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

QOMPLX LLC, *Plaintiff,* v. MICROSOFT CORPORATION, *Defendant.*	Case No. 1:25-cv-01383-ADA

MICROSOFT CORPORATION'S OPENING CLAIM CONSTRUCTION BRIEF

MICROSOFT EXHIBIT 1049
Microsoft v. Qomplx
IPR2026-00183

TABLE OF CONTENTS

I. INTRODUCTION 1

II. LEGAL STANDARDS 1

III. OVERVIEW OF THE PATENTS 1

A. The ’663 Patent. 1

B. The ’424 Patent. 3

C. The ’934 and ’426 Patents. 4

IV. THE DISPUTED CLAIM TERMS 5

A. “midserver” (’663 patent, claims 1 and 3-4) 5

1. The Court Should Not Undo Qomplx’s Claim Amendments. 6

2. Qomplx Cannot Show the Requirements for Lexicography. 8

3. The Specification Does Not Warrant Undoing Qomplx’s Claim Amendments. 10

B. “virtual appliance software application” (’663 patent, claims 1 and 4) 12

1. The Plain Meaning of “Virtual Appliance.” 12

2. Qomplx’s Construction Is Fatally Flawed. 14

C. “distributed computational graph” (’424 patent, claims 1, 4, and 6) 18

D. “transformation pipeline” (’424 patent, claims 1-6 and 9-12) 22

E. “transformation” (’424 patent, claims 1-6 and 9-12) 25

F. “retrieving, from the multidimensional time-series database, historical information about previous access requests associated with the [first] user account, and determining, based at least on the historical information” (’934 patent, claims 1, 11, 18, and 26; ’426 patent, claims 1, 9, 19, and 29) 27

V. CONCLUSION 30

TABLE OF AUTHORITIES

Cases

Accent Packaging, Inc. v. Leggett & Platt, Inc.,
707 F.3d 1318 (Fed. Cir. 2013).....14

Apple Comp., Inc. v. Articulate Sys., Inc.,
234 F.3d 14 (Fed. Cir. 2000).....17

ArcelorMittal France v. AK Steel Corp.,
700 F.3d 1314 (Fed. Cir. 2012).....17

Blackbird Tech LLC v. ELB Elecs., Inc.,
895 F.3d 1374 (Fed. Cir. 2018).....6, 8, 10

CCS Fitness, Inc. v. Brunswick Corp.
288 F.3d 1359 (Fed. Cir. 2002).....8, 11

C.R. Bard, Inc. v. U.S. Surgical Corp.,
388 F.3d 858 (Fed. Cir. 2004).....12

DDR Holdings, LLC v. Priceline.com LLC,
122 F.4th 911 (Fed. Cir. 2024) 10

Ethicon Endo-Surgery, Inc. v. U.S. Surgical Corp.,
93 F.3d 1572 (Fed. Cir. 1996).....29

Ferring B.V. v. Serenity Pharms., LLC,
356 F. Supp. 3d 353 (S.D.N.Y. 2019).....9

Fontem Ventures, B.V. v. NJOY, Inc.,
No. 2:14-cv-01645, 2015 WL 12766460 (C.D. Cal. Jan. 29, 2015).....9

Helmsderfer v. Bobrick Washroom Equip., Inc.,
527 F.3d 1379 (Fed. Cir. 2008).....8, 29

Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.,
381 F.3d 1111 (Fed. Cir. 2004).....12

Interactive Gift Express, Inc. v. Compuserve Inc.,
256 F.3d 1323 (Fed. Cir. 2001).....28

Int'l Rectifier Corp. v. IXYS Corp.,
361 F.3d 1363 (Fed. Cir. 2004).....29

Laryngeal Mask Co. Ltd. v. Ambu A/S,
618 F.3d 1367 (Fed. Cir. 2010).....6, 8, 10

Liebel-Flarsheim Co. v. Medrad, Inc.,
358 F.3d 898 (Fed. Cir. 2004).....9

Medicines Co. v. Mylan, Inc.,
853 F.3d 1296 (Fed. Cir. 2017).....9

Medtronic, Inc. v. Edwards Lifesciences Corp.,
No. 11-cv-1650, 2013 WL 2147909 (D. Minn. May 16, 2013) 9, 11

Mformation Techs., Inc. v. Research in Motion Ltd.,
764 F.3d 1392 (Fed. Cir. 2014).....28

Microsoft Corp. v. Multi-Tech Sys., Inc.,
357 F.3d 1340 (Fed. Cir. 2004).....21

Nazomi Comms., Inc. v. ARM Holdings, PLC,
403 F.3d 1364 (Fed. Cir. 2005).....29

Nobel Biocare Servs. AG v. Instradent USA, Inc.,
903 F.3d 1365 (Fed. Cir. 2018).....14

Phillips v. AWH Corp.,
415 F.3d 1303 (Fed. Cir. 2005) (en banc).....10, 12, 19

Polaris PowerLED Techs., LLC v. Dell Techs., Inc.,
No. 1:22-cv-00973-RP, 2025 WL 2525457 (W.D. Tex. Mar. 10, 2025) 6, 10, 11

Rembrandt Data Techs., LP v. AOL, LLC,
641 F.3d 1331 (Fed. Cir. 2011).....29

Rheox, Inc. v. Entact, Inc.,
276 F.3d 1319 (Fed. Cir. 2002).....10, 11

TALtech Ltd. V. Esquel Apparel, Inc.,
279 F. App'x 974 (Fed. Cir. 2008) 28

Thorner v. Sony Comput. Ent. Am. LLC,
669 F.3d 1362 (Fed. Cir. 2012).....8, 9, 26

Transonic Sys. v. Non-Invasive Med. Techs. Corp.,
143 F. App'x 320 (Fed. Cir. 2005) 6, 8

U.S. Surgical Corp. v. Ethicon, Inc.,
103 F.3d 1554 (Fed. Cir. 1997).....12

United States v. Telectronics, Inc.,
857 F.2d 778 (Fed. Cir. 1988).....6, 8, 10

Vitronics Corp. v. Conceptronic, Inc.,
90 F.3d 1576 (Fed. Cir. 1996).....14, 16

XY LLC v. Trans Ova Genetics LC,
No. 1:13-cv-00876, 2014 WL 5032693 (D. Colo. Oct. 7, 2014).....10, 11

I. INTRODUCTION

This case involves six asserted patents, four of which relate to computer security and two of which relate to data processing and distributed computing. The claim terms at issue come from four of those six patents.

Microsoft seeks to construe three of the six disputed claim terms. Microsoft seeks to construe claim terms where the Patent Office confirmed that the scope of the term is unclear without explanation ("distributed computational graph"), where the term has a special meaning ("transformation pipeline"), or where the patentee acted as its own lexicographer for the term ("transformation"). Each of Microsoft's proposed constructions relies squarely on the intrinsic evidence and faithfully applies claim construction principles.

Qomplx, in contrast, seeks to alter claim scope with litigation-driven constructions that violate well-established principles of claim construction. For one term, Qomplx asks the Court to read limitations back into a claim that Qomplx deleted during patent prosecution. For another term, Qomplx proposes a meaning that excludes the only disclosed embodiment that describes the claim term. For the last term, Qomplx imposes an order of operations unjustified by logic, grammar, or intrinsic evidence.

