified herein render one or more Asserted Claims of the ‘053 Patent obvious when the references are read in combination with each other, and/or when read in view of the state of the art and knowledge of those skilled in the art. Each and every reference identified is also relevant to the state of the art at the time of the alleged invention. Any of the references disclosed above may be combined with one another to render obvious (and therefore invalid) each of Sandpiper’s Asserted Claims. Defendant may rely upon a subset of the identified references or all of the references identified herein, including all references in Appendix

C, for purposes of obviousness depending on the Court's claim construction, positions taken by Sandpiper during this litigation, and further investigation and discovery.

Reservation of Rights: Moreover, as mentioned above, Defendant has not yet completed their search or discovery concerning additional prior art. As such, Defendant's inclusion of exemplary combinations does not preclude them from identifying other invalidating combinations as appropriate, and Defendant reserves the right to identify additional specific combinations as well as to detail and explain such combinations.

No More Than Ordinary Variations of Prior Art; No Unexpected Results: To the extent not anticipated, the '053 Patent Asserted Claims represent no more than the result of ordinary variations of the prior art. Defendant further believes that no showing of a specific motivation to combine prior art is required to combine the references disclosed above and in the attached charts, as each combination of art would have no unexpected results, and at most would simply represent a known alternative to one of ordinary skill in the art. *See KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 415-16 (2007) (rejecting the Federal Circuit's "rigid" application of the teaching, suggestion, or motivation to combine test, instead espousing an "expansive and flexible" approach); *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1058 (Fed. Cir. 2016); *Intercontinental Great Brands LLC v. Kellogg N. Am. Co.*, 869 F.3d 1336, 1344 (Fed. Cir. 2017). Indeed, the Supreme Court held that a person of ordinary skill in the art is "a person of ordinary creativity, not an automaton" and "in many cases a person of ordinary skill in the art will be able to fit the teachings of multiple patents together like pieces of a puzzle." *Id.* at 420-21. Nevertheless, in addition to the information contained elsewhere in these contentions, Defendant identifies motivation and reason to combine the cited art.

Predictable Results; Obvious to Try: One or more combinations of the prior art references identified in Appendix C would have been obvious because these references would have been combined using: known methods to yield predictable results; known techniques in the same way; a simple substitution of one known, equivalent element for another to obtain predictable results; and/or a teaching, suggestion, or motivation in the prior art generally. In addition, it would have been obvious to try combining the prior art references identified above because there were only a finite number of predictable solutions and/or because known work in one field of endeavor prompted variations based on predictable design incentives and/or market forces either in the same field or a different one. Further, the combinations of the prior art references identified in Appendix C would have been obvious because the combinations represent known potential options with a reasonable expectation of success.

Interrelated Teachings: Additional evidence that there would have been a motivation to combine the prior art references identified above includes the interrelated teachings of multiple prior art references; common authorship; the effects of demands known to the design community or present in the marketplace; the existence of a known problem for which there was an obvious solution encompassed by the '053 Patent Asserted Claims; the existence of a known need or problem in the field of the endeavor at the time of the alleged invention(s); and the background knowledge that would have been possessed by a person having ordinary skill in the art.

Summary of Motivations to Combine: Thus, the motivation to combine the teachings of the prior art references disclosed in Appendix C is found in the references themselves and also in: (1) the nature of the problem being solved; (2) the express, implied and inherent teachings of the prior art; (3) the knowledge of persons of ordinary skill in the art; (4) the predictable results obtained in combining the different elements of the prior art; (5) the predictable results obtained

in simple substitution of one known element for another; (6) the use of a known technique to improve similar devices, methods, or products in the same way; (7) the predictable results obtained in applying a known technique to a known device, method, or product ready for improvement; (8) the finite number of identified predictable solutions that had a reasonable expectation of success; and (9) known work in various technological fields that could be applied to the same or different technological fields based on design incentives or other market forces.

Same Area of Technology: Additionally, it would be obvious to one of ordinary skill in the art to consult and/or combine any of the prior art listed in Appendix C because all of these references relate to the same area of technology and/or are from analogous art. The '053 Patent Asserted Claims are directed to “[a] content delivery network (CDN) includes a control core; and a plurality of caches, each of said caches constructed and adapted to: upon joining the CDN, obtain global configuration data from the control core; and obtain data from other caches.” '053 Patent at Abstract. The '053 Patent Asserted Claims merely unite old elements, well known in the field, with no change in their respective function or result. Given the interrelated teachings of the prior art, the effects of demands known to the design community or present in the marketplace, and the background knowledge possessed by a person having ordinary skill in the art, it would have been obvious for one of ordinary skill in the art to combine these familiar elements, disclosed and/or embodied in the prior art listed above to practice the '053 Patent Asserted Claims.

Same Technical Issues: A person of ordinary skill would have been motivated to combine the above prior art based on his/her knowledge, the nature of the problem to be solved, and the teachings of the prior art. The identified prior art addresses the same or similar technical issues and suggests the same or similar solutions to those issues. Moreover, some of the prior art refer to or discuss other prior art, illustrating the close technical relationship among the prior art. For

the '053 Patent, one of ordinary skill in the art would have been motivated to combine known prior art solutions described in these references relating to obtaining global configuration data from a control core, obtaining customer-specific configuration data, and serving resources in accordance with the global configuration data and the customer configuration information associated with the particular customer. For example, there was a need to efficiently manage and distribute configuration data to edge servers to help increase business productivity and organization. The distribution of configuration data needed to be scalable, dynamic, and capable of delivering updates that were both global and customer-specific.

Demand and Market Forces: For example, market forces suggested a demand for increased security and versatility of user authentication systems, and thus one of skill in the art would have thought to combine or modify references that described known methods that one of skill in the art would have recognized as offering improvements to solutions of that time.

Known Elements; Obvious to Try: Moreover, many of the claim elements were already known as admitted by the applicants in the patent specification. These elements represented design choices available to a person of ordinary skill. When there is a design need or market pressure to solve a problem such as identified previously and/or described in the '053 Patent, and there are a finite number of identified, predictable solutions, a person of ordinary skill would be motivated to combine the known options that are within his or her technical grasp. For the '053 Patent, there are a finite number of approaches for expanding the capabilities of proximity systems and location tracking systems to run or interact with different applications. For example, as of the priority date of the '053 Patent, a POSITA would have been very familiar with basic elements of the '053 Patent's claims such as a control core, global configuration data, and customer-specific configuration data. *See* Appx. C. Such elements were within the knowledge of a POSITA, whether

or not disclosed in any particular reference, and the inclusion of such elements in the '053 Patent's claims does not render them novel or non-obvious.

c. Exemplary Combinations

For at least the reasons described above, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix C with any other reference or references listed in Appendix C along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '053 Patent.

For example, and without limitation, the Asserted Claims of the '053 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

(1) Weihl in view of Crosbie

It would have been obvious to combine Weihl and Crosbie because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

A POSITA would have found it obvious at the time of the invention to include Crosbie's known Configuration Version Control System software instructions within Weihl's known CDN metadata control server 118, to expand the existing, centralized control of Weihl's server to also "control and audit...server configuration files" with a predictable increase in network content routing efficiency and effectiveness due to Crosbie's system-pervasive and simple "roll-forward" and "rollback" configuration file version control. *Crosbie*, Figs. 3, 8, 10:31-11:9. Specifically, Weihl's metadata control server delivers metadata to edge cache servers "as necessary." *Weihl*, 4:53-55. Weihl's CDN is therefore predisposed to benefit from Crosbie's express definition of when that is necessary, e.g., "version control needed for releasing new version of...configuration"

and “simplifie[d] change tracking, auditability,” and reverting to a previous version (or, “rollback”) due to a system “misconfiguration.” *Crosbie*, 11:5-9.

Additionally, a POSITA would have recognized the combination of Weihl and Crosbie offers a predictable improvement in network content routing because Crosbie’s system interfaces with a more robust Customer Database including both customer-specific information and a repository of routing test results that informs a geo-traffic manager how to best route requested content to a particular end user (i.e., flesh out the “customer code[s]” in Weihl’s metadata used to provide the correct customer the correct content in the correct way). *Weihl*, 4:27-33; *Crosbie*, 11:10-18, 12:16-23, 12:29-13:3, 14:17-18.

Also, by using Crosbie’s configuration file version check with Weihl’s metadata-based configuration file, Weihl’s system better determines “how to handle” a request and ensure a current request is correctly handled with current configuration file. *Weihl*, 4:44-56; *Crosbie*, 10:31-11:9.

A POSITA would have had a reasonable expectation of success incorporating Crosbie’s configuration version control system into Weihl’s CDN—for example, at its metadata control server—because the two networks operate the same way to achieve a similar result using similar architecture ripe for a predicable improvement. *Compare Weihl*, 4:50-56 (Weihl’s controlling metadata (“global” and “customer configuration data”) is delivered to server 118, then “periodically delivered to the staging servers”) *with Crosbie*, 16:10-17 (Crosbie’s Configuration Version Control System receives system data from, e.g., network data collectors and monitors, prior to distributing routing tables to distributed server nodes). Additionally, both Weihl and Crosbie rely on server manager software to operate the cache and content request routing and load balancing in edge servers. *Compare, e.g., Weihl*, Figs. 2, 3, 4:57-65 (Fig. 2 embodiment); 6:9-49 (Fig. 3 embodiment), *with Crosbie*, Fig. 8 (22), 7:14-21, 11:1-3, 14:1-6, 15:5-6, 18:1-2.

(2) Lewin in view of in view of Weihl, Wein, and/or Nygren

It would have been obvious to combine Lewin, Weihl, Wein, and/or Nygren because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Lewin, Weihl, Wein, and Nygren are publications of Akamai Technologies, Inc. *Compare Lewin*, Cover (73) (assigned to Akamai Technologies, Inc.); *with Weihl*, Cover (73) (same); *Wein* (73) (same); and *Nygren*, Title (“The Akamai Network: A Platform for High-Performance Internet Applications”). All three references disclose similar Akamai content delivery networks, components, and flow processes, differing in focus, scope, and level of detail. *Compare*, e.g., *Lewin*, 1:17-25 (“Akamai operates a global content delivery network (CDN)”); *Weihl*, 1:17-67 (disclosing content delivery networks and their requirements); *Wein*, 2:13-17 (discussing Akamai FREEFLOW); *Nygren*, 2 (“Akamai first pioneered the concept of Content Delivery Networks (CDNs”).

A POSITA would have therefore been motivated to use any one of the features or components of Weihl’s, Wein’s, or Nygren’s CDN to modify or combine with Lewin’s CDN because they all share Akamai CDN components and functionality. For example, all of Lewin, Nygren, Wein, and Weihl expressly disclose content delivery networks (CDNs) that use cache servers. *See, e.g., Weihl*, 3:15-41, 4:57-5:11; *Wein*, 2:13-27; *Nygren*, 2, 4-5; *Lewin*, 1:17-2:12. Caches are placed along the periphery of a network, closest to a user for serving content to a specific user, and are referred to as “edge servers.” *Weihl*, 4:1-12, 6:21-49; *Nygren*, 5, 11-12; *Wein*, 2:6-8, 22:4-6; *Lewin*, Figs. 2, 4, 1:23-25, 8:63-67. A POSITA would have understood that these common networks and their conventional components would have been easily and predictably tuned for implementation according to specific use cases or improvements requiring specific

configurations. *See, e.g., Nygren*, 17-18 (detailing specific use cases of the Akamai network for customer content, streaming, and application delivery).

For example, a POSITA would have found it obvious to use the teachings of Weihl, Wein, and/or Nygren related to customer configuration data in combination with Lewin's teachings. Advantageously, and as a matter of routine engineering by a POSITA, Lewin's third-party cache specifically, and "machines" in Lewin's CDN more generally, may be configured using the specific classes and types of metadata centrally managed in the Weihl and Nygren CDNs. *Lewin*, 1:51-54, 3:46-49, 5:31-53 (disclosing third party cache configuration when joining the CDN); *Weihl*, Fig. 1, 4:24-43 (sending configuring metadata to an edge server to determine how to handle an object request); *Nygren*, 12-13 (listing specific, metadata-configured edge server capabilities). Predictably, such granular control over any of the Lewin, Weihl, or Nygren CDNS ensures edge server "load balancing" (*Weihl*, 4:27-33) and "configuration options . . . provided to tune performance for difference applications and workloads" (*Nygren*, 13).

Further, Nygren describes the key design principles, results, and components of an Akamai CDN informing on the operation of Lewin and Weihl's CDNs. *See, e.g., Nygren*, 6-7 ("System Design Principles"), 7-11 (disclosing problems and Akamai solutions to streaming, content, and application delivery networks), 11- 16 ("Platform Components" of the Akamai platform). And although Lewin describes configuration of its third-party servers and the customer-specific Akamai ARL (*See generally Lewin*), Wein further expands and explains details of both global and customer-centric configuration of servers and the content of the ARL (*See, e.g., Wein*, 9:33-13:30, FIG. 2). *See* cls. 27(A)-27(C).

Lewin, Wein, and Weihl focus on specific aspects of an overall Akamai platform. The combination of aspects of the Lewin, Wein, Weihl and Nygren systems and integrating,

implementing, modifying, or combining components or functions would have been predictable using routine engineering to a well-known and shared common Akamai platform leading to tailored, predictable results in specific implementations of the Akamai platform. *Lewin*, Title, Abstract (describing third party cache interface support to the Akamai CDN); *Wein*, Title, Abstract (describing customer identification of target objects); *Weihl*, Title, Abstract (describing a forward request queueing feature using the Akamai platform network and configuration).

(3) Akamai Configuration Management System in view of Crosbie

It would have been obvious to combine Akamai Configuration Management System and Crosbie because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Crosbie disclosed Configuration Version Control System software instructions, with system-pervasive and simple “roll-forward” and “rollback” configuration file version control. *Crosbie*, Figs. 3, 8, 10:31-11:9. Crosbie expressly defined when it is necessary to deliver metadata to edge cache servers, e.g., “version control needed for releasing new version of...configuration” and “simplifie[d] change tracking, auditability,” and reverting to a previous version (or, “rollback”) due to a system “misconfiguration.” *Crosbie*, 11:5-9.

The Akamai Configuration Management System disclosed a configuration management system for the Akamai Network that was fully functional as early as 2005, providing configuration management to over 15,000 servers deployed in 1200+ different networks in 60+ countries. ACMS at 1-3. The Akamai Configuration Management System, which was built to support customers’ and internal services’ configuration propagation requirements, accepts distributed

submissions of configuration information and disseminates this information to the Akamai CDN.
See id.

A POSITA would have had a reasonable expectation of success incorporating Crosbie's configuration version control system into Akamai's Configuration Management System—because the two networks operate the same way to achieve a similar result using similar architecture ripe for a predicable improvement. *See Crosbie*, 16:10-17 (Crosbie's Configuration Version Control System receives system data from, e.g., network data collectors and monitors, prior to distributing routing tables to distributed server nodes); ACMS at 1-3 (ACMS accepts distributed submissions of configuration information and disseminates this information to the Akamai CDN).

(4) Akamai Configuration Management System in view of
Lewin, Weihl, Wein, and/or Nygren

It would have been obvious to combine Akamai Configuration Management System and Lewin, Weihl, Wein, and/or Nygren because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Lewin, Weihl, Wein, and Nygren are publications of Akamai Technologies, Inc. *Compare Lewin*, Cover (73) (assigned to Akamai Technologies, Inc.); *with Weihl*, Cover (73) (same); *Wein* (73) (same); and *Nygren*, Title ("The Akamai Network: A Platform for High-Performance Internet Applications"). All references disclose similar Akamai content delivery networks, components, and flow processes, differing in focus, scope, and level of detail. *Compare*, e.g., *Lewin*, 1:17-25 ("Akamai operates a global content delivery network (CDN)"); *Weihl*, 1:17-67 (disclosing content delivery networks and their requirements); *Wein*, 2:13-17 (discussing Akamai FREEFLOW); *Nygren*, 2 ("Akamai first pioneered the concept of Content Delivery Networks (CDNs)").

The Akamai Configuration Management System disclosed a configuration management system for the Akamai Network that was fully functional as early as 2005, providing configuration management to over 15,000 servers deployed in 1200+ different networks in 60+ countries. ACMS at 1-3. The Akamai Configuration Management System, which was built to support customers' and internal services' configuration propagation requirements, accepts distributed submissions of configuration information and disseminates this information to the Akamai CDN. *See id.*

A POSITA would have therefore been motivated to use any one of the features or components of Lewin, Wehl, Wein, and/or Nygren with Akamai Configuration Management System because they all share Akamai CDN components and functionality that would have been predictably adapted to certain use cases or improvements for specific implementations. *See, e.g., Nygren*, 17-18 (detailing specific use cases of the Akamai network for customer content, streaming, and application delivery); ACMS at 1-3 (ACMS accepts distributed submissions of configuration information and disseminates this information to the Akamai CDN).

(5) Microsoft System Center Configuration Manager in view of Crosbie

It would have been obvious to combine Microsoft System Center Configuration Manager and Crosbie because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Crosbie disclosed Configuration Version Control System software instructions, with system-pervasive and simple “roll-forward” and “rollback” configuration file version control. *Crosbie*, Figs. 3, 8, 10:31-11:9. Crosbie expressly defined when it is necessary to deliver metadata to edge cache servers, e.g., “version control needed for releasing new version of...configuration”

and “simplifie[d] change tracking, auditability,” and reverting to a previous version (or, “rollback”) due to a system “misconfiguration.” *Crosbie*, 11:5-9.

Microsoft System Center Configuration Manager is a solution for enterprise management services, including software distribution, inventory, software update, configuration management, and operating system deployment. Microsoft System Center Configuration Manager can be used to manage configuration changes and system updates, including performing automated deployments of full server configurations, and deploying required updates during configured Maintenance Windows to ensure SLA compliance while maintaining a fully updated and protected data center. *See, e.g.*, Microsoft System Center at 1, 8-9, 14,16-19.

A POSITA would have had a reasonable expectation of success incorporating Crosbie’s configuration version control system into Microsoft System Center Configuration Manager—because the two networks operate the same way to achieve a similar result using similar architecture ripe for a predicable improvement. *See Crosbie*, 16:10-17 (Crosbie’s Configuration Version Control System receives system data from, e.g., network data collectors and monitors, prior to distributing routing tables to distributed server nodes).

(6) Microsoft System Center Configuration Manager in view of Lewin, Wehl, Wein, and/or Nygren

It would have been obvious to combine Microsoft System Center Configuration Manager and Lewin, Wehl, Wein, and/or Nygren because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft System Center Configuration Manager is a solution for enterprise management services, including software distribution, inventory, software update, configuration management,

and operating system deployment. Microsoft System Center Configuration Manager can be used to manage configuration changes and system updates, including performing automated deployments of full server configurations, and deploying required updates during configured Maintenance Windows to ensure SLA compliance while maintaining a fully updated and protected data center. *See, e.g.,* Microsoft System Center at 1, 8-9, 14,16-19.

Lewin, Weihl, Wein, and Nygren are publications of Akamai Technologies, Inc. *Compare Lewin*, Cover (73) (assigned to Akamai Technologies, Inc.); *with Weihl*, Cover (73) (same); *Wein* (73) (same); and *Nygren*, Title (“The Akamai Network: A Platform for High-Performance Internet Applications”). All references disclose similar Akamai content delivery networks, components, and flow processes, differing in focus, scope, and level of detail. *Compare, e.g., Lewin*, 1:17-25 (“Akamai operates a global content delivery network (CDN)”); *Weihl*, 1:17-67 (disclosing content delivery networks and their requirements); *Wein*, 2:13-17 (discussing Akamai FREEFLOW); *Nygren*, 2 (“Akamai first pioneered the concept of Content Delivery Networks (CDNs”).

A POSITA would have been motivated to combine Microsoft System Center Configuration Manager, with the CDN architecture described in Lewin, Weihl, Wein, and/or Nygren, and expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems and/or configuration management systems.

