

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

**Valtrus Innovations, Ltd. and
Key Patent Innovations, Ltd.,**

Plaintiffs,

v.

SAP America, Inc. and SAP, SE,

Defendants.

§
§
§
§
§
§
§
§
§
§
§

CASE NO. 2:24-cv-00021-JRG

JURY TRIAL DEMANDED

**PLAINTIFFS VALTRUS INNOVATIONS LTD. AND
KEY PATENT INNOVATIONS LTD.'S
OPENING CLAIM CONSTRUCTION BRIEF**

TABLE OF CONTENTS

	<u>Page</u>
I. Introduction.....	1
II. Argument	2
A. SAP Cannot Meet Its Clear and Convincing Burden on Indefiniteness	2
B. Disputed Terms of U.S. Patent No. 7,936,738.....	3
1. Background for the '738 Patent	3
2. “context information” (claims 1, 2, 5, 6, 10, 33)	5
3. “potentially inaccurate or incomplete” (claim 10).....	6
C. Disputed Terms of U.S. Patent No. 6,889,244.....	7
1. Background for the '244 Patent	7
2. “the first and second nodes are each coupled to a fault tolerant storage system (FTSS)” (claim 1, preamble).....	8
3. “highly reliable fault-tolerant storage media” (claim 1, 2, 3, 6, 9, 10)	12
4. “accessing determining which nodes should receive the broadcast message based on the broadcast message” (claim 10)	13
D. Disputed Terms of U.S. Patent No. 7,313,575.....	14
1. Background of the '575 Patent	14
2. “polymorphic metadata” (claims 1, 4, 28, 33)	15
3. “assembler” and “disassembler” (claims 1, 4, 36).....	16
4. “the RTID that ... create at least one assembler defined according to the polymorphic metadata ...” (claim 28)	17
E. Disputed Terms of U.S. Patent No. 6,823,409.....	17
1. Background Concerning the '409 Patent	17
2. “coherency control module” terms (claims 1, 4)	21
3. “request module” terms (claims 1, 7)	23
4. “static RAM interface module” terms (claims 1, 7, 9)	25
F. Disputed Term of U.S. Patent No. 6,871,264	26
1. Background of the '264 Patent	26
2. “high-speed interconnect coupling” (claim 1)	27
III. Conclusion	29

TABLE OF AUTHORITIES

	Page(s)
Federal Cases	
<i>Aylus Networks, Inc. v. Apple Inc.</i> , 856 F.3d 1353 (Fed. Cir. 2017)	11
<i>BASF Corp. v. Johnson Matthey Inc.</i> , 875 F.3d 1360 (Fed. Cir. 2017)	2
<i>CBT Flint Partners, LLC v. Return Path, Inc.</i> , 654 F.3d 1353 (Fed. Cir. 2011)	17
<i>Chrimar Sys., Inc. v. Alcatel-Lucent USA, Inc.</i> , 2016 WL 1228767 (E.D. Tex. Mar. 28, 2016)	11
<i>DDR Holdings, LLC v. Hotels.com, L.P.</i> , 773 F.3d 1245 (Fed. Cir. 2014)	12
<i>Dyfan, LLC v. Target Corp.</i> , 28 F.4th 1360 (Fed. Cir. 2022)	2, 23, 24, 25
<i>Elcommerce.com, Inc. v. SAP AG</i> , 745 F.3d 490 (Fed. Cir. 2014), <i>vacated on other grounds</i> , 564 F. App'x 599 (Fed. Cir. 2014)	2
<i>Enzo Biochem, Inc. v. Gen-Probe, Inc.</i> , 424 F.3d 1276 (Fed. Cir. 2005)	2
<i>Eolas Techs. Inc. v. Amazon.com, Inc.</i> , 2016 WL 7155294 (E.D. Tex. Dec. 8, 2016)	7
<i>Finberg v. Sullivan</i> , 658 F.2d 93 (3d Cir. 1980)	2
<i>Hillman Group, Inc. v. KeyMe, LLC</i> , 2020 WL 3605626 (E.D. Tex. Jul. 2, 2020)	23, 25
<i>IPCom GmbH & Co. KG v. AT&T Corp.</i> , 2021 WL 4191407 (E.D. Tex. Sept. 15, 2021)	23
<i>Lolli v. County of Orange</i> , 351 F.3d 410 (9th Cir. 2003)	2
<i>Microchip Tech. Inc. v. Nuvoton Tech. Corp. Am.</i> , 2020 WL 978636 (N.D. Cal. Feb. 28, 2020)	23, 26
<i>Nautilus, Inc. v. Biosig Instruments, Inc.</i> , 572 U.S. 898 (2014)	17