II. LEGAL STANDARDS

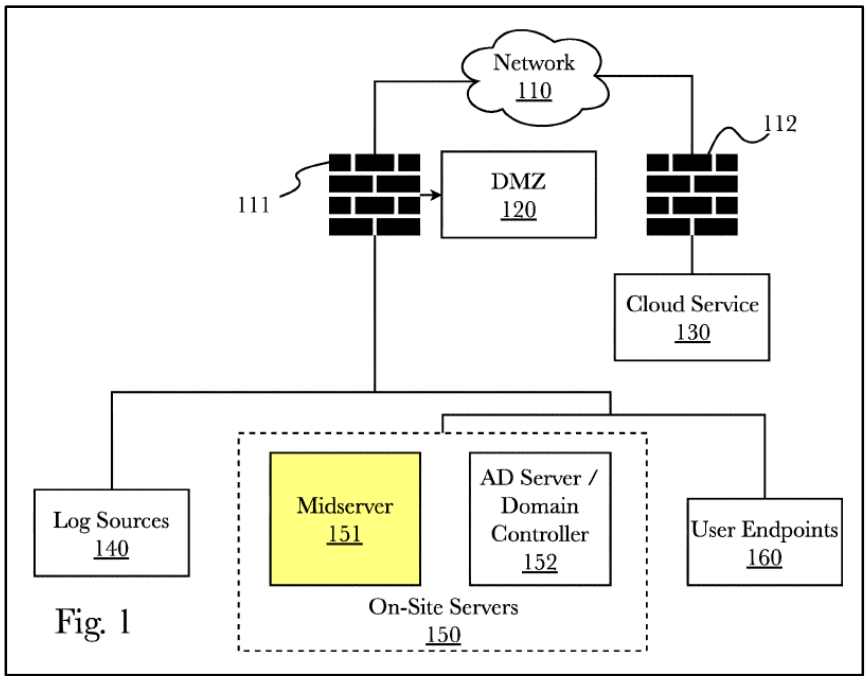
The Court is well versed in the law governing claim construction. Thus, Microsoft does not repeat those principles here and cites the authority most relevant to the claim terms at issue in the analysis below.

III. OVERVIEW OF THE PATENTS

A. The '663 Patent.

U.S. Patent No. 11,539,663 relates to "computer architectures for enterprise data collection, analysis, and transmission to cloud-based services." Ex. A at 7:5-8. It describes that

"[l]arge business enterprises" can have "thousands of computers and peripherals" that transfer data to a "cloud-based service." *Id*. at 7:12-21. Such "heterogenous data transfer" "presents numerous problems" related to reliability, security, and scalability. *Id*. at 7:22-42. To purportedly address these problems, the patent proposes the use of "midservers" that act as an intermediary "between a multitude of computing devices and peripherals" and "a cloud-based service." *Id*. at 7:60-66. The midservers collect data from the business's computers, process it, and transmit it to the cloud-based service. *Id*. at 7:60-66, 9:28-34. Figure 1 shows a system with midserver 151:



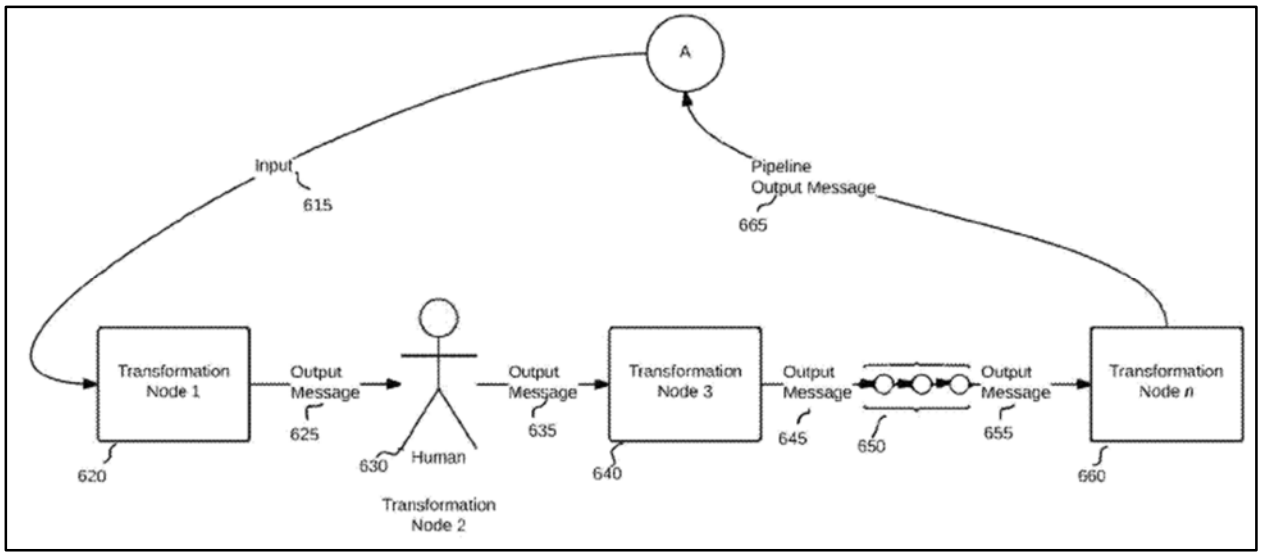
Id. at Fig. 1 (highlighting added). The midserver collates data" that it collects from "other servers 150," "endpoints 160," and "log sources 140" "for transfer to a cloud server 130." *Id*. at 13:26-43. The midserver may also analyze, filter, or transform the collected data before transmitting it to the cloud service. *Id*. at 13:18-26.

A person of ordinary skill in the art of the '663 patent would have had a bachelor's degree in computer science, computer engineering, electrical engineering, or an equivalent degree, with

two to four years of experience designing, implementing, or otherwise working with cloud computing systems. More education would compensate for less work experience and vice versa.

B. The '424 Patent.

U.S. Patent No. 12,143,424 relates to the "analysis of very large data sets using distributed computational graph tools which allow for transformation of data through both linear and non-linear transformation pipelines." Ex. B at 3:32-35. The patent states that a "graph" in the context of the patent is "a representation of information and relationships" made up of nodes and edges between the nodes. *Id*. at 9:5-9. It also explains that a directed graph is one in which the edges have a direction, from one node to the other. *Id*. at 9:23-31. It then explains that a "distributed computational graph system" includes "transformation pipeline[s]," where "each transformation compris[es] a node and the output messages between transformations compris[e] edges" in the graph. *Id*. at 9:31-35. The patent's Figure 6 shows "distributed computational graph 600" that includes a "transformation pipeline" with "transformation nodes" 1 through *n*:



Ex. B at Fig. 6. Transformation node 1 receives "input data stream" 615, applies a "function," and sends the transformed data to transformation node 2 as "output message 625." *Id*. at 17:14-19. Each transformation node does the same until the end of the transformation pipeline. *Id*. at

17:19-24. Figures 7 through 9 show variations of this architecture. *Id*. at 17:52-19:67.

A person of ordinary skill in the art of the ’424 patent would have had a bachelor’s degree in computer science, computer engineering, electrical engineering, or an equivalent degree, with two to four years of experience designing, implementing, or otherwise working with distributed computing systems. More education would compensate for less work experience and vice versa.

C. The ’934 and ’426 Patents.

U.S. Patent Nos. 12,218,934 and 12,231,426 both are continuations of the same parent application and share a common specification. Ex. C at cover page; Ex. D at cover page. Microsoft will cite to the ’934 patent specification for simplicity. The patents relate to “network security, particularly to multi-factor user authentication.” Ex. C at 2:15-16. “Multi-factor authentication” uses “an additional verification step often used in conjunction with a traditional login and password.” *Id*. at 2:20-23. An example of multi-factor authentication is a “one-time use code[] sent to a user’s mobile device or email.” *Id*. at 2:23-27. But such methods have faults and can be compromised. *Id*. at 2:27-34. The patents propose a system that purportedly “uses a combination of verification methods so that over-reliance on a single, and possibly compromised, method is eliminated.” *Id*. at 2:35-37. The system can purportedly “determine the varying amounts of verification needed, based on the context and risks associated with the connection.” *Id*. at 2:36-40.

In an embodiment, the system determines a “verification score that must be obtained by a user before the user may access requested resources.” Ex. C at 2:47-50. The score may be based on “such metrics as origin of the user’s connection, whether the access request is determined to be anomalous by the server based on a predetermined network baseline, accessing files or drives with a higher level security assignment, and the like.” *Id*. at 12:26-31. A user can “collect verification points via a plurality of verification methods to gain access.” *Id*. at 2:53-55.

The patent's Figure 6 shows an embodiment that allows for a user login, a determination of whether additional verification is needed, and the successful completion of the additional verification before granting access. Ex. C at Fig. 6.

A person of ordinary skill in the art of the '934 and '426 patents would have had a bachelor's degree in computer science, or equivalent, with two to four years of experience designing, implementing, or otherwise working with cybersecurity technologies. More education would compensate for less work experience and vice versa.