(7) Microsoft System Center Configuration Manager in view
of Akamai Configuration Management System

It would have been obvious to combine Microsoft System Center Configuration Manager and Akamai Configuration Management System because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the

predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft System Center Configuration Manager is a solution for enterprise management services, including software distribution, inventory, software update, configuration management, and operating system deployment. Microsoft System Center Configuration Manager can be used to manage configuration changes and system updates, including performing automated deployments of full server configurations, and deploying required updates during configured Maintenance Windows to ensure SLA compliance while maintaining a fully updated and protected data center. *See, e.g.,* Microsoft System Center at 1, 8-9, 14,16-19.

The Akamai Configuration Management System disclosed a configuration management system for the Akamai Network that was fully functional as early as 2005, providing configuration management to over 15,000 servers deployed in 1200+ different networks in 60+ countries. ACMS at 1-3. The Akamai Configuration Management System, which was built to support customers' and internal services' configuration propagation requirements, accepts distributed submissions of configuration information and disseminates this information to the Akamai CDN. *See id.*

A POSITA would have been motivated to combine Microsoft SCCM with the CDN architecture described in Akamai Configuration Management System, and expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems and/or configuration management systems.

(8) Microsoft System Center Configuration Manager in view of Microsoft Velocity

It would have been obvious to combine Microsoft System Center Configuration Manager and Microsoft Velocity because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft System Center Configuration Manager is a solution for enterprise management services, including software distribution, inventory, software update, configuration management, and operating system deployment. Microsoft System Center Configuration Manager can be used to manage configuration changes and system updates, including performing automated deployments of full server configurations, and deploying required updates during configured Maintenance Windows to ensure SLA compliance while maintaining a fully updated and protected data center. *See, e.g.,* Microsoft System Center at 1, 8-9, 14,16-19.

Microsoft Velocity Cache is a distributed, in-memory cache that exposes a unified cache tier to client applications by fusing together memory across servers. Velocity implements a configuration store, which consists of cache policies as well as the global partition map, which details the distribution of cache items across the cluster. Velocity uses the policies and global partition map defined within the configuration store to enforce the policies defined by named caches and to run the global partition manager, respectively. *See, e.g.,* MSDN Magazine at 2, 4-8.

A POSITA would have been motivated to combine Microsoft SCCM with Velocity and expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and

skill, to combine known components of compatible CDN systems and/or configuration management systems.

(9) Microsoft Velocity Cache in view of Crosbie

It would have been obvious to combine Microsoft Velocity Cache and Crosbie because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Crosbie disclosed Configuration Version Control System software instructions, with system-pervasive and simple “roll-forward” and “rollback” configuration file version control. *Crosbie*, Figs. 3, 8, 10:31-11:9. Crosbie expressly defined when it is necessary to deliver metadata to edge cache servers, e.g., “version control needed for releasing new version of...configuration” and “simplifie[d] change tracking, auditability,” and reverting to a previous version (or, “rollback”) due to a system “misconfiguration.” *Crosbie*, 11:5-9.

Microsoft Velocity Cache is a distributed, in-memory cache that exposes a unified cache tier to client applications by fusing together memory across servers. Velocity implements a configuration store, which consists of cache policies as well as the global partition map, which details the distribution of cache items across the cluster. Velocity uses the policies and global partition map defined within the configuration store to enforce the policies defined by named caches and to run the global partition manager, respectively. *See, e.g.*, MSDN Magazine at 2, 4-8.

A POSITA would have had a reasonable expectation of success incorporating Crosbie’s configuration version control system into Microsoft Velocity Cache—because the two networks operate the same way to achieve a similar result using similar architecture ripe for a predicable improvement. *See Crosbie*, 16:10-17 (Crosbie’s Configuration Version Control System receives

system data from, e.g., network data collectors and monitors, prior to distributing routing tables to distributed server nodes).

(10) Microsoft Velocity Cache in view of Lewin, Weihl, Wein, and/or Nygren

It would have been obvious to combine Microsoft Velocity and Lewin, Weihl, Wein, and/or Nygren because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft Velocity Cache is a distributed, in-memory cache that exposes a unified cache tier to client applications by fusing together memory across servers. Velocity implements a configuration store, which consists of cache policies as well as the global partition map, which details the distribution of cache items across the cluster. Velocity uses the policies and global partition map defined within the configuration store to enforce the policies defined by named caches and to run the global partition manager, respectively. *See, e.g.*, MSDN Magazine at 2, 4-8.

Lewin, Weihl, Wein, and Nygren are publications of Akamai Technologies, Inc. *Compare Lewin*, Cover (73) (assigned to Akamai Technologies, Inc.); *with Weihl*, Cover (73) (same); *Wein* (73) (same); and *Nygren*, Title (“The Akamai Network: A Platform for High-Performance Internet Applications”). All references disclose similar Akamai content delivery networks, components, and flow processes, differing in focus, scope, and level of detail. *Compare, e.g.*, *Lewin*, 1:17-25 (“Akamai operates a global content delivery network (CDN)”); *Weihl*, 1:17-67 (disclosing content delivery networks and their requirements); *Wein*, 2:13-17 (discussing Akamai FREEFLOW); *Nygren*, 2 (“Akamai first pioneered the concept of Content Delivery Networks (CDNs”).

A POSITA would have been motivated to combine Microsoft Velocity with the CDN architecture described in Lewin, Wehl, Wein, and/or Nygren, and expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems and/or configuration management systems.

(11) Microsoft Velocity Cache in view of Akamai Configuration Management System

It would have been obvious to combine Microsoft Velocity and Akamai Configuration Management System because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Microsoft Velocity Cache is a distributed, in-memory cache that exposes a unified cache tier to client applications by fusing together memory across servers. Velocity implements a configuration store, which consists of cache policies as well as the global partition map, which details the distribution of cache items across the cluster. Velocity uses the policies and global partition map defined within the configuration store to enforce the policies defined by named caches and to run the global partition manager, respectively. *See, e.g.*, MSDN Magazine at 2, 4-8.

The Akamai Configuration Management System disclosed a configuration management system for the Akamai Network that was fully functional as early as 2005, providing configuration management to over 15,000 servers deployed in 1200+ different networks in 60+ countries. ACMS at 1-3. The Akamai Configuration Management System, which was built to support customers' and internal services' configuration propagation requirements, accepts distributed

submissions of configuration information and disseminates this information to the Akamai CDN.
See id.

A POSITA would have been motivated to combine Velocity with the CDN architecture implementation of Akamai Configuration Management System and expand the capabilities of the combined system. The POSITA would expect to be able to successfully implement such a system through routine application of conventional knowledge and skill, to combine known components of compatible CDN systems and/or configuration management systems.

4. U.S. Patent No. 9,722,883

Microsoft identifies in Appendix D the prior art references that render obvious the Asserted Claims of the '883 Patent and include below exemplary combinations showing the obviousness of the '883 Patent Asserted Claims in view of the prior art. To the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix D, the limitation would have been obvious in light of the disclosures within the reference and the knowledge of one of ordinary skill in the art at the time of the '883 Patent. Moreover, to the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix D, such reference may be combined with any other references listed in Appendix D for such element, thereby rendering the claims invalid for obviousness. To the extent a finder of fact determines that a limitation of any of the '883 Patent Asserted Claims is not disclosed by one of the references identified above, the claim is nevertheless unpatentable as obvious because they contain nothing that constitutes a patentable innovation. To the extent a finder of fact determines that a limitation of the '883 Patent Asserted Claims is not anticipated, it does not go beyond combining familiar elements according to known methods to achieve predictable results or does more than choose between clear alternatives known to those of ordinary skill in the art.

a. Obviousness

For at least the reasons described below, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix D with any other reference or references listed in Appendix D along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '883 Patent. For example, and without limitation, the Asserted Claims of the '883 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
Roadknight	• Smith • Wang • Jennings • IBM Load Balancer • Lync Server 2013 • Windows Fabric • Velocity Cache • The knowledge of a person of ordinary skill in the art.
Smith	• Roadknight • Wang • Jennings • IBM Load Balancer • Lync Server 2013 • Windows Fabric • Velocity Cache • The knowledge of a person of ordinary skill in the art.
Wang	• Roadknight • Smith • Jennings • IBM Load Balancer • Lync Server 2013 • Windows Fabric • Velocity Cache • The knowledge of a person of ordinary skill in the art.
Jennings	• Roadknight • Smith • Wang • IBM Load Balancer

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Lync Server 2013 • Windows Fabric • Velocity Cache • The knowledge of a person of ordinary skill in the art.
IBM Load Balancer	• Roadknight • Smith • Wang • Jennings • Lync Server 2013 • Windows Fabric • Velocity Cache • The knowledge of a person of ordinary skill in the art.
Lync Server 2013	• Roadknight • Smith • Wang • Jennings • IBM Load Balancer • Windows Fabric • Velocity Cache • The knowledge of a person of ordinary skill in the art.
Windows Fabric	• Roadknight • Smith • Wang • Jennings • IBM Load Balancer • Lync Server 2013 • Velocity Cache • The knowledge of a person of ordinary skill in the art.
Velocity Cache	• Roadknight • Smith • Wang • Jennings • IBM Load Balancer • Lync Server 2013 • Windows Fabric • The knowledge of a person of ordinary skill in the art.

b. General Motivations to Combine

Obviousness: For at least the reasons described in these contentions, it would have been obvious to one of ordinary skill in the art to combine any of a number of prior art references, including any combination of those prior art references identified in Appendix D along with the knowledge of one of ordinary skill in the art to meet the limitations of the ‘883 Patent Asserted Claims. As a result, the references identified herein render one or more Asserted Claims of the ‘883 Patent obvious when the references are read in combination with each other, and/or when read in view of the state of the art and knowledge of those skilled in the art. Each and every reference identified is also relevant to the state of the art at the time of the alleged invention. Any of the references disclosed above may be combined with one another to render obvious (and therefore invalid) each of Sandpiper’s Asserted Claims. Defendant may rely upon a subset of the identified references or all of the references identified herein, including all references in Appendix D, for purposes of obviousness depending on the Court’s claim construction, positions taken by Sandpiper during this litigation, and further investigation and discovery.

Reservation of Rights: Moreover, as mentioned above, Defendant has not yet completed their search or discovery concerning additional prior art. As such, Defendant’s inclusion of exemplary combinations does not preclude them from identifying other invalidating combinations as appropriate, and Defendant reserves the right to identify additional specific combinations as well as to detail and explain such combinations.

No More Than Ordinary Variations of Prior Art; No Unexpected Results: To the extent not anticipated, the ‘883 Patent Asserted Claims represent no more than the result of ordinary variations of the prior art. Defendant further believes that no showing of a specific motivation to combine prior art is required to combine the references disclosed above and in the attached charts, as each combination of art would have no unexpected results, and at most would

simply represent a known alternative to one of ordinary skill in the art. *See KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 415-16 (2007) (rejecting the Federal Circuit's "rigid" application of the teaching, suggestion, or motivation to combine test, instead espousing an "expansive and flexible" approach); *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1058 (Fed. Cir. 2016); *Intercontinental Great Brands LLC v. Kellogg N. Am. Co.*, 869 F.3d 1336, 1344 (Fed. Cir. 2017). Indeed, the Supreme Court held that a person of ordinary skill in the art is "a person of ordinary creativity, not an automaton" and "in many cases a person of ordinary skill in the art will be able to fit the teachings of multiple patents together like pieces of a puzzle." *Id.* at 420-21. Nevertheless, in addition to the information contained elsewhere in these contentions, Defendant identifies motivation and reason to combine the cited art.

Predictable Results; Obvious to Try: One or more combinations of the prior art references identified in Appendix D would have been obvious because these references would have been combined using: known methods to yield predictable results; known techniques in the same way; a simple substitution of one known, equivalent element for another to obtain predictable results; and/or a teaching, suggestion, or motivation in the prior art generally. In addition, it would have been obvious to try combining the prior art references identified above because there were only a finite number of predictable solutions and/or because known work in one field of endeavor prompted variations based on predictable design incentives and/or market forces either in the same field or a different one. Further, the combinations of the prior art references identified in Appendix D would have been obvious because the combinations represent known potential options with a reasonable expectation of success.

Interrelated Teachings: Additional evidence that there would have been a motivation to combine the prior art references identified above includes the interrelated teachings of multiple

prior art references; common authorship; the effects of demands known to the design community or present in the marketplace; the existence of a known problem for which there was an obvious solution encompassed by the '883 Patent Asserted Claims; the existence of a known need or problem in the field of the endeavor at the time of the alleged invention(s); and the background knowledge that would have been possessed by a person having ordinary skill in the art.

Summary of Motivations to Combine: Thus, the motivation to combine the teachings of the prior art references disclosed in Appendix D is found in the references themselves and also in: (1) the nature of the problem being solved; (2) the express, implied and inherent teachings of the prior art; (3) the knowledge of persons of ordinary skill in the art; (4) the predictable results obtained in combining the different elements of the prior art; (5) the predictable results obtained in simple substitution of one known element for another; (6) the use of a known technique to improve similar devices, methods, or products in the same way; (7) the predictable results obtained in applying a known technique to a known device, method, or product ready for improvement; (8) the finite number of identified predictable solutions that had a reasonable expectation of success; and (9) known work in various technological fields that could be applied to the same or different technological fields based on design incentives or other market forces.

Same Area of Technology: Additionally, it would be obvious to one of ordinary skill in the art to consult and/or combine any of the prior art listed in Appendix D because all of these references relate to the same area of technology and/or are from analogous art. The '883 Patent Asserted Claims are directed to a network comprising multiple service endpoints. '883 Patent at Abstract. The '883 Patent Asserted Claims merely unite old elements, well known in the field, with no change in their respective function or result. Given the interrelated teachings of the prior art, the effects of demands known to the design community or present in the marketplace, and the

background knowledge possessed by a person having ordinary skill in the art, it would have been obvious for one of ordinary skill in the art to combine these familiar elements, disclosed and/or embodied in the prior art listed above to practice the ‘883 Patent Asserted Claims.

Same Technical Issues: A person of ordinary skill would have been motivated to combine the above prior art based on his/her knowledge, the nature of the problem to be solved, and the teachings of the prior art. The identified prior art addresses the same or similar technical issues and suggests the same or similar solutions to those issues. Moreover, some of the prior art refer to or discuss other prior art, illustrating the close technical relationship among the prior art. For the ‘883 Patent, one of ordinary skill in the art would have been motivated to combine known prior art solutions described in these references relating to optimizing the handling of requests in a network of endpoints according to a predetermined policy. The motivation lies in increasing performance and optimizing which endpoint handles a given task—recognized benefits in network design. Using a policy to determine which node should handle a request was a predictable solution, as the techniques was widely documented and interoperable. It would have been further obvious to direct requests to the appropriate endpoint and to process the request according to the policy. The motivation to combine stems from the desire to efficiently handle requests. POSITA would have had a reasonable expectation of success because popularity-based algorithms were routinely applied to networks of endpoints.

It would have been obvious to combine or modify each reference with another reference identified in Sections IV.D to render these claims invalid because each teaches complementary aspects of network design. A person of ordinary skill in the art (“POSITA”) would have been motivated to combine these teachings to improve scalability efficiency—well-recognized design goals in networked systems. Using policy based processing would have been a predictable use of

prior art elements according to their established functions, yielding no unexpected results. Market demand for policy based processing and optimization would have provided further motivation. POSITA would have had a reasonable expectation of success because these techniques were routinely implemented in distributed systems at the time.

Demand and Market Forces: For example, market forces suggested a demand for increased security and versatility of user authentication systems, and thus one of skill in the art would have thought to combine or modify references that described known methods that one of skill in the art would have recognized as offering improvements to solutions of that time.

Known Elements; Obvious to Try: Moreover, many of the claim elements were already known as admitted by the applicants in the patent specification. These elements represented design choices available to a person of ordinary skill. When there is a design need or market pressure to solve a problem such as identified previously and/or described in the ‘883 Patent, and there are a finite number of identified, predictable solutions, a person of ordinary skill would be motivated to combine the known options that are within his or her technical grasp. For the ‘883 Patent, there are a finite number of approaches for expanding the capabilities of proximity systems and location tracking systems to run or interact with different applications. For example, as of the priority date of the ‘883 Patent, a POSITA would have been very familiar with basic elements of the ‘883 Patent’s claims such as processing according to a policy. *See* Appx. D. Such elements were within the knowledge of a POSITA, whether or not disclosed in any particular reference, and the inclusion of such elements in the ‘883 Patent’s claims does not render them novel or non-obvious.

c. Exemplary Combinations

For at least the reasons described above, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix D with any other reference

or references listed in Appendix D along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '883 Patent.

For example, and without limitation, the Asserted Claims of the '883 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

- (1) Roadknight in view of Smith, Wang, Jennings, and/or IBM Load Balancer

It would have been obvious to combine Roadknight with Smith, Wang, Jennings, and/or IBM Load Balancer because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

A POSITA would have understood that the exact configuration of a network—such as (1) the intended operation of a network (e.g., e-mail providers, content-delivery providers, distributed processing services, or shared storage services), (2) arrangements of network nodes, (3) how tasks are determined, assigned, shared, or distributed between nodes, (4) the rules each node applies in carrying out their tasks, each had well-known options that were known to be mere design choices based on factors such as their intended purpose, complexity, latency, and redundancy. Modifying Roadknight to adopt specific configuration details taught by Smith, Wang, Jennings, IBM Load Balancer, and/or a POSITA's general understanding of the art would have merely amounted to a design choice, a simple substitution of one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Roadknight describes a generic network configuration to enable network nodes to automatically and adaptively update their operating policies based on demand/utilization, (see generally Exhibit D-04 [Roadknight]), a POSITA would have understood that Roadknight's teachings would apply to a variety of specific network applications—such as

distributed processing networks (like IBM Load Balancer) that involved different types of resources and network elements of different capacities.

In a content-delivery network like Smith's that provides different types of nodes assigned different tasks (e.g., storing specific portions of a content library, checking licenses, seeding content, proxying), for instance, Roadknight's network configuration would enable network elements to automatically change what functions each network element served (e.g., storing content versus seeding content versus checking licenses), or even what portions of content is stored in storage elements based on the capacity and demands of the network, as taught by Roadknight. Similarly, in distributed processing networks like IBM Load Balancer, nodes would adjust their policies based on their own capacity and demands of the network to take on specific processing tasks. A POSITA would have appreciated that these modifications would have merely required taking Roadknight's existing automatic and adaptive policy updating with a specific type of network.

(2) Smith in view of Wang and/or Jennings

It would have been obvious to combine Smith with Wang, and/or Jennings because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

A POSITA would have understood that the exact configuration of a network—such as (1) the intended operation of a network (e.g., e-mail providers, content-delivery providers, distributed processing services, or shared storage services), (2) arrangements of network nodes, (3) how tasks are determined, assigned, shared, or distributed between nodes, (4) the rules each node applies in carrying out their tasks, each had well-known options that were known to be mere design choices based on factors such as their intended purpose, complexity, latency, and redundancy. Modifying

Smith to adopt specific configuration details taught by Wang, Jennings, and/or a POSITA's general understanding of the art would have merely amounted to a design choice, a simple substitution of one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Smith describes a network of distributed caching arrays of nodes in which different nodes are responsible for caching different resources (see generally Exhibit D-05 [Smith]), a POSITA would have recognized that other prior art (such as Wang) provides detailed algorithms similar to Smith's to also distribute resources to and locate resources in different caching arrays. A POSITA would have been motivated to, for instance, configure Smith's nodes to determine what kinds of responsibilities other nodes have for a resource (or when no node has responsibility for a resource) based on Wang's teachings. See, e.g., Exhibit D-06 [Wang], cl. 1[pre][ii]-[iii], 1[b], 5-7, 9, 19. In doing so, a POSITA would have appreciated that a node can determine when no nodes in an array are responsible for a resource to acquire a copy from the origin faster, rather than searching for nodes in the array or directing requests to different nodes. As another example, a POSITA would have been motivated to configure Smith's arrays with routing tables that map routing rules to resource types, like Jennings, to avoid needing to generate or search for resources if its resource type is not present in the routing table, producing faster determinations of whether a resource is locally available or not. Jennings, 1:68-2:6; see, e.g., Exhibit D-02 [Jennings], cl. 1[pre][ii]-[iii], 1[b], 5-7, 9, 19. These amount to merely applying known techniques to Smith's prior-art device to achieve predictable results, for instance.