<i>Niazi Licensing Corp. v. St. Jude Med. S.C., Inc.</i> , 30 F.4th 1339 (Fed. Cir. 2022).....	28
<i>Ollnova Techs. Ltd. v. ecobee Techs., ULC</i> , 2023 WL 2871051 (E.D. Tex. Apr. 10, 2023).....	14
<i>Pavo Sols. LLC v. Kingston Tech. Co., Inc.</i> , 35 F.4th 1367 (Fed. Cir. 2022).....	13
<i>PC Connector Sols. LLC v. SmartDisk Corp.</i> , 406 F.3d 1359 (Fed. Cir. 2005).....	11
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005).....	6
<i>SAS Inst. Inc. v. World Programming Ltd.</i> , 2020 WL 569856 (E.D. Tex. Feb. 5, 2020).....	23, 26
<i>SB IP Holdings, LLC v. Vivint Smart Home, Inc.</i> , 2021 WL 5804286 (E.D. Tex. Dec. 7, 2021)	7
<i>STA Group LLC v. Motorola Sol's, Inc.</i> , 2024 WL 3843731 (E.D. Tex. Jul. 10, 2024).....	28
<i>In re Taasera Licensing LLC, Pat. Litig.</i> , 2023 WL 8628323 (E.D. Tex. Dec. 13, 2023).....	11
<i>Team Worldwide Corp. v. Wal-Mart Stores, Inc.</i> , 2018 WL 1353116 (E.D. Tex. Mar. 15, 2018).....	12
<i>Tech. Props. Ltd. LLC v. Huawei Techs. Co., Ltd.</i> , 849 F.3d 1349 (Fed. Cir. 2017).....	11
<i>Wasica Fin. GmbH v. Cont'l Auto. Sys., Inc.</i> , 853 F.3d 1272 (Fed. Cir. 2017).....	2
<i>Williamson v. Citrix Online, LLC</i> , 792 F.3d 1339 (Fed. Cir. 2015).....	24, 25
<i>Zeroclick, LLC v. Apple Inc.</i> , 891 F.3d 1003 (Fed. Cir. 2018).....	26
Federal Statutes	
35 U.S.C. § 112.....	<i>passim</i>

TABLE OF EXHIBITS

Ex.	Description
1	U.S. Patent No. 6,823,409 (“the ’409 patent”).
2	Excerpts from U.S. Patent No. 6,823,409 File History.
3	Declaration of Michael C. Brogioli, Ph.D. in Support of Plaintiffs’ Proposed Claim Constructions (“Brogioli Decl.”) and Appendix A thereto.
4	John Hennessy et al., <i>Cache-Coherent Distributed Shared Memory: Perspectives on Its Development and Future Challenges</i> , 87 PROCEEDINGS OF THE IEEE 418 (Mar. 1999) (also Brogioli Decl., Ex. 11, (“Hennessy (1999)”).
5	A.K. Nanda et al., <i>High-throughput coherence control and hardware messaging in Everest</i> , 45 IBM J. RES. & DEV. 229 (Mar. 2, 2001) (also Brogioli Decl., Ex. 12, “Nanda (2001)”).
6	U.S. Patent No. 6,871,264 (“the ’264 patent”).
7	Excerpts from U.S. Patent No. 6,871,264 File History.
8	David A. Patterson and John K. Hennessy, <i>Computer Architecture: A Quantitative Approach</i> (2nd ed., 1996) (also Brogioli Decl., Ex. 5, “Hennessy & Patterson”).
9	Rudolf F. Graf, <i>Modern Dictionary of Electronics</i> (7th Ed. 1999) (excerpt 2) (also Brogioli Decl., Ex. 28).
10	Steven M. Kaplan, <i>Wiley Electrical and Electronics Engineering Dictionary</i> (2004) (excerpt 2) (also Brogioli Decl., Ex. 29).
11	U.S. Patent No. 6,681,296 to Liao et al. (also Brogioli Decl., Ex. 30).
12	U.S. Patent No. 6,604,174 to Dean et al. (also Brogioli Decl., Ex. 31).
13	U.S. Patent No. 5,809,537 to Itskin et al. (also Brogioli Decl., Ex. 32).
14	U.S. Patent No. 7,936,738 (“the ’738 patent”).
15	Declaration of Martin Farach-Colton, M.D., Ph.D. in Support of Plaintiffs’ Proposed Claim Constructions (“F.C. Decl.”).
16	U.S. Patent No. 6,889,244 (“the ’244 patent”).
17	U.S. Patent No. 6,938,071 (“the ’071 patent”).
18	Excerpts from the prosecution history of the ’244 patent (also F.C. Decl., Ex. 18).
19	U.S. Patent No. 7,313,575 (“the ’575 patent”).
20	Excerpts from Douglass Downing and Michael Covington, <i>Dictionary of Computer and Internet Terms, Seventh Ed.</i> (2000).
21	Thomas J. Sheffler, <i>A Portable MPI-Based Parallel Vector Template Library</i> (1995).
22	Excerpts from Simon St. Laurent and Robert Biggar, <i>Inside XML DTDs</i> (1999).