IV. THE DISPUTED CLAIM TERMS

A. "midserver" ('663 patent, claims 1 and 3-4)

Microsoft's Proposed Construction	Qomplx's Proposed Construction
Plain and ordinary meaning.	A server that functions as an interface between an external network and a cloud-based service, and which runs one or more containerized services that perform one or more of: collecting, aggregating, analyzing, filtering, transforming, and securely transmitting data.

The parties dispute whether this claim term requires construction (it does not) and, if it does, whether Qomplx can show that it clearly intended to redefine this claim term from its plain meaning in light of contradictory amendments made during prosecution (it cannot).

The Court does not need to construe this claim term because "mid" is a commonly used prefix that means middle or intermediate. "Midcentury," "midstream," "midair," "midnight," and "midrange" are a few examples of how this prefix is commonly used. Here, the prefix refers to a server located between (1) the computers on a local network from which it collects data and (2) the cloud service on an external network to which it sends data. *See* § III.A (overview of the '663 patent). The '663 patent's claim 1 reflects this understanding by reciting that the midserver "receive[s] data over a local network from a plurality of computing devices" and then

"retransmit[s] the received data over the secure connection," which is "a secure network connection to an external network." Ex. A at 24:32-38.

Furthermore, Qomplx's construction is not viable for the reasons set forth below.

1. The Court Should Not Undo Qomplx's Claim Amendments.

"Courts are not permitted to read back into the claims limitations which were originally there and were removed during prosecution of the application through the Patent Office." *United States v. Telectronics, Inc.*, 857 F.2d 778, 783 (Fed. Cir. 1988) (citation omitted); *see also Blackbird Tech LLC v. ELB Elecs., Inc.*, 895 F.3d 1374, 1379 (Fed. Cir. 2018) (claim amendment deleting claim language "undisputedly and unequivocally deleted [that requirement] from the claim"); *Laryngeal Mask Co. Ltd. v. Ambu A/S*, 618 F.3d 1367, 1373 (Fed. Cir. 2010) ("[I]nsistence upon this court's reading back into the claims limitations which were originally there and were removed during prosecution of the application through the Patent Office cannot be permitted.") (citation omitted); *Transonic Sys. v. Non-Invasive Med. Techs. Corp.*, 143 F. App'x 320, 326 (Fed. Cir. 2005) ("this court's case law precludes a reading that restricts [the claim term] to the limitations removed by broadening amendment"); *Polaris PowerLED Techs., LLC v. Dell Techs., Inc.*, No. 1:22-cv-00973-RP, 2025 WL 2525457, at *8 (W.D. Tex. Mar. 10, 2025) (rejecting construction that included term "mathematical product" in part because the patentee "amended its claims during prosecution to remove the term 'mathematical product'").

During prosecution of the '663 patent, Qomplx amended claim 1 to delete claim language that Qomplx now includes in its proposed construction, as shown below.

1. (Currently amended) A system for ingestion of data into a cloud-based service from an external network, comprising:

a midserver ~~configured to function as an interface between an external network and a cloud-based service~~ comprising at least a processor, a memory, and a plurality of programming instructions stored in the memory and operating on the processor, wherein the plurality of programming instructions, when operating on the processor, cause the processor to:

automatically install a virtual appliance software application, the virtual appliance software application configured to automatically load a plurality of stored configurations on the midserver;

establish a secure network connection to an external network;

receive data over a local network from a plurality of computing devices ~~on the external network~~;

apply a plurality of transformations to at least a portion of the received data; and

retransmit the received data over the secure connection as a single data stream.

~~run one or more containerized services to process the received data; and~~

~~securely transmit the processed data to a cloud-based service.~~

Ex. E at 2 (highlighting added). In this amendment, Qomplx deleted at least three claim limitations it now seeks to add back into the claim.

First, Qomplx deleted the requirement that the midserver be "configured to function as an interface between an external network and a cloud-based service." *Id.* Qomplx now seeks to write this requirement back into the claim via its construction that limits the midserver to a server "that functions as an interface between an external network and a cloud-based service."

Second, Qomplx deleted the requirement that the midserver (via programming instructions operating on its processor) "run[s] one or more containerized services to process the received data." *Id.* Qomplx now seeks to write this requirement back into the claim via its construction's requirement that the midserver "runs one or more containerized services that

perform one or more of: collecting, aggregating, analyzing, filtering, transforming, and securely transmitting data." Although Qomplx's construction refers to different types of data processing ("collecting," "aggregating," "analyzing," "filtering," and "transmitting" data), each is encompassed by the deleted claim language reciting "process[ing] the received data."

Third, Qomplx deleted the requirement that the midserver "securely transmit the processed data." *Id*. It now seeks to write this requirement back into the claim via its construction's requirement of "securely transmitting data."

In Qomplx's remarks accompanying these amendments, it made clear that it amended claim 1 "to more clearly teach the functions of the midserver" and that "[t]he amendments now clearly teach the operation of the midserver." Ex. E at 6. In sum, everything in Qomplx's construction after "[a] server" would impermissibly read limitations back into the claim that were removed during prosecution. *Telectronics, Inc.*, 857 F.2d at 783; *Blackbird Tech.*, 895 F.3d at 1379; *Laryngeal Mask*, 618 F.3d at 1373; *Transonic*, 143 F. App'x at 326. The Court therefore should reject Qomplx's construction.

2. Qomplx Cannot Show the Requirements for Lexicography.

Qomplx may argue that its construction is supported by supposed lexicography in the specification at 12:66-13:4, but this argument would fail. Courts apply an "exacting" standard for lexicography. *Thorner v. Sony Comput. Ent. Am. LLC*, 669 F.3d 1362, 1366 (Fed. Cir. 2012). "[A] patentee must 'clearly set forth a definition of the disputed claim term' other than its plain and ordinary meaning." *Id*. at 1365 (citing *CCS Fitness, Inc. v. Brunswick Corp*., 288 F.3d 1359, 1366 (Fed. Cir. 2002)). As a separate requirement, "the patentee must 'clearly express an *intent*' to redefine the term." *Id*. (quoting *Helmsderfer v. Bobrick Washroom Equip., Inc*., 527 F.3d 1379, 1381 (Fed. Cir. 2008)) (emphasis added); *see also Laryngeal Mask*, 618 F.3d at 1372 (no lexicography where the specification did "not clearly convey the patentee's intent to appoint a

special meaning to the term"); *Medicines Co. v. Mylan, Inc.*, 853 F.3d 1296, 1306 (Fed. Cir. 2017) (no lexicography where statement "lacks the clear expression of intent").

Qomplx cannot satisfy this intent requirement because the prosecution amendments described above preclude a finding of clear intent to redefine "midserver" in these claims. An amendment made during prosecution that removes claim language indicates an intent to broaden the claim such that the removed language no longer limits the claim. *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 909 (Fed. Cir. 2004) (the patentee's removal of "pressure jackets" claim language during prosecution provided a "strong indication that [it] intended those claims to reach injectors that did not use pressure jackets"); *see also Ferring B.V. v. Serenity Pharms., LLC*, 356 F. Supp. 3d 353, 364-65 (S.D.N.Y. 2019) (patentee "broadening his claims during patent prosecution suggests an intent to claim more, not less than was originally contemplated"); *Fontem Ventures, B.V. v. NJOY, Inc.*, No. 2:14-cv-01645, 2015 WL 12766460, at *16 n.21 (C.D. Cal. Jan. 29, 2015) ("By removing references to a bottle, the prosecution history shows an intent to broaden the claims."); *Medtronic, Inc. v. Edwards Lifesciences Corp.*, No. 11-cv-1650, 2013 WL 2147909, at *16-17 (D. Minn. May 16, 2013) (amendment changing "collecting means" to "holding surface" "evinces a clear intent to broadly claim a 'holding surface' rather than a 'collecting means'").

Qomplx's prosecution amendment to claim 1 shows an intent that "midserver" not be limited by the deleted claim language. Because the deleted claim language tracks the specification passage at 12:66-13:4, Qomplx cannot be found to have "clearly express[ed]" an intent for that passage to redefine the term "midserver" in claim 1. *Thorner*, 669 F.3d at 1366. The specification passage therefore does not satisfy the standard for lexicography and does not redefine "midserver."