(3) Roadknight in view of Lync Server 2013

It would have been obvious to combine Roadknight with Lync Server 2013 because the combination does nothing more than combine prior art elements from similar systems according

to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

A POSITA would have understood that the exact configuration of a network—such as (1) the intended operation of a network (e.g., e-mail providers, content-delivery providers, distributed processing services, or shared storage services), (2) arrangements of network nodes, (3) how tasks are determined, assigned, shared, or distributed between nodes, (4) the rules each node applies in carrying out their tasks, each had well-known options that were known to be mere design choices based on factors such as their intended purpose, complexity, latency, and redundancy. Modifying Roadknight to adopt specific configuration details taught by Lync Server 2013, and/or a POSITA's general understanding of the art would have merely amounted to a design choice, a simple substitution of one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Roadknight describes a generic network configuration to enable network nodes to automatically and adaptively update their operating policies based on demand/utilization, (*see generally* Exhibit D-04 [Roadknight]), a POSITA would have understood that Roadknight's teachings would apply to a variety of specific network applications—such as distributed processing networks (like Lync Server 2013) that involved different types of resources and network elements of different capacities.

In distributed processing networks like Lync Server 2013, nodes would adjust their policies based on their own capacity and demands of the network to take on specific processing tasks. A POSITA would have appreciated that these modifications would have merely required taking Roadknight's existing automatic and adaptive policy updating with a specific type of network.

(4) Roadknight in view of Velocity Cache

It would have been obvious to combine Roadknight with Velocity Cache because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

A POSITA would have understood that the exact configuration of a network—such as (1) the intended operation of a network (e.g., e-mail providers, content-delivery providers, distributed processing services, or shared storage services), (2) arrangements of network nodes, (3) how tasks are determined, assigned, shared, or distributed between nodes, (4) the rules each node applies in carrying out their tasks, each had well-known options that were known to be mere design choices based on factors such as their intended purpose, complexity, latency, and redundancy. Modifying Roadknight to adopt specific configuration details taught by Velocity Cache, and/or a POSITA's general understanding of the art would have merely amounted to a design choice, a simple substitution of one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Roadknight describes a generic network configuration to enable network nodes to automatically and adaptively update their operating policies based on demand/utilization, (*see generally* Exhibit D-04 [Roadknight]), a POSITA would have understood that Roadknight's teachings would apply to a variety of specific network applications—such as distributed processing networks (like Velocity Cache) that involved different types of resources and network elements of different capacities.

In distributed processing networks like Velocity Cache, nodes would adjust their policies based on their own capacity and demands of the network to take on specific processing tasks. A

POSITA would have appreciated that these modifications would have merely required taking Roadknight's existing automatic and adaptive policy updating with a specific type of network.

(5) Smith in view of Lync Server 2013

It would have been obvious to combine Smith with Lync Server 2013 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

A POSITA would have understood that the exact configuration of a network—such as (1) the intended operation of a network (e.g., e-mail providers, content-delivery providers, distributed processing services, or shared storage services), (2) arrangements of network nodes, (3) how tasks are determined, assigned, shared, or distributed between nodes, (4) the rules each node applies in carrying out their tasks, each had well-known options that were known to be mere design choices based on factors such as their intended purpose, complexity, latency, and redundancy. Modifying Smith to adopt specific configuration details taught by Lync Server 2013 and/or a POSITA's general understanding of the art would have merely amounted to a design choice, a simple substitution of one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Smith describes a network of distributed caching arrays of nodes in which different nodes are responsible for caching different resources (*see generally* Exhibit D-05 [Smith]), a POSITA would have recognized that other prior art (such as Lync Server 2013) provides detailed architectures similar to Smith's to also distribute resources to and locate resources in different caching arrays. A POSITA would have been motivated to, for instance, configure Smith's nodes to determine what kinds of responsibilities *other* nodes have for a resource (or when no node has responsibility for a resource) based on Lync Server 2013's

teachings. *See, e.g.*, Exhibit D-07 [Lync Server 2013], cl. 1[pre][ii]-[iii], 1[b], 5–7, 9, 19. In doing so, a POSITA would have appreciated that a node can determine when no nodes in an array are responsible for a resource to acquire a copy from the origin faster, rather than searching for nodes in the array or directing requests to different nodes. These amount to merely applying known techniques to Smith’s prior-art device to achieve predictable results, for instance.

(6) Smith in view of Velocity Cache

It would have been obvious to combine Smith with Velocity Cache because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

A POSITA would have understood that the exact configuration of a network—such as (1) the intended operation of a network (e.g., e-mail providers, content-delivery providers, distributed processing services, or shared storage services), (2) arrangements of network nodes, (3) how tasks are determined, assigned, shared, or distributed between nodes, (4) the rules each node applies in carrying out their tasks, each had well-known options that were known to be mere design choices based on factors such as their intended purpose, complexity, latency, and redundancy. Modifying Smith to adopt specific configuration details taught by Velocity Cache and/or a POSITA’s general understanding of the art would have merely amounted to a design choice, a simple substitution of one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Smith describes a network of distributed caching arrays of nodes in which different nodes are responsible for caching different resources (*see generally* Exhibit D-05 [Smith]), a POSITA would have recognized that other prior art (such as Velocity Cache) provides detailed architectures similar to Smith’s to also distribute resources to and locate

resources in different caching arrays. A POSITA would have been motivated to, for instance, configure Smith's nodes to determine what kinds of responsibilities *other* nodes have for a resource (or when no node has responsibility for a resource) based on Velocity Cache's teachings. *See, e.g.*, Exhibit D-09 [Velocity Cache], cl. 1[pre][ii]-[iii], 1[b], 5–7, 9, 19. In doing so, a POSITA would have appreciated that a node can determine when no nodes in an array are responsible for a resource to acquire a copy from the origin faster, rather than searching for nodes in the array or directing requests to different nodes. These amount to merely applying known techniques to Smith's prior-art device to achieve predictable results, for instance.

5. U.S. Patent No. 9,762,692

Microsoft identifies in Appendix E the prior art references that render obvious the Asserted Claims of the '692 Patent and include below exemplary combinations showing the obviousness of the '692 Patent Asserted Claims in view of the prior art. To the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix E, the limitation would have been obvious in light of the disclosures within the reference and the knowledge of one of ordinary skill in the art at the time of the '692 Patent. Moreover, to the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix E, such reference may be combined with any other references listed in Appendix E for such element, thereby rendering the claims invalid for obviousness. To the extent a finder of fact determines that a limitation of any of the '692 Patent Asserted Claims is not disclosed by one of the references identified above, the claim is nevertheless unpatentable as obvious because they contain nothing that constitutes a patentable innovation. To the extent a finder of fact determines that a limitation of the '692 Patent Asserted Claims is not anticipated, it does not go beyond combining familiar elements according to known methods to achieve

predictable results or does more than choose between clear alternatives known to those of ordinary skill in the art.

a. Obviousness

For at least the reasons described below, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix E with any other reference or references listed in Appendix E along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '692 Patent. For example, and without limitation, the Asserted Claims of the '692 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
Dilley	• Karger • Seed • Swildens • Windows Media Services • Squid • Akamai EdgeSuite • Kovacs • Roth The knowledge of a person of ordinary skill in the art.
Karger	• Dilley • Seed • Swildens • Windows Media Services • Squid • Akamai EdgeSuite • Kovacs • Roth The knowledge of a person of ordinary skill in the art.
Seed	• Dilley • Karger • Swildens • Windows Media Services • Squid • Akamai EdgeSuite

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Kovacs • Roth The knowledge of a person of ordinary skill in the art.
Swildens	• Dilley • Karger • Seed • Windows Media Services • Squid • Akamai EdgeSuite • Kovacs • Roth • The knowledge of a person of ordinary skill in the art.
Windows Media Services	• Dilley • Karger • Seed • Swildens • Squid • Akamai EdgeSuite • Kovacs • Roth • The knowledge of a person of ordinary skill in the art
Squid	• Dilley • Karger • Seed • Swildens • Windows Media Services • Akamai EdgeSuite • Kovacs • Roth • The knowledge of a person of ordinary skill in the art
Akamai EdgeSuite	• Dilley • Karger • Seed • Swildens • Windows Media Services • Squid • Kovacs • Roth • The knowledge of a person of ordinary skill in the art
Kovacs	• Dilley • Karger

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Seed • Swildens • Windows Media Services • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Roth	• Dilley • Karger • Seed • Swildens • Windows Media Services • Squid • Akamai EdgeSuite • Kovacs • The knowledge of a person of ordinary skill in the art

b. General Motivations to Combine

Obviousness: For at least the reasons described in these contentions, it would have been obvious to one of ordinary skill in the art to combine any of a number of prior art references, including any combination of those prior art references identified in Appendix E along with the knowledge of one of ordinary skill in the art to meet the limitations of the ‘692 Patent Asserted Claims. As a result, the references identified herein render one or more Asserted Claims of the ‘692 Patent obvious when the references are read in combination with each other, and/or when read in view of the state of the art and knowledge of those skilled in the art. Each and every reference identified is also relevant to the state of the art at the time of the alleged invention. Any of the references disclosed above may be combined with one another to render obvious (and therefore invalid) each of Sandpiper’s Asserted Claims. Defendant may rely upon a subset of the identified references or all of the references identified herein, including all references in Appendix

E, for purposes of obviousness depending on the Court's claim construction, positions taken by Sandpiper during this litigation, and further investigation and discovery.

Reservation of Rights: Moreover, as mentioned above, Defendant has not yet completed their search or discovery concerning additional prior art. As such, Defendant's inclusion of exemplary combinations does not preclude them from identifying other invalidating combinations as appropriate, and Defendant reserves the right to identify additional specific combinations as well as to detail and explain such combinations.

No More Than Ordinary Variations of Prior Art; No Unexpected Results: To the extent not anticipated, the '692 Patent Asserted Claims represent no more than the result of ordinary variations of the prior art. Defendant further believes that no showing of a specific motivation to combine prior art is required to combine the references disclosed above and in the attached charts, as each combination of art would have no unexpected results, and at most would simply represent a known alternative to one of ordinary skill in the art. *See KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 415-16 (2007) (rejecting the Federal Circuit's "rigid" application of the teaching, suggestion, or motivation to combine test, instead espousing an "expansive and flexible" approach); *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1058 (Fed. Cir. 2016); *Intercontinental Great Brands LLC v. Kellogg N. Am. Co.*, 869 F.3d 1336, 1344 (Fed. Cir. 2017). Indeed, the Supreme Court held that a person of ordinary skill in the art is "a person of ordinary creativity, not an automaton" and "in many cases a person of ordinary skill in the art will be able to fit the teachings of multiple patents together like pieces of a puzzle." *Id.* at 420-21. Nevertheless, in addition to the information contained elsewhere in these contentions, Defendant identifies motivation and reason to combine the cited art.

Predictable Results; Obvious to Try: One or more combinations of the prior art references identified in Appendix E would have been obvious because these references would have been combined using: known methods to yield predictable results; known techniques in the same way; a simple substitution of one known, equivalent element for another to obtain predictable results; and/or a teaching, suggestion, or motivation in the prior art generally. In addition, it would have been obvious to try combining the prior art references identified above because there were only a finite number of predictable solutions and/or because known work in one field of endeavor prompted variations based on predictable design incentives and/or market forces either in the same field or a different one. For example, incorporating hashing-based lookup in a tiered CDN would have been obvious because hashing was a well-known technique for deterministic resource mapping. Such a method would provide predictable routing and load distribution—recognized goals in distributed systems. POSITA would have expected success because hashing was widely implemented in similar contexts. Further, the combinations of the prior art references identified in Appendix E would have been obvious because the combinations represent known potential options with a reasonable expectation of success.

Interrelated Teachings: Additional evidence that there would have been a motivation to combine the prior art references identified above includes the interrelated teachings of multiple prior art references; common authorship; the effects of demands known to the design community or present in the marketplace; the existence of a known problem for which there was an obvious solution encompassed by the '692 Patent Asserted Claims; the existence of a known need or problem in the field of the endeavor at the time of the alleged invention(s); and the background knowledge that would have been possessed by a person having ordinary skill in the art.

Summary of Motivations to Combine: Thus, the motivation to combine the teachings of the prior art references disclosed in Appendix E is found in the references themselves and also in: (1) the nature of the problem being solved; (2) the express, implied and inherent teachings of the prior art; (3) the knowledge of persons of ordinary skill in the art; (4) the predictable results obtained in combining the different elements of the prior art; (5) the predictable results obtained in simple substitution of one known element for another; (6) the use of a known technique to improve similar devices, methods, or products in the same way; (7) the predictable results obtained in applying a known technique to a known device, method, or product ready for improvement; (8) the finite number of identified predictable solutions that had a reasonable expectation of success; and (9) known work in various technological fields that could be applied to the same or different technological fields based on design incentives or other market forces.

Same Area of Technology: Additionally, it would be obvious to one of ordinary skill in the art to consult and/or combine any of the prior art listed in Appendix E because all of these references relate to the same area of technology and/or are from analogous art. The ‘692 Patent Asserted Claims are directed to a “content delivery network.” ‘692 Patent at Abstract. The ‘692 Patent Asserted Claims merely unite old elements, well known in the field, with no change in their respective function or result. Given the interrelated teachings of the prior art, the effects of demands known to the design community or present in the marketplace, and the background knowledge possessed by a person having ordinary skill in the art, it would have been obvious for one of ordinary skill in the art to combine these familiar elements, disclosed and/or embodied in the prior art listed above to practice the ‘692 Patent Asserted Claims.

Same Technical Issues: A person of ordinary skill would have been motivated to combine the above prior art based on his/her knowledge, the nature of the problem to be solved, and the

teachings of the prior art. The identified prior art addresses the same or similar technical issues and suggests the same or similar solutions to those issues. Moreover, some of the prior art refer to or discuss other prior art, illustrating the close technical relationship among the prior art. For the ‘692 Patent, one of ordinary skill in the art would have been motivated to combine known prior art solutions described in these references relating to optimizing content distribution based on the usage patterns of content. The motivation lies in reducing bandwidth costs and improving hit ratios—recognized benefits in CDN design. Combining popularity metrics with tiered caching was a predictable solution, as both techniques were widely documented and interoperable. It would have been further obvious to store only high-demand content in the cache. The motivation to combine stems from the desire to minimize latency for high-demand content. Additionally, this reduces unnecessary replication of unpopular content. POSITA would have had a reasonable expectation of success because popularity-based algorithms were routinely applied to distributed caching systems.

It would have been obvious to combine or modify each reference with another reference identified in Sections IV.A to render these claims invalid because each teaches complementary aspects of tiered content delivery. A person of ordinary skill in the art (“POSITA”) would have been motivated to combine these teachings to improve scalability and reduce latency—well-recognized design goals in CDN systems. Combining known tiered architectures with resource distribution techniques would have been a predictable use of prior art elements according to their established functions, yielding no unexpected results. Market demand for efficient content delivery and widespread adoption of hierarchical caching would have provided further motivation. POSITA would have had a reasonable expectation of success because these techniques were routinely implemented in distributed systems at the time.

Demand and Market Forces: For example, market forces suggested a demand for increased security and versatility of user authentication systems, and thus one of skill in the art would have thought to combine or modify references that described known methods that one of skill in the art would have recognized as offering improvements to solutions of that time.

Known Elements; Obvious to Try: Moreover, many of the claim elements were already known as admitted by the applicants in the patent specification. These elements represented design choices available to a person of ordinary skill. When there is a design need or market pressure to solve a problem such as identified previously and/or described in the '692 Patent, and there are a finite number of identified, predictable solutions, a person of ordinary skill would be motivated to combine the known options that are within his or her technical grasp. For the '692 Patent, there are a finite number of approaches for organizing content on a content delivery network. For example, as of the priority date of the '692 Patent, a POSITA would have been very familiar with basic elements of the '692 Patent's claims such as tiered caching. *See* Appx. E. Such elements were within the knowledge of a POSITA, whether or not disclosed in any particular reference, and the inclusion of such elements in the '692 Patent's claims does not render them novel or non-obvious.

Before the alleged inventions of the '692 Patent, "content delivery networks" or "CDNs" were widely used for delivering content. *See, e.g.,* Weller at 1 :16-18 ("It is well-known to deliver digital content (e.g., HTTP content, streaming media and applications) using an Internet content delivery network (ICDN)."); Raciborski at 0002 ("A content delivery network (CDN) is used by many web sites to deliver content more efficiency."); Fujimoto at 1: 11-15 ("[T]here is known a contents delivery system which includes a network player, which can reproduce contents data ... such as moving picture and an audio, and plural network servers, which are connected to the

network player through a network."); Kovacs at 1: 15 ("Distributed computer systems are well known in the prior art. One such distributed computer system is a 'content delivery network' or 'CDN'"); see also Dilley at 1: 15-17 ("It is well-known to deliver digital content (e.g., HTTP content, streaming media and applications) using an Internet content delivery network (CDN)."). Many other systems utilized computer networks with servers to manage content delivery even if not specifically called a "CDN." *See, e.g.*, Katz at 0002 ("As computers and computer networks become more prevalent in our society, media distributors more frequently employ computer network channels rather than traditional channels such as television or radio for the distribution of information."); Armstrong at 1 :24-28 ("Recent advances in digital signal processing techniques and, in particular, improvements in digital compression techniques, have led to a plethora of proposals for providing new digital services via existing telephone, coaxial cable and satellite data networks"); Allegrezza at 0004 ("Delivery of client selected video content directly to a subscriber at a time chosen by the subscriber via a cable TV or other distribution network is a fast growing segment of the entertainment industry."); Tang '753 at 0004 ("Web server clusters are the most popular configurations used to meet the growing traffic demands imposed by the internet.").

Content delivery networks consisted of servers, sometimes called "cache servers" or "proxy servers," that stored copies or "caches" of content and processed requests for delivery. *See, e.g.*, Dilley at Abstract ("[A] cache hierarchy is established in the CDN comprising a given edge server region and either (a) a single parent region, or (b) a subset of the edge server regions.

In response to a determination that a given object request cannot be serviced in the given edge region, instead of contacting the origin server, the request is provided to either the single parent region or to a given one of the subset of edge server regions for handling."); Li at Abstract ("A virtual active network architecture of proxy servers for providing streaming media data over

wide area networks includes forming a hierarchical structure of proxy servers for multiplexing and delivering the live streaming media, and dynamically reconfiguring the hierarchical structure based upon user population, user distribution, usage patterns and network conditions."); Weller at 1 :27-54 ("The content delivery infrastructure usually comprises a set of 'surrogate' origin servers that are located at strategic locations (e.g., Internet network access points, Internet Points of Presence, and the like) for delivering copies of content to requesting end users."); Cottrile at 8: 16-26 ("Each one of the servers 402 has a local cache 403 for quickly retrieving annotation data."); Karger at 2:53-57 ("One approach to solving this dilemma is to use a group of servers that function together as a cache. Responsibility for answering requests is divided among the group."); Pathan at Abstract ("A CDN replicates content from the origin server to cache servers, scattered over the globe, in order to deliver content to end-users in a reliable and timely manner from nearby optimal surrogates."). Content delivery networks also included servers from which the cache servers obtained original content, sometimes called "origin servers." See, e.g., Raciborski at 0002 ("A web site or origin server is linked to the CDN such that some or all content can be sourced from the CDN rather than the web site."); Dilley at 1: 15-43 ("The content delivery infrastructure usually comprises a set of 'surrogate' origin servers that are located at strategic locations (e.g., Internet network access points, Internet Points of Presence, and the like) for delivering content to requesting end users."); Kovacs at 4:9-35 ("Preferably, a tiered distribution hub 304 maintains a persistent connection to the origin server."); Pathan at 9 ("The server where the definitive version of a resource resides is called origin server, which is updated by the content provider"); Seed at 0062 ("The origin server sites are typically owned and/or maintained by the network provider's customers for storing and serving one or more objects."); Swildens '460 at Abstract ("Origin sites store all of a customer's available content files."); see also A.K. Pathan et al., A Taxonomy and

Survey of Content Delivery Networks 1 (2006); see also Cisco White Paper at 3. Further, CDNs directed requests for content to particular servers or groups of servers based on certain criteria to ensure expeditious delivery. See, e.g., Weller at 1:21-26 ("A request from a requesting end user for given content is directed to a 'best' replica, where 'best' usually means that the item is served to the client quickly compared to the time it would take to fetch it from the content provider origin server.").