Pursuant to P.R. 4-5(a), Plaintiffs Valtrus Innovations, Ltd. and Key Patent Innovations, Ltd. (collectively “Valtrus”) respectfully submit this Opening Claim Construction Brief.

I. INTRODUCTION

SAP argues that all but one of the terms at issue are indefinite.¹ SAP seeks to narrow one term based on the applicant’s description of a prior art reference during prosecution. However, SAP fails to offer any expert testimony in support of its positions. Accordingly, it is unable to meet its clear and convincing evidentiary burden on indefiniteness or demonstrate that the applicant’s arguments during prosecution require a limiting construction.

On the other hand, Valtrus’s positions are supported by two experts: (1) Dr. Michael Brogioli, a leading industry technologist and an Adjunct Professor of Electrical and Computer Engineering at Rice University; and (2) Dr. Martin Farach-Colton, the chair of the Computer Science Department at New York University. As discussed below, Valtrus’s experts explain in detail why SAP’s indefiniteness positions are wrong, that the relevant POSITA would have understood the terms at issue (and, where applicable, the structure or class of structures being claimed), and that SAP’s interpretation of the prosecution history of one term is simply incorrect from a technical perspective. *See* Exs. 3, 15.

In view of SAP’s lack of supporting evidence, the unrebutted positions from Valtrus’s experts, and the other evidence discussed herein, the Court should adopt Valtrus’s positions, as set out in Exhibit A to the parties’ P.R. 4-3 Statement (Dkt. 96).²

¹ The parties’ Joint Claim Construction and Prehearing Statement (Dkt. 96, “P.R. 4-3 Statement”) identified eighteen terms, however, pursuant to the agreed claim-narrowing schedule (Dkt. 65), Valtrus has since withdrawn several claims, mooted five of the claim construction disputes.

² The parties have agreed to the constructions set out in their P.R. 4-3 Statement. Dkt. 96 at 1-2. Valtrus respectfully requests that the Court enter those constructions.

II. ARGUMENT

A. SAP Cannot Meet Its Clear and Convincing Burden on Indefiniteness

SAP argues that all but one of the terms at issue are indefinite. SAP must demonstrate indefiniteness by clear and convincing evidence. *BASF Corp. v. Johnson Matthey Inc.*, 875 F.3d 1360, 1365 (Fed. Cir. 2017). SAP has an additional burden as it concerns “means-plus-function” terms. SAP must overcome the presumption that claims without the word “means” are not means-plus-function claims. *See Dyfan, LLC v. Target Corp.*, 28 F.4th 1360, 1365 (Fed. Cir. 2022).

Further, SAP has proffered its indefiniteness positions without any supporting expert evidence. While this is not dispositive *per se*, attorney argument alone cannot outweigh the evidence from Valtrus’s experts, who explain in detail why the claims are not indefinite to a POSITA and why, in the case of the alleged means-plus-function terms, they describe sufficient structure. *See Wasica Fin. GmbH v. Cont’l Auto. Sys., Inc.*, 853 F.3d 1272, 1284–85 (Fed. Cir. 2017) (affirming Board acceptance of patentee’s proffered expert testimony over patent challenger’s bare attorney argument without “adduc[ing] any evidence to the contrary”); *see also Elcommerce.com, Inc. v. SAP AG*, 745 F.3d 490, 502 (Fed. Cir. 2014) (finding that SAP’s failure to provide expert testimony or other evidence of the existing knowledge in the relevant field prevented it from meeting its burden on indefiniteness), *vacated on other grounds*, 564 F. App’x 599 (Fed. Cir. 2014)³; *Enzo Biochem, Inc. v. Gen-Probe, Inc.*, 424 F.3d 1276, 1284 (Fed. Cir. 2005) (“Attorney argument is no substitute for evidence.”); *Dyfan*, 28 F.4th at 1366–67 (“[W]here

³ Vacated opinions may be used as persuasive authority. *See Lolli v. County of Orange*, 351 F.3d 410, 413 (9th Cir. 2003) (“[T]he reasoning of a vacated opinion may be looked to as persuasive authority if its reasoning is unaffected by the decision to vacate” (cleaned up)); *Finberg v. Sullivan*, 658 F.2d 93, 101 (3d Cir. 1980) (“Even if a decision is vacated, however, the force of its reasoning remains, and the opinion of the Court may influence resolution of future disputes.”). Here, the Federal Circuit vacated the *Elcommerce.com* panel opinion after the PTAB invalidated the asserted claims and the parties jointly moved to dismiss the appeal while a petition for *en banc* review was pending. *See* Dkt. 50, 51 (Case No. 11-1369).