3. The Specification Does Not Warrant Undoing Qomplx's Claim Amendments.

Even if Qomplx met the requirements for lexicography (it does not), the specification does not warrant undoing Qomplx's claim amendments. The case law that prohibits courts from adding back into the claims limitations that were deleted during prosecution makes no exception for lexicography. *Telectronics*, 857 F.2d at 783; *Laryngeal Mask*, 618 F.3d at 1373; *Blackbird Tech*, 895 F.3d at 1379; *Polaris*, 2025 WL 2525457, at *8.

This result makes sense because "[w]hen construing claims, [a] court looks to how a skilled artisan would read the claim term 'in the context of the entire patent,' including the specification and prosecution history." *DDR Holdings, LLC v. Priceline.com LLC*, 122 F.4th 911, 916 (Fed. Cir. 2024) (citing *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc)). Courts also recognize that "an *evolution* of the applicant's intended meaning of [a] claim term" can occur after the applicant first defines the term. *Id*. (emphasis added); *see also Rheox, Inc. v. Entact, Inc*., 276 F.3d 1319, 1327 (Fed. Cir. 2002) (noting the chronology that "the prosecution history ... was generated after the written description was drafted" in rejecting a specification-based argument in light of a prosecution amendment). Thus, courts have rejected allowing a specification to undo the effect of a later claim amendment, even where the specification sets forth an express definition.

For example, in *XY LLC v. Trans Ova Genetics, LC*, the court found that "the specification explicitly sets forth a definition of the term 'elliptical-like' as including shapes that are not true mathematical ellipses." No. 1:13-cv-00876, 2014 WL 5032693, at *7 (D. Colo. Oct. 7, 2014). The specification expressly stated that such shapes include those that are "close to a rectangle-shape:"

> This elliptical-like, single torsional interior surface may include different shapes in its cross sections. For example, besides being

> ellipse-shaped, it may be oval-shaped, or even close to a rectangle-shape. ... As should be understood, each of these shapes is intended to be encompassed by the term 'elliptical-like' even though a true mathematical ellipse is not present at a given cross section.

Id. at *6 (quoting U.S. Patent No. 6,263,745 at 7:64-8:8). During prosecution, however, the patentee amended the claim to recite "an ellipse-shape or an oval-shape." *Id*. The court did not adopt the specification's definition because the prosecution history "belie[d] any 'clearly set forth ... definition of the disputed claim term' that would include 'close to rectangle-shape.'" *Id*. at *7 (quoting *CCS Fitness*, 288 F.3d at 1366). Thus, the claim amendment precluded adoption of the specification's definition.

Similarly, in *Medtronic*, the specification explained that "[t]he *key element of the present invention* resides in the combination of the valve holder and the attaching threads which permit the threads to be *collected* by the valve holder ...." 2013 WL 2147909, at *15 (emphasis added). During prosecution of the patent's reissue application, however, the patentee broadened the claim language that recited a "coaxial thread *collecting* means" to recite a "coaxial thread *holding* surface." *Id*. at *16 (emphasis added). The court honored the broadening effect of the claim amendment and rejected the argument that the specification required that "the surface must 'collect' threads rather than simply 'hold' threads." *Id*. at *15-17.

Finally, in *Rheox*, the court considered whether the claim term "calcium orthophosphate" included triple superphosphate ("TSP"). 276 F.3d at 1324-25. The patentee argued that the patent "defined" TSP as "calcium orthophosphate." *Id*. at 1327. The court acknowledged that "this argument might be effective" if "[r]eading the written description alone." *Id*. During prosecution, however, the patentee deleted the claim's reference to TSP. *Id*. at 1326-27. The court therefore rejected the patentee's definitional argument. For these additional reasons, the Court should reject Qomplx's construction.

B. "virtual appliance software application" ('663 patent, claims 1 and 4)

Microsoft's Proposed Construction	Qomplx's Proposed Construction
Plain and ordinary meaning.	A cloud-based service delivered as a software stack that utilizes one or more virtual systems.

The '663 patent's claim 1 recites that the "midserver" includes programming instructions that "automatically install a virtual appliance software application, the virtual appliance software application configured to automatically load a plurality of stored configurations on the midserver." Ex. A at 24:28-31. Claim 4 recites a similar limitation. *Id*. at 24:52-55. The parties dispute whether this claim term requires construction (it does not), and whether Qomplx's construction is correct (it is not).

The limitation "virtual appliance software application" does not require construction because a person of ordinary skill in the art can readily understand it as a software application of a virtual appliance. *See U.S. Surgical Corp. v. Ethicon, Inc*., 103 F.3d 1554, 1568 (Fed. Cir. 1997) (claim construction "is not an obligatory exercise in redundancy").

1. The Plain Meaning of "Virtual Appliance."

If the Court wishes to construe "virtual appliance" by itself, Microsoft sets forth its plain meaning here. The '663 patent specification uses the term "virtual appliance" in only one place and assumes that the reader is familiar with the term. Ex. A at 16:16-21. The Court therefore may appropriately refer to extrinsic evidence to determine this term's plain meaning. *Phillips*, 415 F.3d at 1317 (extrinsic evidence can "shed useful light on the relevant art") (quoting *C.R. Bard, Inc. v. U.S. Surgical Corp.*, 388 F.3d 858, 862 (Fed. Cir. 2004)); *see also id*. at 1314 ("the court looks to 'those sources available to the public that show what a person of skill in the art would have understood disputed claim language to mean'") (quoting *Innova/Pure Water, Inc. v. Safari Water Filtration Sys., Inc.*, 381 F.3d 1111, 1116 (Fed. Cir. 2004)).

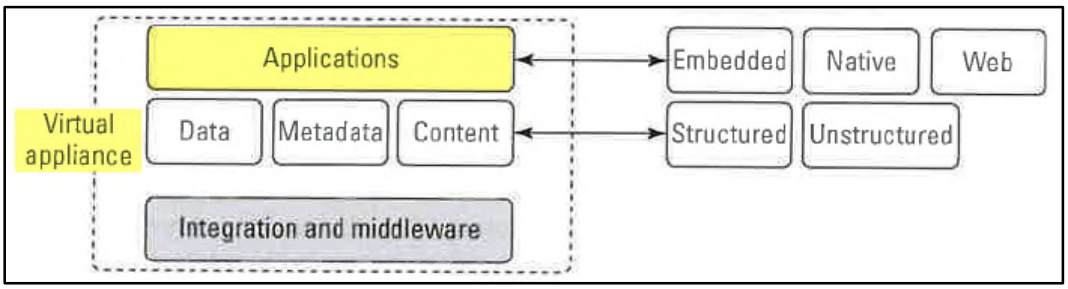
The extrinsic evidence shows that a virtual appliance was known in the art as a virtual machine image containing a pre-installed operating system and one or more pre-installed software applications, such that the software applications run when loaded in a virtual machine. For example, U.S. Patent Application Publication No. 2012/0246639 ("Kashyap"), filed in 2011, describes virtual appliances in its "Background" section as follows:

> **[0003]** A virtual appliance is a virtual machine image file that includes a pre-configured operating system and a single application. It is packaged as a pre-installed system image for a particular virtualized environment and hardware platform. A common file format for the virtual appliance is the open virtualization format (OVF). In one embodiment, an OVF package may include multiple virtual appliances, with an appliance have one or more applications. Deploying a new virtual appliance requires copying the appliance image, including the operating system and application, to a pre-installed and pre-configured virtual machine. Accordingly, the virtual appliance is a software image containing an application designed to run inside a virtual machine.

Ex. F at ¶ 3 (highlighting added). This passage first states that a "virtual appliance is a virtual machine image file," consistent with the plain meaning set forth above. This passage also states that a virtual appliance "include a pre-configured operating system" and "is packaged as a pre-installed system image," which together confirm that it includes a pre-installed operating system, as the plain meaning above recites. This passage then says that "an appliance [can] have one or more applications," consistent with the plain meaning set forth above, which recites "one or more pre-installed software applications." This passage concludes that a virtual appliance's "application [is] designed to run inside a virtual machine," consistent with the plain meaning's recitation that "the software applications run when loaded in a virtual machine."