One well-established framework for organizing the servers used a "multi-tier" structure, where CDNs would have multiple tiers of servers to facilitate delivery. See, e.g., Li at 0006 ("In one aspect the invention affords a method of distributing streaming data in a wide area network that has an overlay network of proxy servers that comprises activating the proxy servers to form a hierarchical structure comprising multiple tiers of proxy servers with respect to a data stream from a corresponding data source to distribute the data stream to a plurality of users."); Adrangi at 2:35-43 ("Elements of the present invention may be included within a multi-tiered networking architecture."); Cottrille at 2:2-6 ("The computing system consists of multiple tiers of servers. A tier I server indicates whether there are annotations associated with a content source. A tier II server indexes the annotations. A tier III server stores the body of the annotation."); Kovacs at 8:44-55 ("A multi-tier cache hierarchy solution may provide for additional origin offload, particularly in the event that a region in the tiered distribution map is taken out of service for any reason.").

In order to improve delivery efficiency and avoid traffic flow congestion, it was well known in the art to store or cache content among different servers based on its popularity. See, e.g., Weller at 4:60-5:5 ("CDN software 206, among other things, is used to create and manage a 'hot' object cache 212 for popular objects being served by the CDN."); Adrangi at Abstract.

("Popularity values are calculated for a plurality of files over a period of time. The popularity values are then used to determine which files should be cached at various remote sites."); Armstrong at 3 :44-49 ("The video assets considered popular, as determined by the frequency of subscriber requests, are stored on the primary storage partitions in all the neighborhoods. The remaining infrequently requested video assets are divided and then selectively distributed amongst the neighborhood head-ends on the secondary storage partitions."); Dilley at 4:50-65 ("CDN software 206 creates and manages a 'hot' object cache 212 for popular objects being served by the CDN."); Katz at 0040 ("The decision as to whether an item should be cached hinges primarily on an indication of how popular the item is, i.e. the number of times that the item has been requested."); Kovacs at 5: 11-31 ("As a result, the cache hierarchy protects the content providers from requests for large, popular objects causing the CDN to make more requests than the content provider can handle."); see also Weller at 4:60-5:5.

Further, those skilled in the art developed methods under which different servers stored distinct portions of the content provider's library. See, e.g., Slater at 0014 (describing an aspect of the invention which involves "providing the network with a demand director server, a first resource server having a first resource thereon, and a second resource server having a second, different resource thereon[.]"), 0016 ("the demand director is aware of where different resources are stored so as to be able to direct a request for resources to the appropriate server that actually has the requested resource stored on it."), 0131 ("Typically the data stored on the servers will be file based data that is readily partitioned and for which access patterns are known or is predictable, for example video or audio data."); Sue at 12: 19-24 ("Different types of content or assets may be stored in the network partition 320 or the user partition 322. The content stored in the different partitions may relate to the tier of the content.").

Finally, a well-known method of routing requests among various servers on a CDN was hashing, a technique that has been known for about eighty years. See, e.g., H. Stevens, Hans Peter Luhn and the Birth of the Hashing Algorithm, IEEE Spectrum (2018) ("Starting in the 1940s, Luhn devised machines and schemes for parsing information, most notably the now widely used hashing algorithm, which he suggested as a way to sort both numbers and text."); Karger at 3:27-29 ("[A]n alternative approach is for clients to use a standard hash function to determine from which server to request a document."); Kovacs at 7 :31-44 (describing a technique that "creates serial number spreading, preferably by hashing a cache key into a serial number and forwarding the request to an in-region peer (in other words, another CDN server) across a back-end interface of content servers within a region."); Wilson at 0107 ("With information stored in a number of hierarchical file structures, the management subsystem repository naturally lends itself to partitioning between directories, volumes and servers. These hierarchies may also be further partitioned according to the logical groupings within them (alphabetically A-K and L-Z for example), as required."); Swildens '460 at 004 7 ("The intelligent DNS server(s) and all other servers are able to compute a hash (this can also be a directory address) from a content file identifier to determine which origin server should store the file on its local storage device.").

c. Exemplary Combinations

For at least the reasons described above, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix E with any other reference or references listed in Appendix E along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '692 Patent.

For example, and without limitation, the Asserted Claims of the '692 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

(1) Seed in view of Swildens, Karger, and/or Dilley

A POSITA would have been motivated to combine Seed with Swildens, Karger, and/or Dilley because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

For instance, although Seed describes a CDN designed to alleviate server overload by storing popular content at the edge and distributing less-popular files across other servers, (*see generally* Exhibit E-03 [Seed]), Swildens and Karger each also teach storing distinct portions of file sets over clusters of servers using a hash function, (*see, e.g.*, Exhibits E-04-[Swildens], E-02 [Karger], cls. 1[d][ii]-[iii], 2, 5, 7, 15[c][i]-[ii], 15[d][i]-[ii]). A POSITA would have been motivated to store distinct portions of file sets over Seed's cluster of parent and/or origin servers using a hash function to, for instance, distribute request loads across more servers, reduce the memory required of each server to store the content provider's complete library, increase redundancy of distinct portions of the file set.

A POSITA would have further understood that the exact configuration of a network—including (1) the number and types of server tiers, and number of servers in each tier, (2) the arrangement and connections between servers within each tier and between different tiers, (3) how file sets are stored and/or distributed between tiers and servers in each tier, (4) how to locate and retrieve files for delivery to an end user, and (5) which specific network elements are configured to perform specific functions, each had well-known options that were known to be mere design choices based on factors such as cost, complexity, latency, and redundancy. Modifying Seed to adopt specific configuration details taught by Swildens, Karger, Dilley, and/or a POSITA's general understanding of the art would have merely amounted to a design choice, a simple substitution of

one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Seed suggests redirecting edge servers to parent servers when a popular file is missing from an edge server (Seed, [0017]-[0018], Seed also recognizes that fetching files from other servers to the edge was a well-known alternative to redirecting, (*id.*, [0013], [0046], [0059]; *see also* U.S. Pat. No. 6,301,617 (“the server uses the virtual URL system to locate the best current location for the resource and either redirects the requesting browser to it...or else fetches and returns the resource”), and a POSITA would have known that redirecting and fetching each provided their own distinct benefits.

(2) Swildens in view of Seed, Karger, and/or Dilley

A POSITA would have been motivated to combine Swildens with Seed, Karger, and/or Dilley because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

For instance, although Swildens describes a CDN designed to alleviate server overload by storing popular content at the edge and distributing less-popular files across a separate tier of origin servers, (*see generally* Exhibit E-04 [Swildens]), Seed expressly includes a tier of parent servers between the edge and origin servers that are configured to perform substantially the same functions as Swildens’ origin-tier servers (*see, e.g.*, Exhibit E-03 [Seed], cls. 1[d][i]-[iv], 15[c], 17, 18). Similarly, Dilley discloses including an intermediate parent tier of servers between edge servers and origin servers. *See, e.g.*, Exhibit E-01 [Dilley], cls. 1[d][i]-[iv], 15[c], 17, 18. A POSITA would have been motivated to implement a tier of parent servers in Swildens and configure those parent servers to perform the functions Swildens assigns to the origin servers, to further distribute request

loads across more servers (which may also be located in different geographic regions than the edge and origin servers to improve latency), and increase redundancy of distinct portions of the file set.

A POSITA would have further understood that the exact configuration of a network—including (1) the number and types of server tiers, and number of servers in each tier, (2) the arrangement and connections between servers within each tier and between different tiers, (3) how file sets are stored and/or distributed between tiers and servers in each tier, (4) how to locate and retrieve files for delivery to an end user, and (5) which specific network elements are configured to perform specific functions, each had well-known options that were known to be mere design choices based on factors such as cost, complexity, latency, and redundancy. Modifying Swildens to adopt specific configuration details taught by Seed, Karger, Dilley, and/or a POSITA’s general understanding of the art would have merely amounted to a design choice, a simple substitution of one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Swildens suggests determining the popularity of a file directly in response to determining the file is missing from an edge server, if the claims expressly require this configuration and Swildens cannot be read to teach it, Seed does, (*see, e.g.*, Exhibit E-03 [Seed], cl. 1[b]), and a POSITA would have known that determining popularity directly in response to finding a missed file would provide its own benefits, such as expediting popularity determinations. As another example, although Swildens suggests redirecting edge servers to origin servers when a popular file is missing from an edge server (Swildens, [0068]-[0069]), other prior art such as Seed recognized that fetching files from other servers to the edge was a well-known alternative to redirecting, (Seed, [0013], [0046], [0059]; *see also* U.S. Pat. No. 6,301,617 (“the server uses the virtual URL system to locate the best current location for the resource and either

redirects the requesting browser to it...or else fetches and returns the resource”), and a POSITA would have known that redirecting and fetching each provided their own distinct benefits. As one final example, although Swildens discloses using a hash function to distribute distinct sets of files across a tier of origin servers, Karger similarly provides an alternative hashing algorithm to similarly distribute distinct sets of files across a tier of servers, (*see, e.g.*, Karger, 9:22-25, 12:29-63, 18:14-24, 96:61-66), and a POSITA would have been motivated to replace one with another as a simple substitution or application of a known-technique, among other distinct benefits taught by Karger.

(3) Dilley in view of Seed, Swildens, and/or Karger

A POSITA would have been motivated to combine Dilley with Seed, Swildens, and/or Karger because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

For instance, although Dilley describes a CDN designed to alleviate server overload by storing “hot” (or popular) content at the edge and distributing less-popular files across other servers, (*see generally* Exhibit E-01 [Dilley]), Swildens and Karger each also teach storing distinct portions of file sets over clusters of servers using a hash function, (*see, e.g.*, Exhibits E-04 [Swildens], E-02 [Karger], cls. 1[d][ii]-[iii], 2, 5, 7, 15[c][i]-[ii], 15[d][i]-[ii]). A POSITA would have been motivated to store distinct portions of file sets over Dilley’s cluster of parent and/or origin servers using a hash function to, for instance, distribute request loads across more servers, reduce the memory required of each server to store the content provider’s complete library, increase redundancy of distinct portions of the file set.

A POSITA would have further understood that the exact configuration of a network—including (1) the number and types of server tiers, and number of servers in each tier, (2) the

arrangement and connections between servers within each tier and between different tiers, (3) how file sets are stored and/or distributed between tiers and servers in each tier, (4) how to locate and retrieve files for delivery to an end user, and (5) which specific network elements are configured to perform specific functions, each had well-known options that were known to be mere design choices based on factors such as cost, complexity, latency, and redundancy. Modifying Swildens to adopt specific configuration details taught by Seed, Karger, Dilley, and/or a POSITA's general understanding of the art would have merely amount to a design choice, a simple substitution of one known element for another to achieve a predictable result, or applying a known technique to a known device to achieve a predictable result.

As one example, although Dilley describes storing “hot” objects at the edge, (*see generally* Exhibit E-01 [Dilley]), Seed and Swildens provide additional known implementation details for this concept, such as storing only the most popular files at the edge, and evaluating whether a requested file is popular (like Dilley's “hot”) when an edge server is missing the requested file. *See, e.g.*, Exhibits E-03 [Seed], E-04 [Swildens], cls. 1[b]-[d][i], 8, 15[b], 15[d], 17, 18). A POSITA would have been motivated to combine Dilley with Seed or Swildens as merely design choices as to implementation details, while ensuring the most popular files are stored closest to end-users as the network edge, where latency is critical, and not popular files deeper in the network, fetching not popular files from the edge to the servers storing those files deeper in the network as needed.

(4) Akamai EdgeSuite in view of Swildens and Karger

It would have been obvious to combine Akamai EdgeSuite with Swildens and/or Karger because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features. Akamai EdgeSuite teaches a

“Comprehensive Content Delivery Solution” with multiple tiers. Akamai EdgeSuite Enterprise Edition Brochure at 1; Turbo-Charging Dynamic Web Sites with Akamai EdgeSuiteSM White Paper. Akamai EdgeSuite also made use of hashing to improve its network. David Karger, Consistent Hashing and Random Trees: Distributed Caching Protocols for Relieving Hot Spots on the World Wide Web. A POSITA would be motivated to incorporate other uses of hashing such as those described in Swildens and Karger to distribute request loads across more servers, reduce the memory required of each server to store the content provider’s complete library, increase redundancy of distinct portions of the file set.

(5) Akamai EdgeSuite in view of Seed

It would have been obvious to combine Akamai EdgeSuite with Seed because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features. Akamai EdgeSuite used selective caching to optimize the network and handle dynamic content. EdgeSuite 5.0: ESI Developer’s Guide at 11. It would have been obvious to investigate additional selective caching techniques in furtherance of “efficient distribution of any type of content.” Akamai EdgeSuite Enterprise Edition Brochure at 3. A POSITA would have been motivated to combine Akamai EdgeSuite with Seed because Seed teaches popularity-based caching and tiered architecture (see Exhibit E-03 [Seed], cls. 1[b]-[d]). Adding Seed’s popularity evaluation to Akamai EdgeSuite would allow dynamic placement of hot content at the edge, reducing latency and improving efficiency. This is applying a known technique to a known system for predictable results.

(6) Akamai EdgeSuite in view of Dilley

A POSITA would have been motivated to combine Akamai EdgeSuite with Dilley because Dilley teaches multi-tier architectures and storing hot content at the edge (see Exhibit E-01

[Dilley]). Incorporating Dilley's tiering into Akamai EdgeSuite would further distribute load and improve redundancy. This would have been a mere design choice among well-known options.

(7) Akamai EdgeSuite in view of Swildens, Dilley, Kovacs

A POSITA would have further been motivated to combine Akamai EdgeSuite with other Akamai patents and publications such as Swildens, Dilley, and Kovacs. A POSITA would be motivated to review other teachings from Akamai for other potential functionality in the system. For example, a POSITA would be motivated to use the "map rule" because it is "useful for reducing traffic to the origin server or for avoiding duplication of content that is large and infrequently requested." Kovacs at 13:50-58.

(8) Squid in view of Swildens and Karger

It would have been obvious to combine Squid with Swildens and Karger because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features. Squid used CARP to partition material requests to parents. Squid: The Definitive Guide at 262. A POSITA would have been motivated to combine Squid with Swildens because Squid's proxy caching could benefit from Swildens' hashing-based distribution because it is less CPU-intensive. Squid: The Definitive Guide at 262. This would also reduce duplication and improve scalability, a predictable result of combining known techniques.

(9) Squid in view of Seed

It would have been obvious to combine Squid and Seed because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features. Squid used various caching policies such as "heap GDSF

policy optimizes object hit rate by keeping smaller popular objects in cache so it has a better chance of getting a hit.” https://web.archive.org/web/20080511215445/http://www.squid-cache.org/Versions/v3/3.0/cfgman/cache_replacement_policy.html. A POSITA would have been motivated to combine Squid with Seed because Seed’s popularity-based caching could enhance Squid’s proxy caching efficiency by prioritizing hot content at the edge. This would have been a routine improvement.

(10) Squid in view of Dilley

A POSITA would have been motivated to combine Squid with Dilley because Dilley’s tiered architecture could be applied to Squid to improve load distribution and redundancy. This is a known design choice.

(11) Windows Media Services in view of Swildens and Karger

It would have been obvious to combine Windows Media Services and Swildens or Karger because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features. Windows Media Services was directed at “conserving network bandwidth, decreasing network-imposed latency, and offsetting the load on the origin server.”

<https://web.archive.org/web/20070811164210/http://www.microsoft.com/windows/windowsmedia/forpros/serve/features.aspx>. A POSITA would have been motivated to combine Windows Media Services with Swildens and/or Karger because Swildens teaches distributing distinct file sets across servers using hashing, and Karger teaches improved hashing algorithms for dynamic scaling. Applying these to Windows Media Services would allow content-aware distribution, reducing storage overhead and improving fault tolerance. This combination would involve routine engineering and yield predictable results.

(12) Windows Media Services in view of Seed

A POSITA would have been motivated to combine Windows Media Services with Seed because Seed teaches popularity-based caching. Windows Media Services include the capability to control “cache and proxy policies.” Incorporating Seed’s popularity evaluation into Windows Media Services could optimize which media files are replicated or cached at edge nodes, reducing latency for high-demand streams. This is applying a known technique to a known system for predictable benefits.

(13) Windows Media Services in view of Dilley

A POSITA would have been motivated to combine Windows Media Services with Dilley because Dilley teaches tiered architectures for distributing content. Applying this to Windows Media Services would allow intermediate tiers for media distribution, improving scalability and reducing load on origin servers. This is a well-known design choice.

(14) Windows Media Services in view of Roth

A POSITA would have been motivated to combine Windows Media Services with Roth because Roth teaches related to caching control. A POSITA would have further been motivated to combine Windows Media Services with other Microsoft patents and publications such as Roth. A POSITA would be motivated to review other teachings from Microsoft for other potential functionality in the system.

(15) Windows Media Services in view of Akamai EdgeSuite and Squid

A POSITA would have been motivated to combine Windows Media Services with Akamai EdgeSuite because Akamai EdgeSuite’s CDN techniques for edge caching and request routing could complement Windows Media Services’ streaming capabilities, reducing latency and improving global delivery. This is a predictable result of combining known systems. Similarly, a

POSITA would have been motivated to combine Windows Media Services with Squid because Squid's proxy caching could reduce repeated requests for the same media streams, complementing Windows Media Services' streaming architecture. This is a routine improvement.

6. U.S. Patent No. 10,701,173

Microsoft identifies in Appendix F the prior art references that render obvious the Asserted Claims of the '173 Patent and include below exemplary combinations showing the obviousness of the '173 Patent Asserted Claims in view of the prior art. To the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix F, the limitation would have been obvious in light of the disclosures within the reference and the knowledge of one of ordinary skill in the art at the time of the '173 Patent. Moreover, to the extent Plaintiff contends that an element is not disclosed in any one of the anticipatory references described in Appendix F, such reference may be combined with any other references listed in Appendix F for such element, thereby rendering the claims invalid for obviousness. To the extent a finder of fact determines that a limitation of any of the '173 Patent Asserted Claims is not disclosed by one of the references identified above, the claim is nevertheless unpatentable as obvious because they contain nothing that constitutes a patentable innovation. To the extent a finder of fact determines that a limitation of the '173 Patent Asserted Claims is not anticipated, it does not go beyond combining familiar elements according to known methods to achieve predictable results or does more than choose between clear alternatives known to those of ordinary skill in the art.

a. Obviousness

For at least the reasons described below, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix F with any other reference or references listed in Appendix F along with the knowledge of one of ordinary skill in the art to

arrive at the inventions claims in the '173 Patent. For example, and without limitation, the Asserted Claims of the '173 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
O'Rourke	• Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art.
Eriksen	• O'Rourke • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art.
Ben-Shaul	• O'Rourke • Eriksen • Ortiz Rodriguez • Blumofe • Middleton

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Ortiz Rodriguez	• O'Rourke • Eriksen • Ben-Shaul • Blumofe • Middleton • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Blumofe	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Middleton • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Middleton	• O'Rourke

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Plamondon	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
HTTP/1.1	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Cohen	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • HTTP/1.1 • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Touboul	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • HTTP/1.1 • Cohen • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Xie	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • HTTP/1.1

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Cohen • Touboul • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Server 2008 including Windows Media Services 9	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Squid • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Akamai EdgeSuite • The knowledge of a person of ordinary skill in the art
Squid	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez

Patent No. or Title (Primary Inventor/Author)	In Combination with One or More of: (Patent No. or Title (Primary Inventor/Author))
	• Blumofe • Middleton • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Akamai EdgeSuite • Roth • The knowledge of a person of ordinary skill in the art
Akamai EdgeSuite	• O'Rourke • Eriksen • Ben-Shaul • Ortiz Rodriguez • Blumofe • Middleton • Plamondon • HTTP/1.1 • Cohen • Touboul • Xie • Server 2008 including Windows Media Services 9 • Squid • Roth • The knowledge of a person of ordinary skill in the art

b. General Motivations to Combine

Obviousness: For at least the reasons described in these contentions, it would have been obvious to one of ordinary skill in the art to combine any of a number of prior art references, including any combination of those prior art references identified in Appendix F along with the knowledge of one of ordinary skill in the art to meet the limitations of the '173 Patent Asserted Claims. As a result, the references identified herein render one or more Asserted Claims of the '173 Patent obvious when the references are read in combination with each other, and/or when read in view of the state of the art and knowledge of those skilled in the art. Each and every

reference identified is also relevant to the state of the art at the time of the alleged invention. Any of the references disclosed above may be combined with one another to render obvious (and therefore invalid) each of Sandpiper's Asserted Claims. Defendant may rely upon a subset of the identified references or all of the references identified herein, including all references in Appendix F, for purposes of obviousness depending on the Court's claim construction, positions taken by Sandpiper during this litigation, and further investigation and discovery.