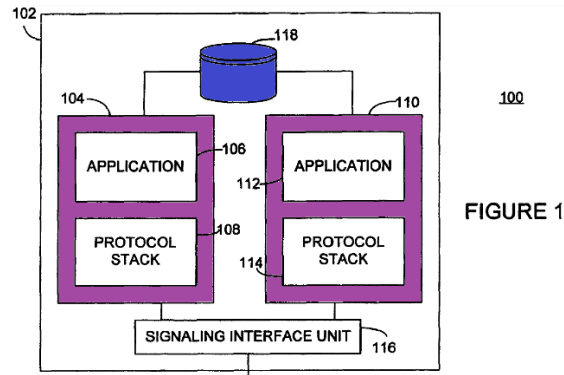
it is clear that a claim term itself connotes some structure to a [POSITA], ‘the presumption that § 112, ¶ 6 does not apply is determinative’ in the absence of ‘more compelling evidence of the understanding of [a POSITA].’” (quoting *Apex Inc. v. Raritan Comput., Inc.*, 325 F.3d 1364, 1373 (Fed. Cir. 2003)).

B. Disputed Terms of U.S. Patent No. 7,936,738

1. Background for the ’738 Patent

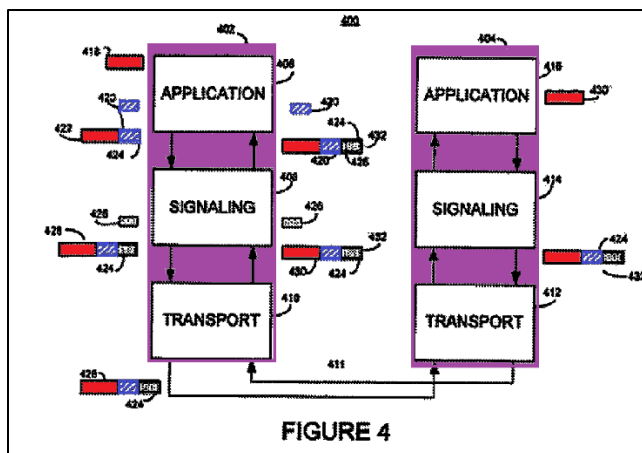
The ’738 patent is directed to providing fault tolerance and high availability in critical communication systems. Ex. 14 at 1:1-16. Typically, such a system would have one “active” system and one “standby” system. *Id.* at 1:17-26. In the event of a failure of the active system, a switchover occurs (*i.e.*, “context-switch,” *see* Ex. 15, ¶ 91), such that the standby system becomes the active system and vice-versa. Ex. 14 at 1:20-23. In this way, services that were available before the failure would still be available once the switchover was complete. *Id.* at 1:23-26.

While this basic layout may generally provide “service continuity” (*e.g.*, the ability to make new calls in a telephone network continues after the switchover), more is required to provide “task preservation” (*e.g.*, calls that were active during the failure continue as if nothing happened, without being dropped). *Id.* at 1:27-45. Task preservation requires that context information be accessible to both active and standby systems. *See id.* at 1:46-60. Prior art systems would therefore provide “a common storage element” where context information relating to each task was stored and accessible to both active and standby systems. *Id.* at 1:45-52. In the event of a switchover, the stored context information would be used “for reinitializing the new active system with the context of any tasks which were in progress when the switchover occurred.” *Id.* at 1:50-52. Figure 1 depicts the prior art (excerpted below).



Systems 104 and 110 are arranged in a fault tolerant configuration “in which one of the systems is arranged in an ‘active’ mode, whilst the other system is arranged in a ‘standby’ mode.” *Id.* at 4:6-11. In the event of a fault being detected in the current active system, a switchover will occur such that the current active system becomes the standby system, and vice-versa. *Id.* at 4:6-13. The context data stored in common storage area 118 would then be used to rebuild context in the event of a switchover. *Id.* at 4:20-24.

An embodiment of the '738 patent is shown in Figure 4. *Id.* at 3:54-55. Rather than store the context information in a common storage area, context information is stored in messages sent back and forth between nodes. *See id.* at 5:24-42. Context information 420 is appended to the message 418 as it is sent through the layers of the protocol stack between nodes 402 and 404. *Id.* at 5:62-6:11. As such, “the need for a high-availability system to store context data in a central data store is removed by appending context data to a message sent from [a] system, such that a message sent in response thereto contains the context data. In the event of a switchover, the context data received in a response message may be used to restore the context data of one or more layers of the protocol stack of [the] system.” *Id.* at 5:3-10.