Other extrinsic evidence confirms this plain meaning. For example, the "Cloud

Computing Bible" by Barrie Sosinsky (2011) confirms that a virtual appliance can include one or more applications, as shown by its Figure 3.2, an excerpt of which is shown below.



Ex. G at 50 (highlighting added). Likewise, "Reinventing Virtual Appliances" by X. Jin et al. (2009) ("Jin") says, "Traditionally, a virtual appliance is defined as one minimalist virtual machine image with one or more specific applications and an optimized operating system," again consistent with the plain meaning above. Ex. H at 3-4. And "Cloud Computing: A Practical Approach," by Anthony T. Velte (2010) ("Velte") defines a "virtual appliance" as a "virtual machine image" that "is fully preinstalled and preconfigured," consistent with the plain meaning above. Ex. I at 318.

2. Qomplx's Construction Is Fatally Flawed.

Qomplx's construction is flawed in multiple ways. First, "[t]here is a strong presumption against a claim construction that excludes a disclosed embodiment." *Nobel Biocare Servs. AG v. Instradent USA, Inc.*, 903 F.3d 1365, 1381 (Fed. Cir. 2018) (citation omitted). "A claim interpretation that excludes a preferred embodiment from the scope of the claim is rarely, if ever, correct." *Accent Packaging, Inc. v. Leggett & Platt, Inc.*, 707 F.3d 1318, 1326 (Fed. Cir. 2013) (citation omitted); *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1583-84 (Fed. Cir. 1996) (such an interpretation would require "highly persuasive evidentiary support").

The patent's Figure 6, shown below, shows an embodiment that includes "the midserver side 610 and the cloud-based service side 620."

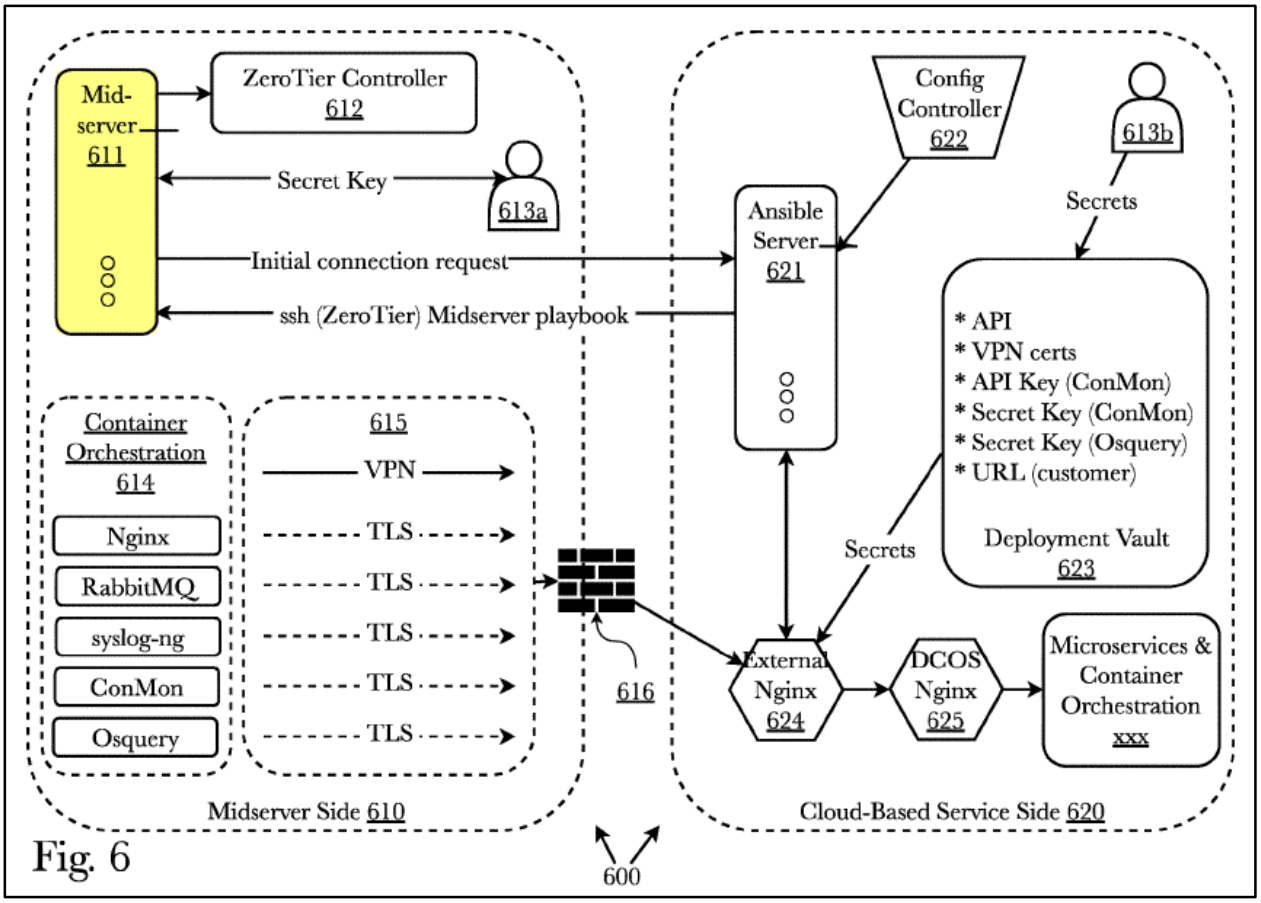


Fig. 6

Ex. A at 16:11-16 (highlighting added). In this embodiment, "[a]fter a midserver 611 is physically installed, an automated package called a midserver Open Virtual Appliance (OVA) *is installed and run on the midserver 611*." *Id*. at 16:16-19 (emphasis added). An "Open Virtual Appliance" (.ova) file is used with virtual appliances. *See* Ex. J at ¶1 (an "OVA (open virtual appliance) template file ... contains files for portability and deployment of virtualization appliances," which can include an "OVF (open virtual format) descriptor file"); Ex. K at 14 (for an "OVF package," "[t]he file extension is .ova (open virtual appliance or application)"). Thus, in this embodiment, the virtual appliance (and any virtual appliance software application) resides on the "midserver side 610" with the midserver 611, not on the "cloud-based service side 620." To be sure, the specification then discloses that "[t]he OVA will ... establish a ... connection to the cloud-based service side," again indicating that the virtual appliance (and any virtual

appliance software application) is on the midserver side 610. Ex A at 16:21-23. The specification then says that "the OVA will initiate a ZeroTier Controller 612," which Figure 6 illustrates as residing on the midserver side 610, again indicating that the virtual appliance (and any virtual appliance software application) is on the midserver side 610. *Id*. at 16:24-27.

Qomplx's construction, however, excludes this embodiment. Qomplx's construction requires a "virtual appliance software application" to be a "cloud-based service." Qomplx's construction therefore would require the virtual appliance (and its software applications) to reside on the cloud-based service side 620 of Figure 6. But in the embodiment of Figure 6, the virtual appliance (and its software applications) are deployed instead on the midserver side 610, as established above. *See also* Ex. A at 16:16-19 ("midserver Open Virtual Appliance (OVA) is installed and run on the midserver 611"). Qomplx's construction therefore excludes the embodiment of Figure 6. Worse, the embodiment of Figure 6 is the only embodiment for which the specification discloses a "virtual appliance." *Id*. at 16:11-60. Qomplx's construction therefore excludes the only embodiment that refers to a "virtual appliance." There is no "highly persuasive evidentiary support" that warrants doing so. *Vitronics Corp.*, 90 F.3d at 1583-84.