Reservation of Rights: Moreover, as mentioned above, Defendant has not yet completed their search or discovery concerning additional prior art. As such, Defendant's inclusion of exemplary combinations does not preclude them from identifying other invalidating combinations as appropriate, and Defendant reserves the right to identify additional specific combinations as well as to detail and explain such combinations.

No More Than Ordinary Variations of Prior Art; No Unexpected Results: To the extent not anticipated, the '173 Patent Asserted Claims represent no more than the result of ordinary variations of the prior art. Defendant further believes that no showing of a specific motivation to combine prior art is required to combine the references disclosed above and in the attached charts, as each combination of art would have no unexpected results, and at most would simply represent a known alternative to one of ordinary skill in the art. *See KSR Int'l Co. v. Teleflex, Inc.*, 550 U.S. 398, 415-16 (2007) (rejecting the Federal Circuit's "rigid" application of the teaching, suggestion, or motivation to combine test, instead espousing an "expansive and flexible" approach); *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1058 (Fed. Cir. 2016); *Intercontinental Great Brands LLC v. Kellogg N. Am. Co.*, 869 F.3d 1336, 1344 (Fed. Cir. 2017). Indeed, the Supreme Court held that a person of ordinary skill in the art is "a person of ordinary creativity, not an automaton" and "in many cases a person of ordinary skill in the art will be able

to fit the teachings of multiple patents together like pieces of a puzzle.” *Id.* at 420-21. Nevertheless, in addition to the information contained elsewhere in these contentions, Defendant identifies motivation and reason to combine the cited art.

Predictable Results; Obvious to Try: One or more combinations of the prior art references identified in Appendix F would have been obvious because these references would have been combined using: known methods to yield predictable results; known techniques in the same way; a simple substitution of one known, equivalent element for another to obtain predictable results; and/or a teaching, suggestion, or motivation in the prior art generally. In addition, it would have been obvious to try combining the prior art references identified above because there were only a finite number of predictable solutions and/or because known work in one field of endeavor prompted variations based on predictable design incentives and/or market forces either in the same field or a different one. For example, a POSITA would have been motivated to use an HTTP request because HTTP was the dominant protocol for web content delivery and this provides interoperability with existing web infrastructure. Further, the combinations of the prior art references identified in Appendix F would have been obvious because the combinations represent known potential options with a reasonable expectation of success.

Interrelated Teachings: Additional evidence that there would have been a motivation to combine the prior art references identified above includes the interrelated teachings of multiple prior art references; common authorship; the effects of demands known to the design community or present in the marketplace; the existence of a known problem for which there was an obvious solution encompassed by the ‘173 Patent Asserted Claims; the existence of a known need or problem in the field of the endeavor at the time of the alleged invention(s); and the background knowledge that would have been possessed by a person having ordinary skill in the art.

Same Area of Technology: Additionally, it would be obvious to one of ordinary skill in the art to consult and/or combine any of the prior art listed in Appendix F because all of these references relate to the same area of technology and/or are from analogous art. The ‘173 Patent Asserted Claims are directed to a “content delivery network.” ‘173 Patent at Abstract. The ‘173 Patent Asserted Claims merely unite old elements, well known in the field, with no change in their respective function or result. Given the interrelated teachings of the prior art, the effects of demands known to the design community or present in the marketplace, and the background knowledge possessed by a person having ordinary skill in the art, it would have been obvious for one of ordinary skill in the art to combine these familiar elements, disclosed and/or embodied in the prior art listed above to practice the ‘173 Patent Asserted Claims.

Same Technical Issues: A person of ordinary skill would have been motivated to combine the above prior art based on his/her knowledge, the nature of the problem to be solved, and the teachings of the prior art. The identified prior art addresses the same or similar technical issues and suggests the same or similar solutions to those issues. Moreover, some of the prior art refer to or discuss other prior art, illustrating the close technical relationship among the prior art. For the ‘173 Patent, one of ordinary skill in the art would have been motivated to combine known prior art solutions described in these references relating to caching and cache policies. Furthermore, a POSITA would have been motivated to use cache policies that relate to content outside of the CDN. This would have been obvious to enforce uniform caching rules across multiple domains. A POSITA would have been motivated to do so to improve consistency and compliance. Similarly a POSITA would have been motivated to use a database for cache policies for to simplify policy updates, maintain consistency across nodes, and provide centralized control.

Demand and Market Forces: For example, market forces suggested a demand for increased security and versatility of user authentication systems, and thus one of skill in the art would have thought to combine or modify references that described known methods that one of skill in the art would have recognized as offering improvements to solutions of that time.

Known Elements; Obvious to Try: Moreover, many of the claim elements were already known as admitted by the applicants in the patent specification. These elements represented design choices available to a person of ordinary skill. When there is a design need or market pressure to solve a problem such as identified previously and/or described in the '173 Patent, and there are a finite number of identified, predictable solutions, a person of ordinary skill would be motivated to combine the known options that are within his or her technical grasp. For the '173 Patent, there are a finite number of approaches for expanding the capabilities of proximity systems and location tracking systems to run or interact with different applications. For example, as of the priority date of the '173 Patent, a POSITA would have been very familiar with basic elements of the '173 Patent's claims such cache policies. *See* Appx. F. Such elements were within the knowledge of a POSITA, whether or not disclosed in any particular reference, and the inclusion of such elements in the '173 Patent's claims does not render them novel or non-obvious.

Summary of Motivations to Combine: Thus, the motivation to combine the teachings of the prior art references disclosed in Appendix F is found in the references themselves and also in: (1) the nature of the problem being solved; (2) the express, implied and inherent teachings of the prior art; (3) the knowledge of persons of ordinary skill in the art; (4) the predictable results obtained in combining the different elements of the prior art; (5) the predictable results obtained in simple substitution of one known element for another; (6) the use of a known technique to improve similar devices, methods, or products in the same way; (7) the predictable results obtained

in applying a known technique to a known device, method, or product ready for improvement; (8) the finite number of identified predictable solutions that had a reasonable expectation of success; and (9) known work in various technological fields that could be applied to the same or different technological fields based on design incentives or other market forces.

c. Exemplary Combinations

For at least the reasons described above, it would have been obvious to one of ordinary skill in the art to combine each prior art reference listed in Appendix F with any other reference or references listed in Appendix F along with the knowledge of one of ordinary skill in the art to arrive at the inventions claims in the '173 Patent.

For example, and without limitation, the Asserted Claims of the '173 Patent would have been obvious to one of ordinary skill in the art in view of the following combinations:

(1) O'Rourke in view of Eriksen

It would have been obvious to combine O'Rourke and Eriksen because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

O'Rourke's disclosure is directed to a content delivery network (CDN) having an origin server coupled to a cache server which distributes content to a client. *See*, O'Rourke, 1:41-2:7. A In existing CDNs, the cache server comprises "fixed architecture that is difficult to customize...[and] perform various cache-related functions that are controlled by a set of cache policies that are determined by the manufacturer." *Id.*, 2:8-27. O'Rourke's disclosure is directed to solving the problem by "providing a single cache server that can operate as either an origin server or a cache server." *Id.*, 2:28-29. The CDN comprises a cache having a "flexible architecture

that separates the various cache functions from the policies that determine which functions are to be performed in a particular situation.” *Id.*, 2:29-32. This allows a “common cache server to be configured in a variety of different ways depending on the set of policies applied to the cache server.” *Id.*, 2:32-35. The benefit of this architecture allows for “different manufacturers or end users can create different cache server functionality based on the set of policies selected.” *Id.*, 2:35-37. O’Rourke discloses a number of policy determinations made after the determination of the content in the server, including calculating the time-to-live (TTL). *Id.*, 6:31-38 (“the requested content is available, then the cache policies determine whether the cached version of the requested content is an acceptable version (block 510). An acceptable version is determined based on the cache policies (e.g., whether the cached version is the same as the origin server’s version or is within thirty minutes of the origin server’s version).”), *see also* Fig. 5. O’Rourke further discloses where the cache server is a Windows Media Services and that “the set of cache policies are contained in one or more cache plug-in modules that are loaded in the Windows Media Services and called by the Windows Media Services.” *Id.*, 4:57-67.

Eriksen is a disclosure similarly directed to improving the speed and reliability of caching content. Eriksen, ¶5. Eriksen discloses a similar method of determining whether to serve content stored in a cache by first determining if content is in the cache (303) and then determining the policy of the content (305/306), where the policies include freshness, TTL, or parameters of the content. *See id.*, ¶39-43, Fig. 3. Eriksen further discloses where “some of the content freshness parameters are derived from the HTTP header of the document content 205” using the HTTP.1,1 standard. *Id.*, ¶35, *see also id.*, ¶¶41-42. Specifically, Eriksen discloses where determining the current cache policy comprises determining a time period for expiration of content in the cache (i.e., a freshness check), and where the freshness check is implemented using the HTTP/1.1 Cache-

Control protocol. *Id.*, ¶42 (“this freshness check examines the HTTP/1.1 Expires header value, which gives the date/time after which the HTTP response is considered stale.”). Eriksen further discloses where the HTTP headers are used to determine the freshness of the cached object. *Id.*, ¶35 (“the client cache 108 (in client 102) includes a table 201 having a plurality of URL fingerprints. In some embodiments, a URL fingerprint is a 64-bit binary number (or a value of some other predetermined bit length) generated from the corresponding URL by first normalizing the URL text, e.g., by applying a predefined set of normalization rules to the URL text (e.g., converting web host names to lower case), and then applying a hash function to the normalized URL text to produce the 64-bit URL fingerprint. These URL fingerprints correspond to the documents stored in the client cache 108. Each entry in the URL fingerprint table 201 points to a unique entry in another table 203 that stores the content information of a plurality of documents. Each entry in the table 203 includes a unique content fingerprint (also known as content checksum), one or more content freshness parameters and a pointer to a copy of the corresponding document content 205. In one embodiment, some of the content freshness parameters are derived from the HTTP header of the document content 205. For example, the Date field in the HTTP header indicates when the document was downloaded to the client 102.”) (emphasis added), *see also*, Fig. 2.

A POSITA would have been motivated to combine O’Rourke, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-01 [O’Rourke]) with Eriksen, which describes managing content freshness in a comparable CDN using HTTP/1.1 caching standards to retrieve the content and to calculate content expiration (*see, e.g.*, Exhibit F-02 [Eriksen], cls. 1[b1]-[b2], 3, 6). A POSITA would have recognized that O’Rourke discloses storing and delivering content through a CDN and that Eriksen discloses a CDN that operates in a

comparable manner where the specifics of calculating the expiration of the content based on a caching policy, and the use of the HTTP/1.1 caching standard for determining the content's policy and expiration. A POSITA would have been motivated to use Eriksen's HTTP headers to determine the cache policies of the content because HTTP was a well-known standard at the time of application, thereby reinforcing that caching policies applied to content freshness, i.e., where determining a current cache policy comprises determining a time period for expiration of content in the cache; and where HTTP requests were used to request content comprising one or more resources were known at the time of application. This combination of O'Rourke and Eriksen would involve only routine engineering leading to predictable results. Additionally, a POSITA would have found it obvious to use HTTP/1.1, as disclosed by Eriksen, as the communication protocol in a CDN because HTTP/1.1 is a well-known, industry-standard protocol for enabling interoperability between networked devices and web-based applications. The widespread adoption and documentation of HTTP/1.1 for similar enterprise systems would have motivated its use in Windows Media Services implementations without requiring inventive skill.

(2) O'Rourke in view of Ben-Shaul

It would have been obvious to combine O'Rourke and Ben-Shaul because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen, O'Rourke teaches a CDN having a flexible cache server that serves content based on caching policies to manage content freshness and ensure that only fresh content is served to a client.

Ben-Shaul describes a decentralized and differentiated content delivery architecture that empowers content providers to directly control how their content is cached and delivered across distributed edge servers. Ben-Shaul, ¶48. Unlike traditional content delivery networks (CDNs), where caching policies are centrally managed and often coarse-grained, Ben-Shaul’s disclosure enables each content provider to define fine-grained caching policies tailored to specific content, regions, times, and user groups. *Id.*, ¶¶48-55. These policies can specify cache priorities (e.g., normal, high, sticky/mirrored, or no-cache), active update schedules, prefetching times, and cache invalidation rules. *Id.*, ¶¶218-237. The command server (also referred to as the policy control server or content delivery manager), acts as the authoritative source for service policies and directives. *Id.*, ¶72. Content providers use graphical tools to author these policies, typically in an XML-based markup language (CDML), and store them on the command server. *Id.*, ¶¶182-192. Edge servers periodically poll the command server to download updated directives, ensuring that their caching behavior and other service attributes remain synchronized with the provider’s intent. *Id.*, ¶69. This push/pull mechanism allows for dynamic, real-time adjustment of edge server behavior, including cache management, resource transformation, and application delivery. *Id.*, ¶¶186-193. That is to say, Ben-Shaul discloses where the policy is determined at a database distinct from the node. *Id.*, ¶182 (“The content delivery manager 44 is associated with an origin web server 46, and optionally with a policy control server 42, although it does not have to reside in the same physical site as the origin web server 46 or the policy control server 42. The content delivery manager 44 is responsible for specifying content delivery policies, preferably using CDML, and for embedding CDML in the content, either within actual web pages or at separately addressable CDML pages, as well as deploying it to edge servers.”), *see also*, Fig. 6. Ben-Shaul discloses where this enables “[e]ach edge server [to] implement[s] the policy that is specified by

the content provider and other sources of control. Thus, the content provider may control the operation of its edge servers, although a specific policy can delegate some or all the control to the edge servers, to a local policy derived at the target site, or in other cases to a different (third party) source of control.” *Id.*, ¶37.

A POSITA would have been motivated to combine O’Rourke, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-01 [O’Rourke]) with Ben-Shaul, which describes the components of a CDN and where a database (external to a node in the CDN) stores the caching policies and accessed by the nodes to apply policies to content (*see, e.g.*, Exhibit F-03 [Ben-Shaul], cls. 1[b1]-[b2], 4-5). A POSITA would have recognized that O’Rourke discloses storing and delivering content through a CDN and that Ben-Shaul discloses a CDN that operates in a comparable manner, where both disclosures are directed to improving the flexibility and modification of cache server polies. A POSITA would have been motivated to combine Ben-Shaul with O’Rourke to achieve the same improvement in edge server behavior. A POSITA would have found it further obvious to store policy information on a remote server because centralized storage facilitates easier management, updating, and access control across distributed client devices. The use of remote servers for storing and managing shared data is well-established in networked computing environments, making its application to policy information a routine design choice rather than an inventive step. This combination of O’Rourke and Ben-Shaul would involve only routine engineering leading to predictable results.

(3) O’Rourke in view of Ortiz Rodriguez

It would have been obvious to combine O’Rourke and Ortiz Rodriguez because the combination does nothing more than combine prior art elements from similar systems according

to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen, O'Rourke teaches a CDN having a flexible cache server that serves content based on caching policies to manage content freshness and ensure that only fresh content is served to a client.

The disclosure of Ortiz Rodriguez is directed to an improved caching framework that "allows the client application to store data and then efficiently query for that data." Ortiz Rodriguez, ¶9. This caching framework "determines when the data in the cache is stale and should not be used by the application" in offline use of the service. *Id.* Ortiz Rodriguez discloses a comparable process for checking the expiration of the content stored in the cache. *See id.*, ¶56 ("Cache service 501 sends the call IndexedDBCACHE.getEntites() 601 to IndexedDBCACHE 503, which then sends the call getExpirationDate 602 to IndexDB 504. IndexDB 504 returns the expiration date/time in message 603. IndexedDBCACHE 503 then compares the expiration date/time to its expiration policy. If the expiration date/time has exceeded the expiration policy, then IndexedDBCACHE 503 returns empty call 604 to cache service 501."). Ortiz Rodriguez discloses that the use of the service agent "ensures that the user interface receives the requested data without having to know if the data is coming from a local cache or a remote system. The service agent 203 knows that it will receive the requested data from cache service instance 210. It is the responsibility of cache service 210 to get the data from storage 211, if it is available, or from system 208, 209, if the data is not available locally." *Id.*, ¶34, *see also*, Fig. 2.

A POSITA would have been motivated to combine O'Rourke, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-01 [O'Rourke]) with Ortiz

Rodriguez, which describes the interface between the cache and calculating the expiration time of content within the cache before serving content to the client (*see, e.g.*, Exhibit F-04 [Ortiz Rodriguez], cls. 1[b1]-[b2]). A POSITA would have recognized that O'Rourke discloses storing and delivering content through a CDN and Ortiz Rodriguez discloses calculating content freshness in a comparable manner by comparing the life of stored content to a cache policy. A POSITA would have been motivated to use Ortiz Rodriguez's expiration calculation in the CDN of O'Rourke for the purpose of ensuring that only fresh content is delivered to the client. A POSITA would have found it obvious to employ a service agent to collect policy information and calculate expiration of content, as service agents are commonly used to automate data retrieval and processing tasks in distributed systems. The delegation of expiration computation to such agents is a routine design choice that streamlines policy enforcement and resource management, reflecting established practices in network and systems administration, for example, as disclosed by Ortiz Rodriguez. This combination of O'Rourke and Ortiz Rodriguez would involve only routine engineering leading to predictable results.

(4) O'Rourke in view of Blumofe

It would have been obvious to combine O'Rourke and Blumofe because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen, O'Rourke teaches a CDN having a flexible cache server that serves content based on caching policies to manage content freshness and ensure that only fresh content is served to a client.

Blumofe teaches “extend[ing] [a CDN] into one or more mobile or wireline networks that cannot or do not otherwise support fully-managed CDN edge servers.” Blumofe, Abstract. Blumofe recognized that “[b]ecause of the way that fixed or wireless mobile networks are architected and operated, it is not always possible or practical to deploy fully-managed CDN servers therein (i.e., close to end-users).” *Id.*, ¶7. For example, Blumofe observed that in 3G wireless mobile networks, the gateways used to connect wireless end-users to the internet, at which CDN servers could be deployed, may be thousands of miles and hundreds of milliseconds away from the end-users. *Id.*

To address this issue and provide caching of CDN-provisioned content closer to end-users, Blumofe describes a CDN Extender. *Id.*, ¶25. The Extender may be implemented in network environments that do not support CDN services (such as a wireless mobile network), and operates as a cache beyond the edge of the CDN. *Id.*, ¶¶25, 27; *see also id.* at Figure 3 The “Extender” is deployed on the network, for example, as a “as a passive web caching proxy that is beyond the edge of the CDN but that serves CDN-provisioned content under the control of the CDN.” *Id.*, ¶9. The CDN controls the Extender and caching policies are supplied by the CDN to the Extender, i.e., caching policies within the CDN relate to caching content outside of the CDN. *Id.*, ¶36 (“As noted above, the Extender caches and serves the CDN customer content based on information it receives, in terms of content and cache-control instructions, from the CDN edge servers.”). This enables serving of “CDN-provisioned content under the control of the CDN” in situations where it is not “possible or practical to deploy fully-managed CDN servers.” *Id.*, ¶¶7-9. That is to say, Blumofe discloses where the cache policy relates to caching content outside the CDN (“the Extender caches and serves the CDN customer content based on information it receives, in terms of content and cache-control instructions, from the CDN edge servers”). *Id.*, ¶36.