2. “context information” (claims 1, 2, 5, 6, 10, 33)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not indefinite	Indefinite

Contrary to SAP’s position, there is nothing indefinite about “context information.” As described by Dr. Farach-Colton, the term would be familiar to a POSITA, especially when referring to “context switching” or “switchover” (as the ’738 patent does). *See* Ex. 15, ¶ 91. Specifically, a POSITA would understand that it “has about the same meaning as it would to a lay person,” *i.e.*, “background or situational information related to a current activity, similar to state information” and “generally analogous to metadata.” *Id.* at ¶¶ 87-89. The ’738 patent specifically describes this use of context information in the prior art:

“In order to provide task preservation a common storage element is often provided in addition to a high-availability configuration. Context data relating to each task is stored in the common storage area, and may be used in the event of a switchover for reinitializing the new active system with the context of any tasks which were in progress when the switchover occurred.

Ex. 14 at 1:46-52.

Moreover, this understanding is consistent with the example of the invention given in the ’738 patent, which states that for “a billing application ... required to know the time a call is established and the time a call ended in order to generate a billing record,” context information can

relate “to the start time of the call.” Ex. 15, ¶ 90.

Accordingly, this term should be construed under its plain and ordinary meaning. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005).

3. “potentially inaccurate or incomplete” (claim 10)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not indefinite	Indefinite

Claim 10 recites:

The method of claim 1, further comprising adding, to the message, an indication associated with the obtained context information where it is determined that the obtained context information is potentially inaccurate or incomplete.

SAP presumably believes that the word “potentially” makes this claim indefinite. That view ignores the context of the claim, which requires “**adding**, to the message, **an indication . . . where it is determined** that the obtained context information is potentially inaccurate or incomplete.” In other words, the claim does not require making some specific determination about the word “potentially”; it requires adding an indication that the information might not be accurate, when there is such a determination. All that matters is that the user determined and added an indication that the context information is “potentially inaccurate or incomplete.” For example, the indication could simply be a note or tag that the obtained information might not be accurate. The claim does not actually require a determination of accuracy or incompleteness, or even some degree of uncertainty. It just requires adding an indication or tag reflecting its potentially inaccurate status.

This is consistent with both the specification and the testimony of Dr. Farach-Colton. For example, the specification describes this phrase as a “flag” that can be attached to context information, *e.g.*, “the start time of [a] call” stored “when the phone starts ringing” is potentially inaccurate because “an indeterminable amount of time [may pass before] the call is answered.”

Ex. 14 at 7:28-52; *see also id.* at 7:45-50 (“Preferably any billing record generated as a result of restored context data should be flagged as such, thereby allowing the service provider to either offer the call free of charge, or to reduce the cost of the call in question to take into account that the billing record generated thereby may be inaccurate.”); Ex. 15, ¶¶ 95-97; *SB IP Holdings, LLC v. Vivint Smart Home, Inc.*, 2021 WL 5804286, at *19 (E.D. Tex. Dec. 7, 2021) (“irregular activity” not indefinite where “it is irrelevant what criteria are used...what matters is that some criteria are used to categorize an activity and then some action is taken.”); *Eolas Techs. Inc. v. Amazon.com, Inc.*, 2016 WL 7155294, at *6 (E.D. Tex. Dec. 8, 2016) (“interactive” not indefinite where “interactive is a binary decision”).

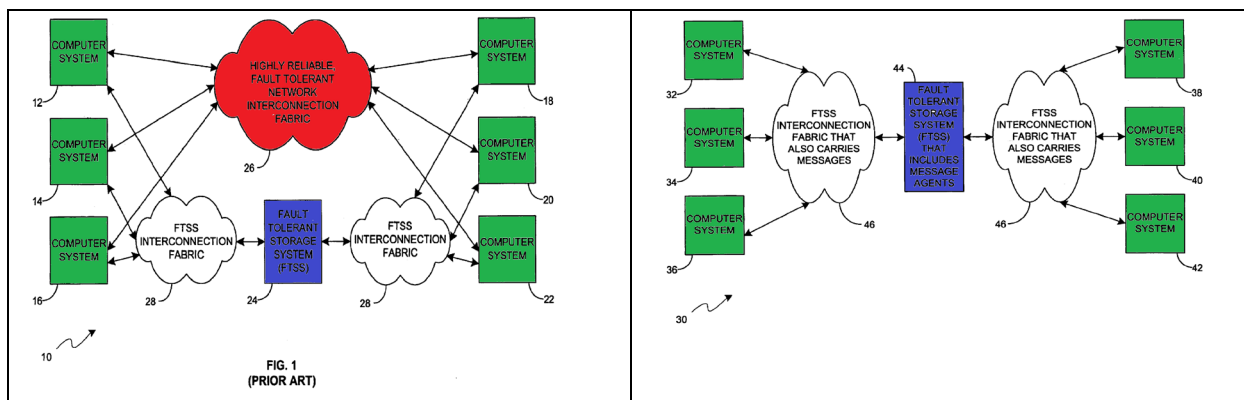
C. Disputed Terms of U.S. Patent No. 6,889,244