Furthermore, the distinction between the cloud-based service side 620 and the midserver side 610 is important in the context of the patent. A fundamental aspect of patent's purported invention is a midserver that is "located *between* the business enterprise computer infrastructure and the cloud-based infrastructure." Ex. A at 9:28-34 (emphasis added). Hence the term *mid*server. The midserver is positioned between the business's computers and the cloud-based services because it collects and processes data from "a multitude of computing devices" and then transmits the data "to a cloud-based service." *Id*. With this positioning, the midserver addresses the "security risk" caused by each of "thousands of computing devices sending data [directly] to

a cloud-based service on separate connections." *Id*. at 7:31-34, 11:28-32 ("Using the midserver as a *gateway* to the cloud-based service dramatically reduces the number of connections ..., greatly reducing the number of avenues of attack and improving network security.") (emphasis added). Qomplx's construction that obviates the distinction for the midserver's virtual appliance of Figure 6 therefore conflicts with a fundamental purpose of the patent. *See also ArcelorMittal France v. AK Steel Corp.*, 700 F.3d 1314, 1328 (Fed. Cir. 2012) (favoring construction "in light of the ... purpose of the invention" described in the specification); *Apple Comp., Inc. v. Articulate Sys., Inc.*, 234 F.3d 14, 25 (Fed. Cir. 2000) (same).

Qomplx's requirement of a "*cloud-based* service" also betrays the document from which it apparently obtained its construction. The "Open Virtualization Format Specification" (2015) defines "virtual appliance" as used in that document as "*a service* delivered as a software stack that utilizes one or more virtual systems," not "a *cloud-based* service ...." Ex. K at 12.

Second, the OVF specification definition on which Qomplx's construction apparently is based differs from the commonly accepted meaning of "virtual appliance." The OVF specification acknowledges as much when it introduces its section defining terms with the following caution:

> In this document, some terms have a specific meaning beyond the normal English meaning. Those terms are defined in this clause.

Ex. K at 10. In other words, the OVF specification defines terms where they have a special meaning that applies specially to that document and that differs from the "normal English meaning." Because the Court must give "virtual appliance" its plain meaning, the OVF specification definition does not apply.

That caution is well-deserved for the OVF specification's definition of "virtual appliance" because it materially differs from how that term was used in the art. For example, a

fundamental part of a "virtual appliance" is that it is a virtual machine image, as acknowledged by multiple definitions from the art:

1. Ex. F (Kashyap) at ¶ 3 ("A virtual appliance is a virtual machine image file ...");
2. Ex. L (U.S. Patent App. Pub. No. 2012/0216181 to Arcese) at ¶ 3 ("Virtual appliances are virtual machine images ...");
3. Ex. H (Jin) at 3-4 ("Traditionally, a virtual appliance is defined as one minimalist virtual machine image ...").
4. Ex. I (Velte) at 318 (defining "virtual appliance" as "A minimalist virtual machine image ...").

Even the patent describes its one example of a virtual appliance as "a virtual image," meaning a virtual machine image. Ex. A at 16:19-21. But the OVF specification definition makes no mention of a virtual machine image, confirming that it deviates from the term's plain meaning. The Court therefore should reject Qomplx's construction.

C. "distributed computational graph" ('424 patent, claims 1, 4, and 6)

Microsoft's Proposed Construction	Qomplx's Proposed Construction
A directed graph with each transformation comprising a node and the output messages between transformations comprising edges.	Plain and ordinary meaning.

The '424 patent's claim 1 recites a plurality of computer systems that each store data representing "a respective portion of a distributed computational graph," where "the distributed computational graph describes a flow of output data of a first transformation pipeline to an input of a second transformation pipeline." Ex. B at 23:42-51 (claim 1), 24:38-40 (claim 4), 24:61-63 (claim 6). The parties dispute whether "distributed computational graph" requires construction (it does) and whether Microsoft's construction is faithful to the intrinsic evidence (it is).

A person of ordinary skill in the art would not understand "distributed computational graph" to have a plain meaning. The prosecution history of related U.S. Patent Application No.

15/141,752 confirms this fact, as the Patent Office found "computational graph" to be indefinite and "unclear:"

> 7. **Claims 1, 7-8 and 14** are rejected under 35 U.S.C. 112(b) or 35 U.S.C. 112 (pre-AIA), second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which the inventor or a joint inventor, or for pre-AIA the applicant regards as the invention.
>
> **Claim(s) 1 and 8** recite limitation "**computational graph**". It is unclear what scope "**computational graph**" refers to and in what technical meaning a "graph" is defined as "computational". For purpose of art analysis, "**computational graph**" is interpreted as "executable step".

Ex. M at 3.[1] In response, Qomplx did not contend that this term had a known meaning and instead pointed to the application's description of a "directed computational graph." Ex. N at 8-9. The Court should therefore construe this term.

The claim language recites what a "distributed computational graph" "describes" ("a flow of data of a first transformation pipeline to an input of a second transformation pipeline") but does not recite what it *is*. The specification answers that question. *Phillips*, 415 F.3d at 1315 (the specification is the "primary basis for construing the claims") (citation omitted). The specification includes a section titled "Definitions," which begins by defining a "graph" with reference to nodes and edges, in accordance with its plain meaning. *Id*. at 9:4-23. It then defines a "directed" graph as one where the edges have a direction, "from one node to another in one direction." *Id*. at 9:23-31.

The section then describes a "distributed computational graph" system:

> In the *distributed computational graph* system, transformations within transformation pipeline are represented as [a] directed graph with each transformation comprising a node and the output messages between transformations comprising edges.

[1] The '424 patent's priority chain cites the '752 application three times. Ex. B at 2-3.

Ex. B at 9:31-35 (emphasis added). The specification then moves on and does not elsewhere provide a description of "distributed computational graph." *See, e.g.*, *id*. at 9:35-50. This passage's reference to "transformation pipeline[s]" matches how "distributed computational graph" is used in the claims—as something that describes "a flow of output data of a first transformation pipeline to an input of a second transformation pipeline." *Id*. at 23:42-51 (claim 1). And this passage's description is consistent with the embodiments in the specification, as established below. Microsoft therefore adopts this passage's description as its construction.

Microsoft's construction is consistent with the specification's distributed computational graphs. For example, Figure 7 shows "transformation pipeline" 700 that would be described by a distributed computational graph. Ex. B at Fig. 7. Each "transformation node" corresponds to a "function" that is performed on a stream of input data and that outputs an "output message" to the next transformation node. *Id*. at 17:56-58 ("Transformation node [1]'s function is performed on input data stream and transformed output message 715 is sent to transformation node 2."). Consistent with Microsoft's construction, each transformation corresponds to a node: "Transformation Node 1" 710, "Transformation Node 2" 720, up to "Transformation Node *n*" 750. *Id*. at 17:52-18:32. And each edge between transformation nodes corresponds to an output message: "output message 715," 725, 735, and 745. *Id*. at 17:56-58. The same is true of Figure 8, which shows transformation nodes 1 through *n* and 3' and 4', along with output messages 815, 825, 835, 845, 850, and 870. *Id*. at Fig. 8, 18:37-39 ("Transformation node [1]'s function is performed on input data stream and transformed output message 815 is sent to transformation node 2 820."), 18:33-19:13. And the same is true of Figure 9, which shows transformation nodes 1 through *n* and output messages 915, 925, 935, 940, and 950. *Id*. at Fig. 9, 19:14-67. The same also is true for Figures 13 to 15, which show various transformation pipeline configurations: a

transformation node receiving input data from two sources (Figure 13), a transformation node sending output data to two destination nodes (Figure 14), and sending output from one transformation node to act as input to an "antecedent" transformation node (Figure 15). *Id.* at 21:42-22:51.

The prosecution history of the related '752 application further confirms Microsoft's construction. *See Microsoft Corp. v. Multi-Tech Sys., Inc.*, 357 F.3d 1340, 1349 (Fed. Cir. 2004) ("[A patentee's] statement made during prosecution of [a] patent is relevant to an understanding of the common disclosure in the sibling ... patents."). As described above, Qomplx responded to the Patent Office's finding that "computational graph" is indefinite by excerpting the application's description of a "directed computational graph:"

> decomposable transformer service 150 for branching or iterative transformations that are part of analysis. The *directed computational graph* 155 represents all data as directed graphs where the transformations are nodes and the result messages between transformations edges of the graph. These graphs which contain considerable intermediate transformation data are stored and further analyzed within graph stack module 145.