A POSITA would have been motivated to combine O’Rourke, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-01 [O’Rourke]) with Blumofe, which describes a CDN where the caching policies are related to caching content within the CDN and also related to caching content outside of the CDN (e.g., within the Extender) (*see, e.g.*, Exhibit F-05 [Blumofe], cls. 6-7). A POSITA would have recognized that O’Rourke discloses storing and delivering content through a CDN and Blumofe discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Blumofe’s Extender and caching policy relating to storage of content outside of the CDN in order to achieve the taught benefit of serving CDN-provisioned content in locations where doing so would otherwise be unavailable. A POSITA would have further found it obvious to deploy the Extender of Blumofe for applying CDN policies outside the traditional CDN infrastructure, as this enables consistent policy enforcement and content optimization. The extension of CDN functionalities via intermediary or edge servers is a well-known approach in network architecture, motivated by the need for uniform policy application and improved performance in distributed systems. This combination of O’Rourke and Blumofe would involve only routine engineering leading to predictable results. A POSITA would understand that in such a combined system, the Extender operates in a similar manner as the cache servers in order to deliver CDN content to users. *See, e.g., id.* at ¶9 (“[t]he CDN is extended into one or more mobile or wireline networks that cannot or do not otherwise support fully-managed CDN edge servers.”); *id.* (the Extender “is deployed in the mobile or wireline network, preferably as a passive web caching proxy that is beyond the edge of the CDN but that serves CDN-provisioned content.”); *id.* at ¶29 (the Extender “preferably delivers all forms of CDN content (i.e., content that has been off-loaded by content providers for delivery by the CDNSP over the CDN).”

Moreover, a POSITA would understand that in such a combined system, the CDN sends the cache control information to the Extender so that the Extender performs according to the CDN customer's rules, and the cache policy that configures the cache server within the CDN (e.g., with respect to whether cached content has expired or is an acceptable version) would also be used to configure the Extender outside the CDN. *Id.* at ¶¶ 30, 36, 37, 54.

(5) O'Rourke and Middleton

It would have been obvious to combine O'Rourke and Middleton because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen, O'Rourke teaches a CDN having a flexible cache server that serves content based on caching policies to manage content freshness and ensure that only fresh content is served to a client.

Middleton is directed to solving the problem of determining whether content in the cache that appears to be stale is still fresh/current. Middleton, ¶34. Middleton further discloses an extensible programming framework 1000 (e.g., an event-based extensible programming framework) to be incorporated within a cache server. The framework 1000 comprises a core engine 1002, configuration files 1004, and modules 1006, 1008, and 1010. Middleton, ¶105. The modules 1006, 1008, and 1010 "provide for authentication services for different customers." *Id.*, ¶110. For example, the modules may be invoked for different users and/or requests at different times depending upon a specific combination of configuration files 1004; rules, metadata, or other information resident in the core module 1002. *Id.*, ¶110. The modules also "provide an extensible

platform to write custom logic for features and services within a content delivery network. In one particular embodiment, scripts within modules 1006, 1008, and 1010 are associated with particular content available via a content delivery network. The scripts are then executed every time a user attempts to access that particular content.” *Id.*, ¶109. Thus, Middleton discloses where the cache may be customized, through the use of the modules, to different users and/or requests.

A POSITA would have been motivated to combine O’Rourke, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-01 [O’Rourke]) with Middleton, which describes a comparable CDN for content freshness management (*see, e.g.*, Exhibit F-06 [Middleton], cls. 1-7). A POSITA would have recognized that O’Rourke discloses storing and delivering content through a CDN and that Middleton discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to combine Middleton and O’Rourke to realize the benefits of Middleton’s revalidation which prevents unnecessary downloads from the origin server when the content is expired based on a caching policy but still unchanged from the content in the origin server, thereby reducing network traffic and saving on computational power. Likewise, a POSITA would have recognized the benefit of O’Rourke in that flexible cache architecture allows for a single cache to be configured in a variety of different ways depending on the set of policies applied to the cache server. This combination of O’Rourke and Middleton would involve only routine engineering leading to predictable results.

(6) O’Rourke and Plamondon

It would have been obvious to combine O’Rourke and Plamondon because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen, O'Rourke teaches a CDN having a flexible cache server that serves content based on caching policies to manage content freshness and ensure that only fresh content is served to a client.

Plamondon addresses the problem in conventional CDNs where the freshness value of content stored in the cache might not accurately reflect whether the content has changed. Plamondon, ¶7 (“If a requested object is stored in the cache, and its freshness value indicates that the cached object is still valid, the object is sent to the client. However, this approach generally suffers from a drawback that the freshness value may not accurately reflect whether the object has changed. Therefore, reliance on freshness values can result in the cache transmitting outdated information to a user, perhaps many times, before the freshness value expires.”). Plamondon always defers forwarding the content from the cache to the requestor until determining that the content is fresh. *Id.*, ¶431 (“At step 560, the appliance 200 determines the object is stored in a cache, such as a cache 232 of the appliance. At step 565, the appliance 200 defers providing the object from the cache until receiving a response for the object forwarded by the appliance to the server 106. At step 570, the appliance 200 determines whether to server the object from the cache based upon the response received from the server 106.”) In this way, Plamondon addresses the problem of determining whether content that may appear to be fresh has actually changed at the server.

A POSITA would have been motivated to combine O'Rourke, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-01 [O'Rourke]) with Plamondon, which describes a comparable CDN for content freshness management (*see, e.g.*, Exhibit F-07 [Plamondon], cls. 1-7). A POSITA would have recognized that O'Rourke discloses storing and delivering content through a CDN and Plamondon discloses a CDN that operates in a

comparable manner. A POSITA would have been motivated to use Plamondon's disclosure addressing the issue of the cache validity not necessarily capturing whether the object has changed in the server. Likewise, a POSITA would have recognized the benefit of O'Rourke in that flexible cache architecture allows for a single cache to be configured in a variety of different ways depending on the set of policies applied to the cache server. This combination of O'Rourke and Plamondon would involve only routine engineering leading to predictable results.

(1) O'Rourke in view of HTTP/1.1

It would have been obvious to combine O'Rourke and HTTP/1.1 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen, O'Rourke teaches a CDN having a flexible cache server that serves content based on caching policies to manage content freshness and ensure that only fresh content is served to a client.

HTTP/1.1 describes an "application-level protocol for distributed, collaborative, hypermedia information systems." HTTP/1.1, p. 5. The standard allows for performance improvement "by the use of response caches. The HTTP/ 1.1 protocol includes a number of elements intended to make caching work as well as possible." *Id.*, p. 38. The caching protocol describes calculating the freshness of content to ensure that stale content is not served to a client. *See id.*, p. 40 ("In order to know if a cached entry is fresh, a cache needs to know if its age exceeds its freshness lifetime. We discuss how to calculate the latter in section 13.2.4; this section describes how to calculate the age of a response or cache entry. In this discussion, we use the term

‘now’ to mean ‘the current value of the clock at the host performing the calculation.’ Hosts that use HTTP, but especially hosts running origin servers and caches, SHOULD use NTP or some similar protocol to synchronize their clocks to a globally accurate time standard. HTTP/ 1. 1 requires origin servers to send a Date header, if possible, with every response, giving the time at which the response was generated (see section 14.18). We use the term ‘date_ value’ to denote the value of the Date header, in a form appropriate for arithmetic operations. HTTP/ 1.1 uses the Age response-header to convey the estimated age of the response message when obtained from a cache. The Age field value is the cache's estimate of the amount of time since the response was generated or revalidated by the origin server. In essence, the Age value is the sum of the time that the response has been resident in each of the caches along the path from the origin server, plus the amount of time it has been in transit along network paths. We use the term ‘age_ value’ to denote the value of the Age header, in a form appropriate for arithmetic operations.”); *see, e.g., id.*, pp. 38-47 (“Caching in HTTP”); *see also id.*, pp. 48-67 (“Header Field Definitions”). Thus, HTTP/1.1 improves upon previous versions of the protocol by including caching and content management. *Id.*, pp. 5-7.

A POSITA would have been motivated to combine O’Rourke, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-01 [O’Rourke]) with the HTTP/1.1 Protocol, which describes managing content freshness in a distributed networks, e.g., CDNs, using the well-known and widely adopted standard (*see, e.g.,* Exhibit F-08 [HTTP], cls. 1[b1]-[b2], 3, 6-7). A POSITA would have recognized that O’Rourke discloses storing and delivering content through a CDN and that HTTP/1.1 discloses the caching standard for determining the content’s policy and expiration. A POSITA would have been motivated to use HTTP/1.1 headers to determine the cache policies of the content because HTTP was a well-known

standard at the time of application, thereby reinforcing that caching policies applied to content freshness, i.e., where determining a current cache policy comprises determining a time period for expiration of content in the cache; and where HTTP requests were used to request content comprising one or more resources were known at the time of application. This combination of O'Rourke and HTTP/1.1 would involve only routine engineering leading to predictable results. Additionally, a POSITA would have found it obvious to use HTTP/1.1 because the communication protocol in a CDN because HTTP/1.1 is a well-known, industry-standard protocol for enabling interoperability between networked devices and web-based applications. The widespread adoption and documentation of HTTP/1.1 for similar enterprise systems would have motivated its use in Windows Media Services implementations without requiring inventive skill.

(2) Middleton in view of Eriksen

It would have been obvious to combine Middleton and Eriksen because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen and O'Rourke in view of Middleton, both Middleton and Eriksen teach serving content from a CDN and are directed to improving the reliability of serving content.

A POSITA would have been motivated to combine Middleton, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-06 [Middleton]) with Eriksen, which describes managing content freshness in a comparable CDN using HTTP/1.1 caching standards to retrieve the content and to calculate content expiration (*see, e.g.,* Exhibit F-02

[Eriksen], cls. 1[b1]-[b2], 3, 6). A POSITA would have recognized that Middleton discloses storing and delivering content through a CDN and that Eriksen discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Eriksen's HTTP headers to determine the cache policies of the content because HTTP was a well-known standard at the time of application. This combination of Middleton and Eriksen would involve only routine engineering leading to predictable results.

(3) Middleton in view of Ben-Shaul

It would have been obvious to combine Middleton and Ben-Shaul because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Ben-Shaul and O'Rourke in view of Middleton, both Middleton and Ben-Shaul are directed to CDNs with customizable caching policies and servers.

A POSITA would have been motivated to combine Middleton, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-06 [Middleton]) with Ben-Shaul, which describes the components of a CDN and where a database (external to a node in the CDN) stores the caching policies and accessed by the nodes to apply policies to content (*see, e.g.*, Exhibit F-03 [Ben-Shaul], cls. 1[b1]-[b2], 4-5). A POSITA would have recognized that Middleton discloses storing and delivering content through a CDN and that Ben-Shaul discloses a CDN that operates in a comparable manner, where both disclosures are directed to improving the flexibility and modification of cache server polies. A POSITA would have been motivated to combine Ben-

Shaul with Middleton to achieve the same improvement in edge server behavior. This combination of Middleton and Ben-Shaul would involve only routine engineering leading to predictable results.

(4) Middleton in view of Ortiz Rodriguez

It would have been obvious to combine Middleton and Ortiz Rodriguez because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Ortiz Rodriguez and O'Rourke in view of Middleton, both Middleton and Ortiz Rodriguez are directed to efficiently storing fresh content to the client.

A POSITA would have been motivated to combine Middleton, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-06 [Middleton]) with Ortiz Rodriguez, which describes the interface between the cache and calculating the expiration time of content within the cache before serving content to the client (*see, e.g.*, Exhibit F-04 [Ortiz Rodriguez], cls. 1[b1]-[b2]). A POSITA would have recognized that Middleton discloses storing and delivering content through a CDN and Ortiz Rodriguez discloses calculating content freshness in a comparable manner by comparing the life of stored content to a cache policy. A POSITA would have been motivated to use Ortiz Rodriguez's TTL calculation in the CDN of Middleton for the purpose of ensuring that only fresh content is delivered to the client. This combination of Middleton and Ortiz Rodriguez would involve only routine engineering leading to predictable results.

(5) Middleton in view of Blumofe

It would have been obvious to combine Middleton and Blumofe because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Blumofe and O'Rourke in view of Middleton, both Middleton and Blumofe are directed to providing CDN-provisioned content to clients. Moreover, as discussed above, Blumofe teaches "extend[ing] [a CDN] into one or more mobile or wireline networks that cannot or do not otherwise support fully-managed CDN edge servers," such as, for example, 3G mobile wireless networks Blumofe, Abstract; *id.*, ¶7. To address this issue and provide caching of CDN-provisioned content closer to end-users, Blumofe describes a CDN Extender, which may be implemented in network environments that do not support CDN services (such as a wireless mobile network), and operates as a cache beyond the edge of the CDN. *Id.*, ¶¶25, 27; *see also id.* at Figure 3.

A POSITA would have been motivated to combine Middleton, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-06 [Middleton]) with Blumofe, which describes a CDN where the caching policies are related to caching content within the CDN and also related to caching content outside of the CDN (e.g., within the Extender) (*see, e.g.*, Exhibit F-05 [Blumofe], cls. 6-7). A POSITA would have recognized that Middleton discloses storing and delivering content through a CDN and Blumofe discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Blumofe's Extender and caching policy relating to storage of content outside of the CDN in order to achieve the taught benefit of serving CDN-provisioned content in locations where doing so would otherwise be unavailable. This

combination of Middleton and Blumofe would involve only routine engineering leading to predictable results. A POSITA would understand that in such a combined system, the Extender operates in a similar manner as the edge servers in order to deliver CDN content to users. *See, e.g., id.* at ¶9 (“[t]he CDN is extended into one or more mobile or wireline networks that cannot or do not otherwise support fully-managed CDN edge servers.”); *id.* (the Extender “is deployed in the mobile or wireline network, preferably as a passive web caching proxy that is beyond the edge of the CDN but that serves CDN-provisioned content.”); *id.* at ¶29 (the Extender “preferably delivers all forms of CDN content (i.e., content that has been off-loaded by content providers for delivery by the CDNSP over the CDN).”

Moreover, a POSITA would understand that in such a combined system, the CDN sends the cache control information to the Extender so that the Extender performs according to the CDN customer’s rules, and the cache policy that configures the cache server within the CDN (e.g., with respect to whether cached content has expired or is an acceptable version) would also be used to configure the Extender outside the CDN. *Id.* at ¶¶ 30, 36, 37, 54.

(6) Middleton and Plamondon

It would have been obvious to combine Middleton and Plamondon because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Plamondon and O'Rourke in view of Middleton, both Middleton and Plamondon both address the issue of ensuring the freshness of content served from a CDN.

A POSITA would have been motivated to combine Middleton, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-06 [Middleton]) with Plamondon, which describes a comparable CDN for content freshness management (*see, e.g.*, Exhibit F-07 [Plamondon], cls. 1-7). A POSITA would have recognized that Middleton discloses storing and delivering content through a CDN and that Plamondon discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to combine Middleton and Plamondon to realize the benefits of Middleton's revalidation which prevents unnecessary downloads from the origin server when the content is expired based on a caching policy but still unchanged from the content in the origin server, thereby reducing network traffic and saving on computational power. Likewise, a POSITA would have been motivated to use Plamondon's disclosure addressing the issue of the cache validity not necessarily capturing whether the object has changed in the server. This combination of Middleton and Plamondon would involve only routine engineering leading to predictable results.

(7) Middleton in view of HTTP/1.1

It would have been obvious to combine Middleton and HTTP/1.1 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of HTTP/1.1 and O'Rourke in view of Middleton, both Middleton and HTTP/1.1 are directed to efficiently storing fresh content to the client.

A POSITA would have been motivated to combine Middleton, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-06 [Middleton]) with the HTTP/1.1 Protocol, which describes managing content freshness in a distributed networks, e.g., CDNs, using the well-known and widely adopted standard (*see, e.g.*, Exhibit F-08 [HTTP], cls. 1[b1]-[b2], 3, 6-7). A POSITA would have recognized that Middleton discloses storing and delivering content through a CDN and that HTTP/1.1 discloses the caching standard for determining the content's policy and expiration. A POSITA would have been motivated to use HTTP/1.1 headers to determine the cache policies of the content because HTTP was a well-known standard at the time of application, thereby reinforcing that caching policies applied to content freshness, i.e., where determining a current cache policy comprises determining a time period for expiration of content in the cache; and where HTTP requests were used to request content comprising one or more resources were known at the time of application. This combination of Middleton and HTTP/1.1 would involve only routine engineering leading to predictable results. Additionally, a POSITA would have found it obvious to use HTTP/1.1 because the communication protocol in a CDN because HTTP/1.1 is a well-known, industry-standard protocol for enabling interoperability between networked devices and web-based applications.

(8) Plamondon in view of Eriksen

It would have been obvious to combine Plamondon and Eriksen because the combination does nothing more than combine prior art elements from similar systems according to known

methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen and O'Rourke in view of Plamondon, both Plamondon and Eriksen teach serving content from a CDN and are directed to improving the reliability of serving content.

A POSITA would have been motivated to combine Plamondon, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-07 [Plamondon]) with Eriksen, which describes managing content freshness in a comparable CDN using HTTP/1.1 caching standards to retrieve the content and to calculate content expiration (*see, e.g.*, Exhibit F-02 [Eriksen], cls. 1[b1]-[b2], 3, 6). A POSITA would have recognized that Plamondon discloses storing and delivering content through a CDN and that Eriksen discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Eriksen's HTTP headers to determine the cache policies of the content because HTTP was a well-known standard at the time of application. This combination of Plamondon and Eriksen would involve only routine engineering leading to predictable results.

(9) Plamondon in view of Ben-Shaul

It would have been obvious to combine Plamondon and Ben-Shaul because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Ben-Shaul and O'Rourke in view of Plamondon, both Plamondon and Ben-Shaul are directed to CDNs with customizable caching policies and servers.

A POSITA would have been motivated to combine Plamondon, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-07 [Plamondon]) with Ben-Shaul, which describes the components of a CDN and where a database (external to a node in the CDN) stores the caching policies and accessed by the nodes to apply policies to content (*see, e.g.*, Exhibit F-03 [Ben-Shaul], cls. 1[b1]-[b2], 4-5). A POSITA would have recognized that Plamondon discloses storing and delivering content through a CDN and that Ben-Shaul discloses a CDN that operates in a comparable manner, where both disclosures are directed to improving the flexibility and modification of cache server policies. A POSITA would have been motivated to combine Ben-Shaul with Plamondon to achieve the same improvement in edge server behavior. This combination of Plamondon and Ben-Shaul would involve only routine engineering leading to predictable results.

(10) Plamondon in view of Ortiz Rodriguez

It would have been obvious to combine Plamondon and Ortiz Rodriguez because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Ortiz Rodriguez and O'Rourke in view of Plamondon, both Plamondon and Ortiz Rodriguez are directed to efficiently storing fresh content to the client.

A POSITA would have been motivated to combine Plamondon, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-07 [Plamondon]) with Ortiz Rodriguez, which describes the interface between the cache and calculating the expiration time of content within the cache before serving content to the client (*see, e.g.*, Exhibit F-04 [Ortiz Rodriguez], cls. 1[b1]-[b2]). A POSITA would have recognized that Plamondon discloses storing and delivering content through a CDN and Ortiz Rodriguez discloses calculating content freshness in a comparable manner by comparing the life of stored content to a cache policy. A POSITA would have been motivated to use Ortiz Rodriguez's TTL calculation in the CDN of Plamondon for the purpose of ensuring that only fresh content is delivered to the client. This combination of Plamondon and Ortiz Rodriguez would involve only routine engineering leading to predictable results.

(11) Plamondon in view of Blumofe

It would have been obvious to combine Plamondon and Blumofe because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Blumofe and O'Rourke in view of Plamondon, both Plamondon and Blumofe are directed to providing CDN-provisioned content to clients.

A POSITA would have been motivated to combine Plamondon, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-07 [Plamondon]) with Blumofe, which describes a CDN where the caching policies are related to caching content within the CDN

and also related to caching content outside of the CDN (e.g., within the Extender) (*see, e.g.*, Exhibit F-05 [Blumofe], cls. 6-7). A POSITA would have recognized that Plamondon discloses storing and delivering content through a CDN and Blumofe discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Blumofe's Extender and caching policy relating to storage of content outside of the CDN in order to achieve the taught benefit of serving CDN-provisioned content in locations where doing so would otherwise be unavailable. This combination of Plamondon and Blumofe would involve only routine engineering leading to predictable results.

(12) Plamondon in view of HTTP/1.1

It would have been obvious to combine Plamondon and HTTP/1.1 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of HTTP/1.1 and O'Rourke in view of Plamondon, both Plamondon and HTTP/1.1 are directed to efficiently storing fresh content to the client.