1. Background for the '244 Patent

The '244 patent is directed to improved fault-tolerance in the operation of distributed applications (*i.e.*, an application that is separated into two or more parts). One simple example of a distributed application is an application that “allows a customer to order a product over the Internet from an on-line retailer.” Ex. 16 at 1:41-44. This system must allow the web server to provide information about the products for sale, receive orders from the customer, and then accept online payment. *Id.* at 1:47-63. A developer needs to ensure that “all of the disparate applications communicate in a seamless and reliable manner.” *Id.* at 2:2-3. To this end, operation in a distributed application is similar to a conversation between two people in the sense that “a number of messages pass back and forth” and “over the course of the communication, both sides ‘remember’ the point or state of the conversation so that relatively long operations . . . can be supported.” *Id.* at 2:35-38. “[T]he context of any particular message is interpreted in view of the previous messages.” *Id.* at 2:40-42.

This is the problem that the '244 patent is addressing—ensuring “that a distributed

application has a high rate of availability.” *Id.* at 3:65-66. Figure 1 of the ’244 patent, below, depicts a prior art system, whereas Figure 2, below, “shows a highly reliable fault-tolerant system in accordance with the present invention that may be used to host a distributed application.”



In the prior art system (Fig. 1), computer systems 12, 14, 16, 18, 20, and 22 are coupled to a fault tolerant storage system (FTSS) 24 via interconnection fabric 28. The FTSS is a storage system that guarantees the availability and integrity of files stored therein; examples are given in the specification. *See* ’244 patent at 4:28-48. The interconnection fabric 28 carries file transaction systems between the computer systems and the FTSS 24. *Id.* at 4:9-28. This interconnection fabric 28 does not directly couple the computer systems to each other. *Id.* at 4:56-58. Rather, a separate fault tolerant interconnection fabric 26 carries messages between the computer systems. *Id.* at 4:58-62. The problem with this approach is that if a client “suffers some type of error after receiving a message, the client is not able to request retransmission of the message from fabric 26.” *Id.* at 4:67-5:2.

Figure 2 shows a networked system 30 in accordance with the invention. As can be seen, the computer systems are coupled to an FTSS via an FTSS interconnection fabric that also carries messages (46). This configuration “ensures that FTSS 44 can guarantee delivery of a message transmitted from a sending computer system to a destination computer system.” *Id.* at 6:55-57.

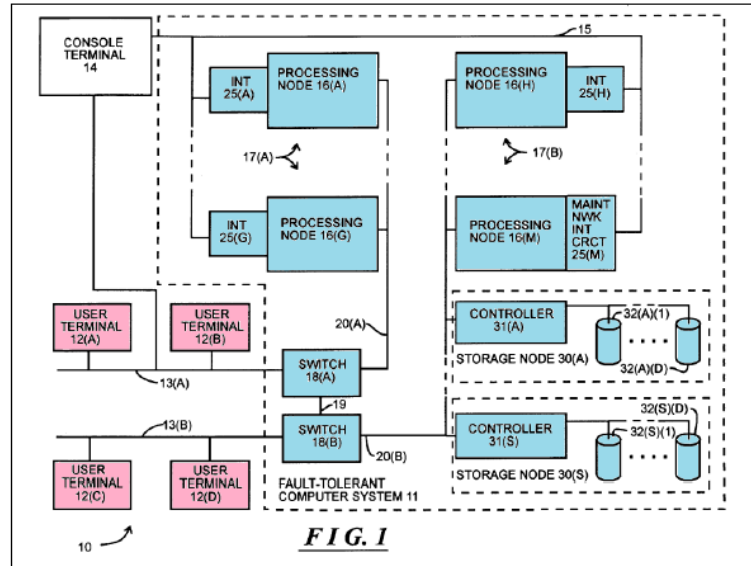
2. “the first and second nodes are each coupled to a fault tolerant

storage system (FTSS)” (claim 1, preamble)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not indefinite.	Plain and ordinary meaning, where the coupling to the FTSS provides for transmitting messages by a store-and-forward method (i.e., not through a conventional computer network such as a local area network (Ethernet) or wide area network (Internet))

A POSITA would understand the preamble of claim 1 to indicate that the FTSS and the first and second nodes are communicatively coupled. Ex. 15, ¶¶ 49-51. This is supported by the specification, which gives numerous examples of technologies used to couple an FTSS to a server, including “an Ultra160 SCSI interface, a fiber channel interface, or an enterprise system connectivity (ESCON) interface.” Ex. 16 at 4:35-48. As Dr. Farach-Colton has explained, a POSITA would understand this to indicate that numerous networking technologies can be used in the invention, including, but not limited to, ethernet and internet. Ex. 15, ¶ 51. Indeed, the disclosure of U.S. Patent No. 6,938,071 at 11:43-12:5 (Ex. 17, incorporated by reference into the ’244 patent at 1:9-15) specifically refers to a node “coupled to [an] FTSS” via ethernet or the internet. Ex. 15, ¶ 52.