Ex. N at 8-9 (highlighting added). This description mirrors Microsoft's construction. Although this passage refers to a "*directed* computational graph" rather than a "*distributed* computational graph," its description also applies to the '424 patent given that the '752 application's graph figures also appear in the '424 patent. *Compare* Ex. O ('752 application) at Figs. 3, 5-7, *with* Ex. B ('424 patent) at Figs. 6, 13-15. Moreover, during prosecution of another patent application, Qomplx amended the specification to replace the term "directed computational graph" with "distributed computational graph," treating them as interchangeable. Ex. P at 8 ("[T]he 'distributed computational graph' was referred to several times as a 'directed computational graph.' The correct terminology is 'distributed computational graph' ...."). The prosecution

history of the related application therefore further confirms Microsoft's construction, which the Court should adopt.

D. "transformation pipeline" ('424 patent, claims 1-6 and 9-12)

Microsoft's Proposed Construction	Qomplx's Proposed Construction
A progression of functions which each perform a transformation on a data stream.	Plain and ordinary meaning.

The '424 patent claims recite the term "transformation pipeline" in various limitations, such as "a flow of output data of a first transformation pipeline to an input of a second transformation pipeline," "apply the first transformation pipeline to the first stream of input data," and "at least one of the first transformation pipeline and the second transformation pipeline is non-linear." Ex. B at 23:49-51, 23:58-59, 24:20-21. The parties dispute whether this claim term warrants construction (it does) and whether Microsoft's construction is correct (it is).

The Court should construe this claim term because it is a technical term with a special meaning in the context of the '424 patent. The prosecution history of a related patent confirms this fact, as Qomplx explained to the Patent Office a special meaning of "pipeline." Specifically, the '424 patent's priority chain cites U.S. Patent Application No. 16/709,598, which therefore is a related application. Ex. B at 2-3. During prosecution of that application, Qomplx responded to the Patent Office's rejection of various claims by distinguishing the prior art from "pipelines" as follows.

> "database creation, configuration, and opening" (Gowda, page 29). The teachings of Gowda pertain to a database mapping tool and improvements to the operation thereof, and Gowda is very clearly referring to the creation and manipulation of databases and file-based contents thereof. This is fundamentally different from the claimed limitations of creating *pipelines*, which are sequences of data transformations performed by the directed computational graph module of the invention (a detailed definition may be found in at least ¶[010] of the disclosure, and the usage of the term in the claims is consistent with the given definition). A sequence of operations is

Ex. Q at 8-9 (highlighting added). Qomplx's explanation of the special meaning of "pipelines" shows that the meaning of the more specific term "transformation pipeline" also warrants explanation.

The '424 patent's claim language quoted above recites that data can flow between transformation pipelines, that a transformation pipeline can be applied to a data stream, and that a transformation pipeline can be non-linear. Ex. B at 23:49-51, 23:58-59, 26:7-8. But the claims do not recite what a "transformation pipeline" *is*. The specification answers that question.

The specification section titled "Discussion of the State of the Art" describes a "data pipeline" as follows.

> Data pipelines, which are a progression of functions which each perform some action or transformation on a data stream, offer a mechanism to process quantities of data in the volume discussed directly above.

Ex. B at 4:31-34. Although this passage refers to "data pipelines" rather than "transformation pipelines," it remains instructive because, first, the recited "transformation pipelines" operate on data. *Id*. at 23:58-59 (a "stream of input data"). Second, the passage refers to the pipeline performing a progression of "some action or *transformation*" on a data stream, and logic dictates that when the pipeline performs a progression of transformations, it is a "transformation pipeline." Indeed, as shown above, when Qomplx explained the meaning of "pipeline" during

prosecution of the related ’598 application, it referred to “a data transformation pipeline” in the next sentence, indicating that “data [transformation] pipelines” and “[data] transformation pipelines” refer to the same concept. Microsoft’s construction therefore abides by the specification’s description of data transformation pipelines.

The specification further supports Microsoft’s construction. The specification’s “Definition” section explains:

> In the invention, transformations are placed into pipelines such that the output of one transformation may serve as an input for another.

Ex. B at 9:62-64. This series of transformations aligns with the construction’s recitation of a progression of functions that each perform a transformation. The next sentence says:

> These pipelines can consist of two or more transformations with the number of transformations limited only by the resources of the system.

Id. at 9:64-67. This sentence further confirms that a transformation pipeline is a *progression* of functions that each perform a transformation, not just one such function. Microsoft’s construction does not limit the number of transformations that a transformation pipeline can include, consistent with this sentence. The specification then explains:

> Historically, transformation pipelines have been linear with each transformation in the pipeline receiving input from one antecedent and providing output to one subsequent with no branching or iteration. Other pipeline configurations are possible. The invention is designed to permit several of these configurations including, but not limited to: linear, afferent branch, efferent branch and cyclical.

Id. at 9:67-10:7. This description again aligns with Microsoft’s construction for the same reasons. Microsoft’s construction also allows a transformation pipeline to be linear, to branch, and to be cyclical.

Microsoft’s construction also is consistent with the rest of the specification. For example,

Figure 7 shows "transformation pipeline 700" with a progression of functions represented by "Transformation Node[s]" 1 through *n*. Ex. B at 17:52-58, Fig. 7. Each transformation node function receives an input data stream, performs a transformation, and sends the output data to the next transformation node. *Id*. at 17:56-58 ("Transformation node's function is performed on input data stream and transformed output message 715 is sent to transformation node 2 720."). The same is true of Figure 8's "transformation pipeline [800]" and Figure 9's "transformation pipeline [900]." *Id*. at 18:37-39, 19:18-21, 18:33-19:67. Figures 13-15 likewise each describe a "transformation pipeline," each of which is a progression of three or four functions represented as transformation nodes, each of which applies a transformation to the input data and sends output data. *Id*. at 21:42-22:51. Figures 13 and 14 show branching pipelines and Figure 15 shows a cyclical pipeline. *Id*. at 21:42-46, 22:1-6, 22:28-39. The Court therefore should adopt Microsoft's construction.

E. "transformation" ('424 patent, claims 1-6 and 9-12)

Microsoft's Proposed Construction	Qomplx's Proposed Construction
A function performed on zero or more streams of input data which results in a single stream of output.	Plain and ordinary meaning.

The '424 patent claims recite the term "transformation" as part of a "first transformation pipeline" and a "second transformation pipeline." Ex. B at *e.g.*, 23:49-51, 23:58-59, 26:7-8. The parties dispute whether this claim term warrants construction (it does) and whether the Qomplx's lexicography for this term is the proper construction (it is).

The Court should construe this claim term because, as established below, Qomplx has acted as its own lexicographer for this claim term by clearly setting forth a definition in the specification that redefines the term from its plain and ordinary meaning. That definition, which Microsoft adopts as its construction, deviates from the term's plain meaning: 1) by allowing the

transformation to operate on "zero" streams of input data; and 2) by requiring that the transformation "results in a single stream of output."

As described above, courts do not give claim terms their plain and ordinary meaning "when a patentee sets out a definition and acts as his own lexicographer." *Thorner*, 669 F.3d at 1365. Lexicography requires that the patentee "clearly set forth a definition of the disputed claim term other than its plain and ordinary meaning" and "clearly express an intent to redefine the term." *Id.* (citations omitted). The '424 patent's "Definitions" section expressly defines "transformation" as follows:

> As used herein, "transformation" is a function performed on zero or more streams of input data which results in a single stream of output which may or may not then be used as input for another transformation. Transformations may

Ex. B at 9:4, 9:51-54. This passage clearly sets forth a definition intended to redefine the term "transformation" for purposes of this patent. No events in the patent's prosecution history lead to a different result. The next sentence says that "[t]ransformations may comprise any combination of machine, human or machine-human interactions," which Microsoft's construction (the specification definition) allows. *Id.* at 9:54-56. The following sentence says that "[t]ransformations need not change data that enters them," because a "transformation" could, for example, store the data. *Id.* at 9:56-60. Microsoft's again allows for such a transformation.