A POSITA would have been motivated to combine Plamondon, which describes a method of managing content freshness in a CDN (see generally Exhibit F-07 [Plamondon]) with the HTTP/1.1 Protocol, which describes managing content freshness in a distributed networks, e.g., CDNs, using the well-known and widely adopted standard (see, e.g., Exhibit F-08 [HTTP], cls. 1[b1]-[b2], 3, 6-7). A POSITA would have recognized that Plamondon discloses storing and delivering content through a CDN and that HTTP/1.1 discloses the caching standard for

determining the content's policy and expiration. A POSITA would have been motivated to use HTTP/1.1 headers to determine the cache policies of the content because HTTP was a well-known standard at the time of application, thereby reinforcing that caching policies applied to content freshness, i.e., where determining a current cache policy comprises determining a time period for expiration of content in the cache; and where HTTP requests were used to request content comprising one or more resources were known at the time of application. This combination of Plamondon and HTTP/1.1 would involve only routine engineering leading to predictable results. Additionally, a POSITA would have found it obvious to use HTTP/1.1 because the communication protocol in a CDN because HTTP/1.1 is a well-known, industry-standard protocol for enabling interoperability between networked devices and web-based applications.

(13) Roth in view of Blumofe

It would have been obvious to combine Roth and Blumofe because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Blumofe, Blumofe is directed to efficiently storing fresh content to the client by providing CDN-provisioned content to clients. Moreover, as discussed above, Blumofe teaches "extend[ing] [a CDN] into one or more mobile or wireline networks that cannot or do not otherwise support fully-managed CDN edge servers," such as, for example, 3G mobile wireless networks. Blumofe, Abstract; *id.*, ¶7. To address this issue and provide caching of CDN-provisioned content closer to end-users, Blumofe describes a CDN Extender, which may be implemented in network environments that do not support CDN services

(such as a wireless mobile network), and operates as a cache beyond the edge of the CDN. *Id.*, ¶¶25, 27; *see also id.* at Figure 3.

Roth is directed to improving caching control for streaming media using a CDN (i.e., proxy servers) to handle the content as indicated by caching policies (i.e., cache control directives). Roth, Abstract, ¶8. Roth discloses that proxy servers provide “a centralized location for multiple client computers to access the Internet, easing security management for the system administrator [and] may also cache content and serve the content to requesting client computers, thereby alleviating the server computer from the burden of providing the same content to multiple client computers behind the same proxy server.” *Id.*, ¶4. In Roth, after the proxy server determines that the content is cached locally (act 324), the proxy server checks the validity (act 326). *Id.*, ¶91 (“if the requested content stream is in the cache, then proxy server 108 checks whether the content stream is valid (act 326).”), Fig. 4a. If the content is not valid, the proxy server revalidates the content, which involves “the proxy server retriev[ing] new cache control information...and a new expiration time...from the origin server” *Id.*, ¶79. The origin server may use HTTP headers to communicate the cache control information, including expiration times, to the proxy servers. *Id.*, ¶38, 42, ¶83 (“Changes to the content can be detected by checking one or more of various parameters for the content. In one implementation, when revalidating content the origin server returns an indication of the last modified date (e.g., this may be a header 204 of FIG. 3).”), ¶85 (“The expiration time for streaming media content can be indicated in a variety of different manners. In one implementation, the expiration time is indicated using the HTTP max-age directive. In another implementation, the expiration time is indicated using the HTTP or RTSP Expires header.”), Fig. 3.

A POSITA would have been motivated to combine Roth, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-15 [Roth]), with Blumofe, which describes a CDN where the caching policies are related to caching content within the CDN and also related to caching content outside of the CDN (e.g., within the Extender) (*see, e.g.*, Exhibit F-05 [Blumofe], cls. 6-7). A POSITA would have recognized that Roth discloses storing and delivering content through a CDN and Blumofe discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Blumofe’s Extender and caching policy relating to storage of content outside of the CDN in order to achieve the taught benefit of serving CDN-provisioned content in locations where doing so would otherwise be unavailable. This combination of Roth and Blumofe would involve only routine engineering leading to predictable results. A POSITA would understand that in such a combined system, the Extender operates in a similar manner as the proxy / edge servers in order to deliver CDN content to users. *See, e.g., id.* at ¶9 (“[t]he CDN is extended into one or more mobile or wireline networks that cannot or do not otherwise support fully-managed CDN edge servers.”); *id.* (the Extender “is deployed in the mobile or wireline network, preferably as a passive web caching proxy that is beyond the edge of the CDN but that serves CDN-provisioned content.”); *id.* at ¶29 (the Extender “preferably delivers all forms of CDN content (i.e., content that has been off-loaded by content providers for delivery by the CDN-SP over the CDN).”)

Moreover, a POSITA would understand that in such a combined system, the CDN sends the cache control information to the Extender so that the Extender performs according to the CDN customer’s rules, and the cache policy that configures the cache server within the CDN (e.g., with respect to whether cached content has expired or is an acceptable version) would also be used to configure the Extender outside the CDN. *Id.* at ¶¶30, 36, 37, 54.

(14) Roth in view of HTTP/1.1

It would have been obvious to combine Roth and HTTP/1.1 because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of HTTP/1.1 and Roth in view of Blumofe, both Roth and HTTP/1.1 are directed to efficiently storing content to the client according to cache policy directives. Roth also explicitly discloses using HTTP/1.1 to both deliver content and deliver cache control directives e.g., content expiration) via HTTP header requests. Roth, ¶¶32, 43, 72, 85.

A POSITA would have been motivated to combine Roth, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-15 [Roth]), with the HTTP/1.1 Protocol, which describes managing content freshness in a distributed networks, e.g., CDNs, using the well-known and widely adopted standard (*see, e.g.*, Exhibit F-08 [HTTP], cls. 1[b1]-[b2], 3, 6-7). A POSITA would have recognized that Roth discloses storing and delivering content through a CDN using HTTP/1.1 delivery protocol and that HTTP/1.1 discloses the caching standard for determining the content's policy and expiration. A POSITA would have been motivated to use HTTP/1.1 headers to determine the cache policies of the content because HTTP was a well-known standard at the time of application, thereby reinforcing that caching policies applied to content freshness, i.e., where determining a current cache policy comprises determining a time period for expiration of content in the cache; and where HTTP requests were used to request content comprising one or more resources were known at the time of application. This combination of Roth and HTTP/1.1 would involve only routine engineering leading to

predictable results. Additionally, a POSITA would have found it obvious to use HTTP/1.1 because the communication protocol in a CDN because HTTP/1.1 is a well-known, industry-standard protocol for enabling interoperability between networked devices and web-based applications.

(15) Roth in view of Eriksen

It would have been obvious to combine Roth and Eriksen because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen and Roth in view of Blumofe, both Roth and Eriksen teach serving content from a CDN and are directed to improving the reliability of serving content.

A POSITA would have been motivated to combine Roth, which describes a method of managing content in a CDN according to caching control policies/ directives (*see generally* Exhibit F-15 [Roth]), with Eriksen, which describes managing content freshness in a comparable CDN using HTTP/1.1 caching standards to retrieve the content and to calculate content expiration (*see, e.g.*, Exhibit F-02 [Eriksen], cls. 1[b1]-[b2], 3, 6). A POSITA would have recognized that Roth discloses storing and delivering content through a CDN and that Eriksen discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Roth's and Eriksen's HTTP headers to determine the cache policies of the content because HTTP was a well-known standard at the time of application. This combination of Roth and Eriksen would involve only routine engineering leading to predictable results.

(16) Roth in view of Ben-Shaul

It would have been obvious to combine Roth and Ben-Shaul because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Ben-Shaul and Roth in view of Blumofe, both Roth and Ben-Shaul are directed to CDNs with customizable caching policies and servers.

A POSITA would have been motivated to combine Roth, which describes a method of managing content in a CDN according to caching control policies/ directives (*see generally* Exhibit F-15 [Roth]), with Ben-Shaul, which describes the components of a CDN and where a database (external to a node in the CDN) stores the caching policies and accessed by the nodes to apply policies to content (*see, e.g.*, Exhibit F-03 [Ben-Shaul], cls. 1[b1]-[b2], 4-5). A POSITA would have recognized that Roth discloses storing and delivering content through a CDN and that Ben-Shaul discloses a CDN that operates in a comparable manner, where both disclosures are directed to improving the flexibility and modification of cache server polies. A POSITA would have been motivated to combine Ben-Shaul with Roth to achieve the same improvement in edge server behavior. This combination of Roth and Ben-Shaul would involve only routine engineering leading to predictable results.

(17) Roth in view of Ortiz Rodriguez

It would have been obvious to combine Roth and Ortiz Rodriguez because the combination does nothing more than combine prior art elements from similar systems according to known

methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Ortiz Rodriguez and Roth in view of Blumofe, both Roth and Ortiz Rodriguez are directed to efficiently storing fresh content to the client according to cache policies.

A POSITA would have been motivated to combine Roth, which describes a method of managing content in a CDN according to caching control policies/ directives (*see generally* Exhibit F-15 [Roth]), with Ortiz Rodriguez, which describes the interface between the cache and calculating the expiration time of content within the cache before serving content to the client (*see, e.g.,* Exhibit F-04 [Ortiz Rodriguez], cls. 1[b1]-[b2]). A POSITA would have recognized that Roth discloses storing and delivering content through a CDN and Ortiz Rodriguez discloses calculating content freshness in a comparable manner by comparing the life of stored content to a cache policy. A POSITA would have been motivated to use Ortiz Rodriguez's TTL calculation in the CDN of Roth for the purpose of ensuring that only fresh content is delivered to the client. This combination of Roth and Ortiz Rodriguez would involve only routine engineering leading to predictable results.

(18) Roth in view of O'Rourke

It would have been obvious to combine Roth and O'Rourke because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Eriksen and Roth in view of Blumofe, O'Rourke and Roth both address the issue of storing content in and delivering content from a CDN using cache control policies / directives.

A POSITA would have been motivated to combine Roth, which describes a method of managing content in a CDN according to caching control policies/ directives (*see generally* Exhibit F-15 [Roth]), with O'Rourke, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-01 [O'Rourke], cls. 1-7). A POSITA would have recognized that O'Rourke discloses storing and delivering content through a CDN and Roth discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Roth's disclosure addressing the issue of the cache revalidation. Likewise, a POSITA would have recognized the benefit of O'Rourke in that flexible cache architecture allows for a single cache to be configured in a variety of different ways depending on the set of policies applied to the cache server. This combination of O'Rourke and Roth would involve only routine engineering leading to predictable results.

(19) Roth in view of Middleton

It would have been obvious to combine Middleton and Roth because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding Roth in view of Blumofe and O'Rourke in view of Middleton, both Middleton and Roth both address the issue of ensuring the freshness of content served from a CDN.

A POSITA would have been motivated to combine Roth, which describes a method of managing content in a CDN according to caching control policies/ directives (*see generally* Exhibit F-15 [Roth]), with Middleton, which describes a method of managing content freshness in a CDN (*see generally* Exhibit F-06 [Middleton], cls. 1-7). A POSITA would have recognized that Middleton discloses storing and delivering content through a CDN and that Roth discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to combine Middleton and Roth to realize the benefits of Middleton's revalidation which prevents unnecessary downloads from the origin server when the content is expired based on a caching policy but still unchanged from the content in the origin server, thereby reducing network traffic and saving on computational power. Likewise, a POSITA would have been motivated to use Roth's disclosure addressing the issue of the cache revalidation. This combination of Middleton and Roth would involve only routine engineering leading to predictable results.

(20) Roth in view of Plamondon

It would have been obvious to combine Roth and Plamondon because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

As discussed above regarding O'Rourke in view of Plamondon and Roth in view of Blumofe, both Roth and Plamondon both address the issue of using cache policies to store content in and server content from proxy servers in a CDN.

A POSITA would have been motivated to Roth, which describes a method of managing content in a CDN according to caching control policies/ directives (*see generally* Exhibit F-15

[Roth]), with Plamondon, which describes a comparable CDN for content freshness management (*see, e.g.*, Exhibit F-07 [Plamondon], cls. 1-7). A POSITA would have recognized that Roth discloses storing and delivering content through a CDN and that Plamondon discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Roth's disclosure addressing the issue of the cache revalidation. Likewise, a POSITA would have been motivated to use Plamondon's disclosure addressing the issue of the cache validity not necessarily capturing whether the object has changed in the server. This combination of Roth and Plamondon would involve only routine engineering leading to predictable results.

(21) Windows Media Services in view of Eriksen

It would have been obvious to combine Windows Media Services and Eriksen because the combination does nothing more than combine prior art elements from similar systems according to known methods to yield the predictable results of providing an improved system having the known and ordinary functionality of the added features.

Windows Media Services deployments often rely on cache policies for streaming media, including TTL and freshness checks. Eriksen teaches determining cache freshness using HTTP/1.1 headers and Cache-Control directives. A POSITA would have been motivated to apply Eriksen's HTTP-based freshness checks to Windows Media Services because HTTP was a well-known standard for web-based content delivery, and using HTTP headers to enforce cache policies would improve interoperability and ensure accurate freshness validation. This combination would involve routine engineering leading to predictable results.

(22) Windows Media Services in view of Ben-Shaul

It would have been obvious to combine Windows Media Services and Ben-Shaul because both disclosures address flexible caching policies. Windows Media Services uses plug-in modules

for cache policy enforcement, while Ben-Shaul teaches centralized policy control and dynamic updates via XML-based directives. A POSITA would have been motivated to integrate Ben-Shaul's centralized policy management with Windows Media Services to allow fine-grained control over streaming content caching and delivery. This is a known design choice for improving scalability and policy synchronization across distributed servers.

(23) Windows Media Services in view of Ortiz Rodriguez

It would have been obvious to combine Windows Media Services and Ortiz Rodriguez because both disclosures address TTL-based freshness checks. Ortiz Rodriguez teaches calculating expiration dates and comparing them to cache policies. A POSITA would have been motivated to incorporate Ortiz Rodriguez's TTL calculation into Windows Media Services to ensure only fresh media streams are served, reducing stale content delivery. This combination would involve routine engineering leading to predictable results.

(24) Windows Media Services in view of Blumofe

It would have been obvious to combine Windows Media Services and Blumofe because Blumofe teaches extending CDN functionality to unmanaged environments using Extenders. A POSITA would have been motivated to apply this concept to Windows Media Services to enable streaming content delivery in locations where deploying full Windows Media Services clusters is impractical. This would allow consistent policy enforcement and improve reach, a predictable result of combining known techniques.

(25) Windows Media Services in view of Middleton

It would have been obvious to combine Windows Media Services and Middleton because both disclosures teach extensible frameworks for cache customization. Middleton's modular architecture complements Windows Media Services' plug-in approach, enabling dynamic policy

enforcement and custom logic for streaming scenarios. A POSITA would have been motivated to combine these to improve flexibility and reduce unnecessary origin fetches, a routine design choice.

(26) Windows Media Services in view of Plamondon

It would have been obvious to combine Windows Media Services and Plamondon because Plamondon addresses validating freshness beyond TTL values. A POSITA would have been motivated to incorporate Plamondon's revalidation approach into Windows Media Services to prevent serving outdated media streams, reducing network traffic and improving reliability. This combination would involve routine engineering leading to predictable results.

(27) Windows Media Services in view of Roth

A POSITA would have been motivated to combine Windows Media Services with Roth because Roth because they both relate related to caching control. A POSITA would have further been motivated to combine Windows Media Services with other Microsoft patents and publications such as Roth. A POSITA would be motivated to review other teachings from Microsoft for other potential functionality in the system.

(28) Squid in view of Eriksen

It would have been obvious to combine Squid and Eriksen because Squid's proxy caching could benefit from Eriksen's HTTP/1.1 freshness checks. A POSITA would have been motivated to implement HTTP-based cache validation in Squid to ensure compliance with web standards and improve cache accuracy. <https://web.archive.org/web/20080124172027/http://www.squid-cache.org/>. This is applying a known technique to a known system for predictable results.

(29) Squid in view of Ben-Shaul

It would have been obvious to combine Squid and Ben-Shaul because Ben-Shaul teaches centralized policy control for edge servers. A POSITA would have been motivated to integrate

this with Squid to allow dynamic cache policy updates across distributed proxies, improving flexibility and reducing manual configuration. This combination would involve routine engineering.

(30) Squid in view of Ortiz Rodriguez

It would have been obvious to combine Squid and Ortiz Rodriguez because Ortiz Rodriguez teaches TTL-based expiration checks (§56). A POSITA would have been motivated to apply this to Squid to ensure stale objects are invalidated efficiently, improving reliability. This is a routine design choice.

(31) Squid in view of Blumofe

It would have been obvious to combine Squid and Blumofe because Blumofe's Extender concept (§36) could complement Squid's proxy caching by enabling CDN-controlled caching outside traditional infrastructure. A POSITA would have been motivated to adopt this approach to extend Squid's reach and enforce consistent policies, a predictable improvement.

(32) Squid in view of Middleton

It would have been obvious to combine Squid and Middleton because Middleton's modular framework (§105-110) could enhance Squid's extensibility for custom caching logic. A POSITA would have been motivated to integrate these features to support dynamic policy enforcement, a routine engineering step.

(33) Squid in view of Plamondon

It would have been obvious to combine Squid and Plamondon because Plamondon's approach (§431) would prevent Squid from serving outdated content when TTL values are inaccurate. A POSITA would have been motivated to implement this to improve cache reliability, a predictable result.

(34) Akamai EdgeSuite in view of Eriksen

It would have been obvious to combine Akamai EdgeSuite and Eriksen to further improve Akamai EdgeSuite's time-to-live and freshness checks. Turbo-Charging EdgeSuite at 6, 8. A POSITA would be motivated to research additional methods for providing fresh content and would have reviewing Eriksen's HTTP/1.1 freshness checks (¶42). This combination of Akamai EdgeSuite and Eriksen would involve only routine engineering leading to predictable results. Additionally, a POSITA would have found it obvious to use HTTP/1.1, as disclosed by Eriksen, as the communication protocol in a CDN because HTTP/1.1 is a well-known, industry-standard protocol for enabling interoperability between networked devices and web-based applications.

(35) Akamai EdgeSuite in view of Ben-Shaul

It would have been obvious to combine Akamai EdgeSuite and Ben-Shaul because Ben-Shaul's centralized policy control would allow Akamai EdgeSuite to offer fine-grained caching directives to content providers. A POSITA would have been motivated to implement this for improved flexibility and dynamic updates, a routine design choice. A POSITA would have been motivated to combine Akamai EdgeSuite, which describes a method of managing content freshness in a CDN (see generally Exhibit F-14 [Akamai EdgeSuite]) with Ben-Shaul, which describes the components of a CDN and where a database (external to a node in the CDN) stores the caching policies and accessed by the nodes to apply policies to content (*see, e.g.*, Exhibit F-03 [Ben-Shaul], cls. 1[b1]-[b2], 4-5). A POSITA would have recognized that Akamai EdgeSuite discloses storing and delivering content through a CDN and that Ben-Shaul discloses a CDN that operates in a comparable manner, where both disclosures are directed to improving the flexibility and modification of cache server polies. A POSITA would have been motivated to combine Ben-Shaul with Akamai EdgeSuite to achieve the same improvement in edge server behavior. A POSITA would have found it further obvious to store policy information on a remote server

because centralized storage facilitates easier management, updating, and access control across distributed client devices. The use of remote servers for storing and managing shared data is well-established in networked computing environments, making its application to policy information a routine design choice rather than an inventive step. This combination of Akamai EdgeSuite and Ben-Shaul would involve only routine engineering leading to predictable results.

(36) Akamai EdgeSuite in view of Ortiz Rodriguez

It would have been obvious to combine Akamai EdgeSuite and Ortiz Rodriguez because Ortiz Rodriguez's TTL calculation would improve Akamai EdgeSuite's freshness validation. A POSITA would have been motivated to adopt this to ensure timely content delivery, a predictable improvement.