Nothing in the prosecution history disclaims this or suggests that the Court should read in limitations like a “store and forward” technique, or exclude “conventional computer network[s],” as SAP demands. The phrases “store and forward” and “conventional ethernet networks” and communications arose in the context of a rejection based on a reference to Utter, reproduced below in relevant part with the examiner’s apparent interpretation of the reference. *Id.* at ¶¶ 54-55. The user terminals 12A-12D (per the examiner, “nodes”) are highlighted in red, and the Fault-Tolerant Computer System (per the examiner, “the FTSS”) is highlighted in blue. *Id.* at ¶ 55.



Repeatedly, the applicant sought to explain to the examiner that communications from one node to the other in the “Utter” reference (*e.g.*, from User Terminal 12(A) to User Terminal 12(B)) would not pass through the FTSS and would not be stored in highly reliable FTSS storage media (and therefore did not anticipate the claims). *See, e.g.*:

For Utter to anticipate, or even be relevant to, the invention claimed in claim 1 of the current application, Utter would need to disclose or suggest a message passed from one user terminal through the fault-tolerant computer system to another user terminal - but Utter does not.

Nov. 24, 2003, Applicant Argument, Ex. 18 at 20 (distinguishing Utter from claim 1).

[I]n Utter’s system, communications between user terminals is conducted in an entirely conventional manner, via entirely conventional Ethernet links, and not through the FTSS, as clearly claimed in the elements of claim 1 directed to Applicants’ invention, and as discussed below.

Apr. 30, 2004, Applicant Argument, *Id.* at 41 (distinguishing Utter’s Ethernet links).

The applicant’s only point was that Utter (and its mere use of conventional ethernet links) did not disclose using the FTSS as the intermediary through which the messages were passed. Ex. 15, ¶¶ 54-55. In other words, Utter did not disclose claim 1’s requirements of “transmitting a message . . . to a communication agent in the FTSS,” “storing the message in . . . the FTSS,” or

“transmitting the message from the FTSS.” The arguments have nothing to do with how the first and second nodes are coupled or whether this is through ethernet or the internet; ethernet and internet connections would be within the scope of the claims if they were used to *transmit messages to and from the FTSS*. But in Utter they did not, and Utter failed to disclose anything about ethernet that might suggest that it inherently contained some FTSS functionality.

SAP’s position as to “store and forward” fails for the same reason: The applicant simply explained that nothing in Utter disclosed passing messages through the FTSS:

Next, the Examiner states that Utter discloses “storing the message in a data structure in highly reliable fault-tolerant storage media of the FTSS [but the cited] passage does not teach, mention, or even remotely suggest storing messages in the highly reliable fault-tolerant storage media of the FTSS for later forwarding to another computer.”

Apr. 30, 2004, Applicant Argument, Ex. 18 at 41; *see also* Ex. 15, ¶¶ 59-63.

Accordingly, SAP cannot meet its burden to show a disclaimer as to how nodes are coupled to the FTSS. *See Tech. Props. Ltd. LLC v. Huawei Techs. Co., Ltd.*, 849 F.3d 1349, 1357-58 (Fed. Cir. 2017) (“When determining whether disclaimer applies, we consider the statements in the context of the entire prosecution.”); *Aylus Networks, Inc. v. Apple Inc.*, 856 F.3d 1353, 1363 (Fed. Cir. 2017).

Moreover, even if this Court finds disclaimer, SAP’s construction must nevertheless be rejected. The patent applicant never discussed “internet,” or “wide area networks” whatsoever. *See In re Taasera Licensing LLC, Pat. Litig.*, 2023 WL 8628323, at *3 (E.D. Tex. Dec. 13, 2023) (“[F]or the Court to adopt their construction, Defendants must show ‘clear and unmistakable statements by the patentee that limit the claims’”). And any disclaimer of “conventional ethernet” must be limited to that standard as it existed at the time of the ’244 patent. *See PC Connector Sols. LLC v. SmartDisk Corp.*, 406 F.3d 1359, 1363 (Fed. Cir. 2005) (construing “conventional” as “limited to technologies existing at the time of the invention”); *Chrimar Sys., Inc. v. Alcatel-Lucent*

USA, Inc., 2016 WL 1228767, at *9 (E.D. Tex. Mar. 28, 2016) (limiting the construction of an ethernet standard to that which existed at the time of the invention).