Microsoft's construction also is consistent with the rest of the specification. For example, Figure 7 shows "Transformation Node[s]" 1 through *n*, each of which performs a function on a stream of input data and sends a single stream of output (labeled as an "output message"). Ex. B at 17:56-58 ("Transformation node [1]'s function is performed on input data stream and transformed output message 715 is sent to transformation node 2."), 17:52-18:32. Likewise, in Figure 8, each "Transformation Node" performs a function on a stream of input data and sends a

single stream of output (again labeled as an "output message"). *Id*. at 18:37-39 ("Transformation node [1]'s function is performed on input data stream and transformed output message 815 is sent to transformation node 2 820."), 18:33-19:13. Although "transformation node 2 820 sends its output stream to two transformation pipelines," each is "*the same data stream*" and therefore the transformation still results in a "single stream of output." *Id*. at Fig. 8 (elements 825 and 850), 18:37-46 (emphasis added). Similarly, each "Transformation Node" in Figure 9 performs a function on a stream of input data and sends a single stream of output (again labeled as an "output message"). *Id*. at 19:18-21 ("Transformation node's function may be performed on an input data stream and transformed output message 915 may then be sent to transformation node 2 920."), 19:14-67. The same is true for Figures 13 to 15, which show various transformation pipeline configurations. *Id*. at 21:42-22:51. In Figure 14, transformation node 1403 "sends [an] output data stream" (singular) "to two destination transformation nodes." *Id*. at 22:1-6. Because the same output data stream is sent each time, there still is a "single stream of output," as with Figure 8. The Court therefore should adopt Microsoft's construction.

F. "retrieving, from the multidimensional time-series database, historical information about previous access requests associated with the [first] user account, and determining, based at least on the historical information" ('934 patent, claims 1, 11, 18, and 26; '426 patent, claims 1, 9, 19, and 29)

Microsoft's Proposed Construction	Qomplx's Proposed Construction
Plain and ordinary meaning.	The "determining" step takes place after the "retrieving" step. Both steps take place after receipt of "a request to authenticate a client."

Each of the '934 patent's independent claims recite the following limitations:

1. "receiv[ing] a request to authenticate a client ...;"
2. "retrieving, from the multidimensional time-series database, historical information about previous access requests associated with the user account;" and

3. "determining, based at least on the historical information, whether the user account is associated with ...."

Ex. C at 18:15-29, 19:21-35, 20:11-28, 21:3-19. The '426 patent's independent claims recite essentially identical limitations. Ex. D at 18:14-30, 19:8-24, 20:14-29, 21:6-20. The parties dispute whether this claim language requires construction (it does not) and whether the intrinsic evidence requires the order of operations of Qomplx's proposed construction (it does not).

"Unless the steps of a method [claim] actually recite an order, the steps are not ordinarily construed to require one." *Mformation Techs., Inc. v. Research in Motion Ltd.*, 764 F.3d 1392, 1398 (Fed. Cir. 2014) (quoting *Interactive Gift Express, Inc. v. Compuserve Inc.*, 256 F.3d 1323, 1342 (Fed. Cir. 2001)). A specific order may be appropriate where "'the claim language, as a matter of logic or grammar, requires that the steps be performed in the order written, or the specification directly or implicitly requires' an order of steps." *Id.* at 1398-99 (quoting *TALtech Ltd. v. Esquel Apparel, Inc.*, 279 F. App'x 974, 978 (Fed. Cir. 2008)).

Here, the claim language does not recite an order for these limitations and does not require one as a matter of logic or grammar. Qomplx first contends that the "determining ..." limitation must occur after the "retrieving ..." limitation. But the "determining ..." limitation does not recite "determining, based at least on the *retrieved* historical information ...." It instead recites "determining, based at least on the historical information ...." It therefore does not require first carrying out the "retrieving ..." limitation. Specifically, a given set of information can be obtained at different times (and perhaps from different sources) and still be the same each time. The "determining ..." limitation's reference to "the historical information" therefore requires only that it use the same information as the "retrieving ..." limitation, even if that information was obtained before the "retrieving ..." limitation was performed. In other words, the "determining ..." limitation is indifferent as to whether the system obtained the historical information it uses as

part of the "retrieving ..." limitation or at a different time.

Qomplx also contends that the "retrieving ..." and "determining ..." limitations happen after the "receiv[ing] ..." limitation. But the claim language does not require that order either. For example, the claim does not recite "*after* receiving the request to authenticate a client, retrieving ..." or "*after* receiving the request to authenticate a client, determining ...," as the claim drafter could have easily written. Although the "retrieving ..." limitation refers to "previous access requests," this language does not require that system has performed the "receiv[ing] ..." limitation because "previous" includes reference to access requests that came before the time at which the system performs the "retrieving." In that case, the "receiv[ing] ..." limitation may happen later.

Adopting Qomplx's construction therefore would improperly rewrite the claim language. *Rembrandt Data Techs., LP v. AOL, LLC*, 641 F.3d 1331, 1339 (Fed. Cir. 2011) ("courts may not redraft claims"); *see also Int'l Rectifier Corp. v. IXYS Corp.*, 361 F.3d 1363, 1372 (Fed. Cir. 2004) (a court may not "excuse the patentee from the consequences of its own word choice"). This is because "[t]he patentee chooses the language and accordingly the scope of his claims." *Helmsderfer*, 527 F.3d at 1383-84. The patent owner therefore "must live with the language it chose." *Ethicon Endo-Surgery, Inc. v. U.S. Surgical Corp.*, 93 F.3d 1572, 1583 (Fed. Cir. 1996).

Nor does the patent's common specification require the order of operations required by Qomplx's construction. For example, the specification includes no disclaimer or lexicography that would require such an order of operations. And the claims are not limited to the embodiments in the specification. *Nazomi Comms., Inc. v. ARM Holdings, PLC*, 403 F.3d 1364, 1369 (Fed. Cir. 2005) (claims may embrace "different subject matter than is illustrated in the specific embodiments in the specification"). Moreover, the specification emphasizes that readers

should not infer an order for its steps. As shown below, the specification says that: (1) steps may occur in "alternate orders, unless specifically stated to the contrary;" (2) steps may be performed in "any order practical;" and (3) even where steps are "implied as occurring sequentially," they may instead be performed "simultaneously:"

> illustrate one or more aspects. Similarly, although process steps, method steps, algorithms or the like may be described in a sequential order, such processes, methods and algorithms may generally be configured to work in alternate orders, unless specifically stated to the contrary. In other words, any sequence or order of steps that may be described in this patent application does not, in and of itself, indicate a requirement that the steps be performed in that order. The steps of described processes may be performed in any order practical. Further, some steps may be performed simultaneously despite being described or implied as occurring non-simultaneously (e.g., because one step is described after the other step). Moreover, the illustration of a process by its

Ex. C at 4:67-5:12 (highlighting added); Ex D at 4:67-5:12. The specification does not "specifically state" the order of Qomplx's proposed construction as contemplated by this passage. The specification then again says that functions may be performed "out of order," "concurrently," or "in reverse order" where possible:

> process. Alternate implementations are included within the scope of various aspects in which, for example, functions may be executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those having ordinary skill in the art.

Ex. C at 5:44-49 (highlighting added); Ex. D at 5:44-49 (same). The Court therefore should reject Qomplx's construction.

V. CONCLUSION

Microsoft respectfully requests that the Court adopt its proposed constructions.

Dated: March 23, 2026

Respectfully submitted,

/s/ John M. Guaragna

John M. Guaragna
Texas State Bar No. 24043308
john.guaragna@dlapiper.com
DLA Piper LLP (US)
303 Colorado Street, Suite 3000
Austin, TX 78701
Telephone: (512) 457-7000
Facsimile: (512) 457-7001

Sean Cunningham
sean.cunningham@dlapiper.com
Erin Gibson
erin.gibson@dlapiper.com
DLA Piper LLP (US)
500 Eighth Street, NW
Washington, DC 20004
Telephone: (202) 799-4000
Facsimile: (202) 799-5000

Robert Buergi
robert.buergi@dlapiper.com
Texas State Bar No. 24140916
DLA Piper LLP (US)
845 Texas Avenue, Suite 3800
Houston, TX 77022
Telephone: (713) 425-8440
Facsimile: (713) 300-6004

Attorneys for Defendant Microsoft Corporation

CERTIFICATE OF SERVICE

I hereby certify that on March 23, 2026, all counsel of record who are deemed to have consented to electronic service are being served with a copy of the foregoing document via the Court's CM/ECF system in accordance with Local Rule CV-5.

/s/ John M. Guaragna
John M. Guaragna