A POSITA would have been motivated to combine Akamai EdgeSuite, which describes a method of managing content freshness in a CDN (see generally Exhibit F-14 [Akamai EdgeSuite]) with Ortiz Rodriguez, which describes the interface between the cache and calculating the expiration time of content within the cache before serving content to the client (see, e.g., Exhibit F-04 [Ortiz Rodriguez], cls. 1[b1]-[b2]). A POSITA would have recognized that Akamai EdgeSuite discloses storing and delivering content through a CDN and Ortiz Rodriguez discloses calculating content freshness in a comparable manner by comparing the life of stored content to a cache policy. A POSITA would have been motivated to use Ortiz Rodriguez's expiration calculation in the CDN of Akamai EdgeSuite for the purpose of ensuring that only fresh content is delivered to the client. A POSITA would have found it obvious to employ a service agent to collect policy information and calculate expiration of content, as service agents are commonly used to automate data retrieval and processing tasks in distributed systems. The delegation of expiration computation to such agents is a routine design choice that streamlines policy enforcement and resource management, reflecting established practices in network and systems

administration, for example, as disclosed by Ortiz Rodriguez. This combination of Akamai EdgeSuite and Ortiz Rodriguez would involve only routine engineering leading to predictable results.

(37) Akamai EdgeSuite in view of Blumofe

It would have been obvious to combine Akamai EdgeSuite and Blumofe because Blumofe's Extender concept would enable Akamai EdgeSuite to serve CDN-provisioned content in unmanaged environments. A POSITA would have been motivated to implement this to expand reach and enforce policies consistently.

A POSITA would have been motivated to combine Akamai EdgeSuite, which describes a method of managing content freshness in a CDN (see generally Exhibit F-14 [Akamai EdgeSuite]) with Blumofe, which describes a CDN where the caching policies are related to caching content within the CDN and also related to caching content outside of the CDN (e.g., within the Extender) (see, e.g., Exhibit F-05 [Blumofe], cls. 6-7). A POSITA would have recognized that Akamai EdgeSuite discloses storing and delivering content through a CDN and Blumofe discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Blumofe's Extender and caching policy relating to storage of content outside of the CDN in order to achieve the taught benefit of serving CDN-provisioned content in locations where doing so would otherwise be unavailable. This combination of Akamai EdgeSuite and Blumofe would involve only routine engineering leading to predictable results.

(38) Akamai EdgeSuite in view of Middleton

It would have been obvious to combine Akamai EdgeSuite and Middleton because Middleton's modular framework would enhance Akamai EdgeSuite's ability to customize cache logic per content or user. A POSITA would have been motivated to integrate this for flexibility, a routine improvement.

A POSITA would have been motivated to combine Akamai EdgeSuite, which describes a method of managing content freshness in a CDN (see generally Exhibit F-14 [Akamai EdgeSuite]) with Middleton, which describes a comparable CDN for content freshness management (see, e.g., Exhibit F-06 [Middleton], cls. 1-7). A POSITA would have recognized that Akamai EdgeSuite discloses storing and delivering content through a CDN and that Middleton discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to combine Middleton and Akamai EdgeSuite to realize the benefits of Middleton's revalidation which prevents unnecessary downloads from the origin server when the content is expired based on a caching policy but still unchanged from the content in the origin server, thereby reducing network traffic and saving on computational power. This combination of Akamai EdgeSuite and Middleton would involve only routine engineering leading to predictable results.

(39) Akamai EdgeSuite in view of Plamondon

It would have been obvious to combine Akamai EdgeSuite and Plamondon because Plamondon's revalidation approach would prevent Akamai EdgeSuite from serving outdated content when TTL values are inaccurate. A POSITA would have been motivated to implement this to improve reliability, a predictable result.

A POSITA would have been motivated to combine Akamai EdgeSuite, which describes a method of managing content freshness in a CDN (see generally Exhibit F-14 [Akamai EdgeSuite]) with Plamondon, which describes a comparable CDN for content freshness management (see, e.g., Exhibit F-07 [Plamondon], cls. 1-7). A POSITA would have recognized that Akamai EdgeSuite discloses storing and delivering content through a CDN and Plamondon discloses a CDN that operates in a comparable manner. A POSITA would have been motivated to use Plamondon's disclosure addressing the issue of the cache validity not necessarily capturing whether the object

has changed in the server. This combination of Akamai EdgeSuite and Plamondon would involve only routine engineering leading to predictable results.

VII. OBJECTIVE INDICIA OF NONOBVIOUSNESS

Sandpiper bears the burden of producing any evidence of secondary considerations that allegedly shows that any Asserted Claim of the Asserted Patents would not have been obvious. *See, e.g., Intercontinental Great Brands LLC v. Kellogg N. Am. Co.*, 869 F.3d 1336, 1345-47 (Fed. Cir. 2017); *Graham v. John Deere Co.*, 383 U.S. 1, 17-18 (1966). Sandpiper has not identified any evidence of secondary considerations. Microsoft reserves the right to rebut any secondary considerations that Sandpiper subsequently alleges.

VIII. INVALIDITY UNDER 35 U.S.C. § 112

The grounds identified below both individually and collectively render certain of the Asserted Claims invalid under the statutory requirements of § 112. By identifying certain claim language below, Microsoft does not imply that such language is entitled to any patentable weight when comparing the claim as a whole to the prior art. Microsoft's identifications are made based on Microsoft's present understanding of the Asserted Claims and Sandpiper's apparent interpretation of these claims as reflected in its Infringement Contentions, and Microsoft reserves the right to amend these identifications, including in response to claim constructions and claim interpretations that would render claim limitations not enabled, lacking in written description, or indefinite. To the extent a claim element is contained within an element identified below or encompass an element identified below, that claim element also renders the claim invalid under 35 U.S.C. § 112.

A. Lack of Enablement and Written Description under Pre-AIA 35 U.S.C. § 112 ¶ 1 or AIA § 112(a)

35 U.S.C. § 112 includes an enablement requirement: “The specification shall contain a written description . . . of the manner and process of making and using [the invention] in such full, clear, concise and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same.” Pre-AIA 35 U.S.C. § 112, ¶ 1; *see also* 35 U.S.C. § 112(a). To satisfy the enablement requirement, the disclosure “must teach those skilled in the art how to make and use the full scope of the claimed invention without ‘undue experimentation.’” *Genentech, Inc. v. Novo Nordisk A/S*, 108 F.3d 1361, 1366 (Fed. Cir. 1997); *MagSil Corp. v. Hitachi Glob. Storage Techs., Inc.*, 687 F.3d 1377, 1381 (Fed. Cir. 2012); *Sitrick v. Dreamworks, LLC*, 516 F.3d 993, 999 (Fed. Cir. 2008). If a specification teaches away from a substantial portion of the claim or does not enable the full scope of the claim, there is no enablement. *AK Steel Corp. v. Sollac*, 344 F.3d 1234 (Fed. Cir. 2003); *see also MagSil Corp.*, 687 F.3d at 1383-84 (Fed. Cir. 2012).

35 U.S.C. § 112 further includes a written description requirement: “The specification shall contain a written description of the invention” 35 U.S.C. § 112, ¶ 1; *see also* 35 U.S.C. § 112(a). “To satisfy the written description requirement, a patent applicant must convey with reasonable clarity to those skilled in the art that, as of the filing date sought, he or she was in possession of the invention.” *ICU Medical Inc. v. Alaris Medical Systems, Inc.*, 558 F.3d 1368, 1377 (Fed. Cir. 2009) (internal quotation marks and citations omitted); *see also Synthes USA, LLC v. Spinal Kinetics, Inc.*, 734 F.3d 1332, 1340 (Fed. Cir. 2013). “The test [for written description support] requires an objective inquiry into the four corners of the specification from the perspective of a person of ordinary skill in the art. Based on that inquiry, the specification must describe an invention understandable to that skilled artisan and show that the inventor actually invented the

invention claimed.” *Ariad Pharms., Inc. v. Eli Lilly & Co.*, 598 F.3d 1336, 1351 (Fed. Cir. 2010) (en banc).

The specification must describe the claimed invention in sufficient detail so that a POSITA can recognize what is claimed. “The appearance of mere indistinct words in a specification or a claim, even an original claim, does not necessarily satisfy that requirement.” *University of Rochester v. G.D. Searle & Co.*, 358 F.3d 916, 923 (Fed. Cir. 2004) (internal quotation marks and citations omitted).

Based on Microsoft’s present understanding of the Asserted Claims and Sandpiper’s apparent interpretation of these claims as reflected in its Infringement Contentions, the following limitations of the Asserted Claims may fail to satisfy the requirements of § 112, ¶ 1 because the specification and original patent application fail to provide an enabling disclosure of and written description support for at least the following limitations:

1. U.S. Patent No. 8,478,903

Term	Relevant Claim(s)
“associating the at least one repeater server with a first alias name” / “associating the at least one repeater server with a second alias name”	28
“associating the first resource with the first alias name”	29, 30, 31
“wherein requests for a first resource located on a first origin server are directed, based at least in part on said first alias name” / “wherein requests for a second resource located on a second origin server are directed, based at least in part on said second alias name”	28
“wherein the second origin server is distinct from the first origin server”	28
“wherein requests for a first resource originating on the first origin server are directed, based at least in part on said first alias name” / “wherein requests for a second resource originating on the second origin server are directed, based at least in part on said second alias name”	37
“providing a table listing origin servers having content located thereon”	28
“providing at least one table associating alias names with origin servers having content located thereon”	37

Term	Relevant Claim(s)
“wherein the at least one repeater server is further constructed and adapted to analyze, using the table, an alias name received with a client request for a particular resource to determine an origin server associated with the particular resource”	28
“identifying the at least one shared repeater server from a set of shared repeater servers based at least in part on a network cost of delivering resources to the particular client machine”	29
“identifying the at least one repeater server from a set of shared repeater servers based at least in part on load characteristics of the set of shared repeater servers”	30
“identifying the at least one repeater server from a set of shared repeater servers based at least in part on a location of the particular client machine relative to the at least one shared repeater server”	31
“wherein a copy of the at least one table is located on each of a plurality of repeater servers”	38
“associating a plurality of alias names with the at least one shared repeater server”	37
“a second origin server distinct from said first origin server”	37
“rejecting the client request for the particular resource if the alias name received with the client request is not included the table” / “rejecting the client request for the resource if the alias name received with the client request is not included the table”	36, 39

2. U.S. Patent No. 8,924,466

Term	Relevant Claim(s)
“said first edge server [site] was selected based at least in part on network traffic conditions”	1, 7, 10
“sending the client a response that causes the client to be redirected to a second server [site] in the CDN to handle the request”	1, 7, 10
“receiving, at the second server [site], a second request for the object from the client and serving the requested object to the client from the second server [site]”	1, 7, 10
“if a copy of the requested object is not stored on the second server [site], directing the client to another server [site] in the CDN”	1, 7, 10

3. U.S. Patent No. 9,456,053

Term	Relevant Claim(s)
“by hardware and software on a particular cache server”	27
“selectively obtaining updated global configuration data”	27
“obtaining customer configuration information associated with a particular customer of the CDN”	27

Term	Relevant Claim(s)
“from one or more other locations in the CDN” / “from one or more locations in the CDN”	28, 40
“wherein the global configuration data comprises at least one CDN resource”	35
“wherein the one or more other locations in the CDN comprise at least one location selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site” / “one or more locations selected from: a cache server, a cache server site, a region of cache servers, a cache cluster, and a cache cluster site”	37, 41
“wherein the one or more other locations in the CDN comprise at least one peer location”	38
“wherein the control core comprises a distributed system consisting of a plurality of machines”	42
“determining validity of a version of global configuration data” / “determining validity of a version of particular customer configuration information”	44, 45, 46
“determines that said version . . . is invalid”	44, 45
“collector system comprising one or more collector mechanisms”	47, 48

4. U.S. Patent No. 9,722,883

Term	Relevant Claim(s)
“configurable policy defines a mapping from a number of discrete resource types to a request peering type with a capacity allocation for each responsibility in the request peering type”	1
“each node in the group assumes one or more discrete responsibilities involved in collaborative processing of a request across the group”	1
“(B) determining what kind of responsibility said node has for said request based on at least one kind of responsibility assigned to said node for requests relating to said particular resource, and [] said mapping”	1
“(C) based on said determining in (B), attempting to process said request in accordance with the type of said particular resource and the kind of responsibility said node has for said particular resource	1
“discrete set of resource types”	5
“discrete set of kinds of responsibilities”	7

5. U.S. Patent No. 9,762,692

Term	Relevant Claim(s)
“if the resource is not available at the first server or at a peer of the first server”	1
“wherein the first portion of the content provider's library is distinct from the second portion”	1
“determining if the resource is popular”	1
“wherein portions of the content provider's library are logically partitioned across servers in the second tier”	2
“wherein the content-types on each of the two or more servers are proportionally distributed”	5
“wherein the step of determining whether the resource is popular comprises determining whether a current popularity value for the resource exceeds a popularity threshold”	8
“if the resource is not available at the first server or at a peer of the first server”	15
“the redirecting being based, at least in part, on a current popularity value for the resource”	15
“wherein at least one other second tier server comprises a second portion of the content provider's library distinct from the first portion”	15
“wherein the first tier of servers comprises a plurality of edge servers and the second tier of servers comprises a plurality of parent servers, and wherein the plurality of tiers of servers further comprises an origin tier, and wherein the first server is a parent server in the second tier and the second server is a server in the origin tier”	18

6. U.S. Patent No. 10,701,173

Term	Relevant Claim(s)
“in response to a request for content, determining, by a node in the CDN, if the content is cached locally at the node”	1
“determining, by the node, a current cache policy associated with the content, wherein determining the current cache policy comprises determining a time period for expiration of content in the cache”	1
“when it is determined that said content is not cached locally, then (C)(1) obtaining the content and then (C)(2) serving the content obtained in (C)(1)”	2
“wherein the current cache policy is determined from a location distinct from the node”	4

B. Indefiniteness Under 35 U.S.C. § 112 ¶ 2

Section 112, ¶ 2 includes a definiteness requirement: “[T]he specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.” 35 U.S.C. § 112, ¶ 2; *see also* 35 U.S.C. § 112(b). “[A] patent is invalid for indefiniteness if its claims, read in light of the patent’s specification and prosecution history, fail to inform, with reasonable certainty, those skilled in the art about the scope of the invention.” *Nautilus, Inc. v. Biosig Instruments, Inc.*, 134 S. Ct. 2120, 2124 (2014).

The definiteness requirement requires that the claim must set forth what the applicant regards as the invention, and do so with sufficient particularity and definiteness. *Allen Eng’g Corp. v. Bartell Indus.*, 299 F.3d 1336, 1348 (Fed. Cir. 2002). Where it would be apparent to one of skill in the art, based on the patent specification, that the “invention” set forth in a claim is not what the patent applicant regarded as the invention, the claim is invalid. *Id.*

Based on Microsoft’s present understanding of the Asserted Claims and Sandpiper’s apparent interpretation of these claims as reflected in its Infringement Contentions, the following limitations of the Asserted Claims may also fail to satisfy the requirements of § 112, ¶ 2.

1. U.S. Patent No. 9,762,692

Dependent claim 18 of the ’692 Patent, appears indefinite when read with this limitation due to multiple (and contradictory) interpretations. Claim 18 specifies that “the first server is a parent server in the second tier” and “the second tier of servers comprises a plurality of parent servers.” Read together, then, limitation 15[b] and claim 18 collectively require that “if the resource is not available at [a parent server in the second tier]...then selectively redirecting the request from the client to a second tier server [(e.g., one of a plurality of parent servers)], the second tier server [(e.g., one of a plurality of parent servers)] being in a tier distinct from the tier containing the [the parent server in the second tier].” In sum, the claim requires redirecting a request

from one second-tier parent server to another second-tier parent server, yet the two are nonetheless in distinct tiers. This limitation in the context of claim 18 may be interpreted any of three ways:

(1) “if the resource is not available at [a parent server in the second tier]...then selectively redirecting the request from the client to a second tier server,” meaning the request is redirected to a peer-parent server, but not one that would be “in a tier distinct from the tier containing the first server” as required by the claim;

(2) “if the resource is not available at [a parent server in the second tier]...then selectively redirecting the request from the client to a second tier server, the second tier server being in a tier distinct from the tier containing the first server” in which the “second tier” of parent servers has a multi-tier substructure of parent servers within it;

(3) “if the resource is not available at [a parent server in the second tier]...then selectively redirecting the request from the client to a second server”—not a “second tier server” as claimed—which claim 18 specifies is an origin server, with “the second [origin] server being in a tier distinct from the tier containing the [the parent server in the second tier]”.

1. U.S. Patent No. 8,924,466

The precise scope of at least the terms listed below cannot be determined with reasonable certainty by a POSITA when reading the claims in light of the specification and prosecution history.

Term	Relevant Claim(s)
“said first edge server [site] was selected based at least in part on network traffic conditions”	1, 7, 10
“sending the client a response that causes the client to be redirected to a second server [site] in the CDN to handle the request”	1, 7, 10
“receiving, at the second server [site], a second request for the object from the client and serving the requested object to the client from the second server [site]”	1, 7, 10
“if a copy of the requested object is not stored on the second server [site], directing the client to another server [site] in the CDN”	1, 7, 10

2. U.S. Patent No. 9,722,883

The precise scope of at least the terms listed below cannot be determined with reasonable certainty by a POSITA when reading the claims in light of the specification and prosecution history.

Term	Relevant Claim(s)
“configurable policy defines a mapping from a number of discrete resource types to a request peering type with a capacity allocation for each responsibility in the request peering type”	1
“each node in the group assumes one or more discrete responsibilities involved in collaborative processing of a request across the group”	1
“(B) determining what kind of responsibility said node has for said request based on at least one kind of responsibility assigned to said node for requests relating to said particular resource, and [] said mapping”	1
“(C) based on said determining in (B), attempting to process said request in accordance with the type of said particular resource and the kind of responsibility said node has for said particular resource	1
“discrete set of resource types”	5
“discrete set of kinds of responsibilities”	7

IX. INVALIDITY UNDER 35 U.S.C. § 101

See attached charts identified at A101, B101, C101, D101, E101, and F101.

X. DOCUMENT PRODUCTION ACCOMPANYING INVALIDITY CONTENTIONS

Microsoft is producing and/or making available for inspection documents required under the Court’s Scheduling Order. Microsoft reserves the right to produce and rely on additional documents relating to its products in view of, for example, additional information revealed during discovery regarding Sandpiper’s allegations and/or amendments to Sandpiper’s Infringement Contentions as discovery processes, including the testimony of inventors, authors and publishers of the printed publications and prior art systems identified herein.

Date: December 5, 2025

Respectfully submitted,

/s/ Benjamin C. Elacqua

Benjamin C. Elacqua (Lead Counsel)

Texas Bar Number 24055443

elacqua@fr.com

Bailey K. Benedict

Texas Bar Number 24083139

benedict@fr.com

Fish & Richardson P.C.

909 Fannin Street, Suite 2100

Houston, TX 77010

Telephone: (713) 654-5300

Facsimile: (713)-652-0109

Joseph R. Dorris

Georgia Bar Number 586708

dorris@fr.com

Fish & Richardson P.C.

1180 Peachtree St. NE, Fl. 21

Atlanta, GA 30309

Telephone: (404) 892-5005

Facsimile: (404) 892-5002

Cheryl Wang

California Bar Number 323305

cwang@fr.com

Fish & Richardson P.C.

12860 El Camino Real, Suite 400

San Diego, CA 92130

Telephone: (858) 678-5070

Claire Chang

California Bar Number 341420

cchang@fr.com

Fish & Richardson P.C.

500 Arguello Street, Suite 400

Redwood City, CA 94063

Telephone: (650) 839-5070

Riley Green

Texas Bar Number 24131352

rgreen@fr.com

Fish & Richardson P.C.

1717 Main Street, Suite 5000

Dallas, Texas 75201

Telephone: (214) 747-5070

Deron R. Dacus
Texas Bar No. 00790553
ddacus@dacusfirm.com
The Dacus Firm
821 ESE Loop 323, Suite 430
Tyler, Texas 75701
Telephone: (903) 705-1117

*Counsel for Defendant
Microsoft Corporation*

CERTIFICATE OF SERVICE

I certify that a copy of the foregoing document was served via email on December 5, 2025, upon all counsel of record.

/s/ Benjamin C. Elacqua

Benjamin C. Elacqua