3. “highly reliable fault-tolerant storage media” (claim 1, 2, 3, 6, 9, 10)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not indefinite.	Indefinite

A POSITA would understand that “highly reliable fault tolerant storage media,” as it appears in claim 1, refers “to the physical storage media of the FTSS,” with “highly reliable” not referring to “any specific value or measurement of reliability, but rather as itself a quality that a system could provide.” Ex. 15, ¶¶ 66-73. As Dr. Farach-Colton explains, the ’244 patent uses the terms “highly reliable” and “fault tolerant” to generally describe systems with additional indications or guarantees of reliability beyond non-redundant, single point of failure systems. *Id.* at ¶ 69. The ’244 patent gives numerous such examples. *Id.* SAP provides no evidence to the contrary. *See Team Worldwide Corp. v. Wal-Mart Stores, Inc.*, 2018 WL 1353116, at *7 (E.D. Tex. Mar. 15, 2018) (“Defendants have failed to carry their burden to establish that ‘substantially airtight’ []renders any claim indefinite.”).

Moreover, the ’244 patent provides a “specific and unequivocal” example of “highly reliable fault-tolerant storage media.” *See DDR Holdings, LLC v. Hotels.com, L.P.*, 773 F.3d 1245, 1260 (Fed. Cir. 2014). The ’244 patent specifically describes that “the SureStore(R) XP512 Disk Array uses a fault-tolerant, redundant architecture that ensures there is no single point-of-failure, and has a mean time between failure (MTBF) of 2.5 million hours, or alternatively 285 years.” Ex. 16 at 4:39-43. The ’244 patent also discusses the phrase “highly reliable” with reference to how programmers versed in distributed applications understood the term: “a designer of a distributed application will often specify that the application be implemented using **highly reliable** and fault tolerant interconnection fabrics and storage systems.” *Id.* at 4:5-8 (emphasis added). This is further

evidence that the term would have been understood by a POSITA; the inventors of the '244 patent used it because that is the type of phrase used by engineers at the time.

4. “accessing determining which nodes should receive the broadcast message based on the broadcast message” (claim 10)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
determining which nodes should receive the broadcast message based on the broadcast message	Indefinite

The inclusion of the verb “accessing” before the additional verb “determining” is an error and should be corrected. A district court may correct “obvious minor typographical and clerical errors in patents.” *Pavo Sols. LLC v. Kingston Tech. Co., Inc.*, 35 F.4th 1367, 1373 (Fed. Cir. 2022) (citations omitted). Correction is appropriate “if (1) the correction is not subject to reasonable debate based on consideration of the claim language and the specification and (2) the prosecution history does not suggest a different interpretation of the claims.” *Id.* The error must be “evident from the face of the patent.” *Id.*

The error here meets all the above criteria. It is clear that there should not be two verbs in a row, so one is an error. And, “accessing” makes no sense grammatically. The only verb that fits grammatically is “determining,” given that it is followed by “which.” In other words, deleting “determining,” would leave the claim as “accessing which nodes should receive . . .” That makes no sense. But, “determining which notes should receive,” makes perfect sense in the context of the claim.

This is confirmed by comparing claims 6 and 10. Claims 6 and 10 are structurally parallel, but claim 6 requires “an event-based communication paradigm” whereas claim 10 requires “an unsolicited communication paradigm.” Per the specification, “unsolicited communication is similar to event-based communication, except that there is no subscription function.” Ex. 16 at

11:15-17. Claim 6 recites “**accessing a subscription manager to determine** whether any nodes have subscribed to receive the event represented by the post message.” *Id.* at 13:18-20. Claim 10 does not have a subscription manager to access (per the specification’s description of an “unsolicited communication paradigm.”). *Id.* at 7:14-56; 14:29. Therefore, there is no need for the term “accessing”; it is a clear mistake. Claim 10 only requires “determining.” *Id.* at 14:39; *see Ollnova Techs. Ltd. v. ecobee Techs., ULC*, 2023 WL 2871051, at *5 (E.D. Tex. Apr. 10, 2023) (applying plaintiff’s requested correction where “the correction is not subject to reasonable debate based on consideration of the claim language and the specification.”).

D. Disputed Terms of U.S. Patent No. 7,313,575

1. Background of the ’575 Patent

The ’575 patent is about business intelligence, business processes, and the data used when business decisions are made. *See* Ex. 19 at 1:5-17; 2:36-49; 5:39-43; 20:21-22. These tasks often require an integrated or common data model, and when data originated from disjointed sources, some initial processing would be necessary as a first step. *See id.* at 1:10-17; 1:34-36; 20:24-27; 2:49-54. While tools and methods operating in “batch mode” were available to do this processing, techniques that could enable this to be done in real time and with low latency were limited. *Id.* at 1:14-16.

For example, two such techniques in the prior art were to: (1) require that all applications share a standard data model or (2) use “table-driven code,” in which information about data (*i.e.*, metadata) is stored in tables with each row defining a data element and each column specifying the behaviors used to process that data element. *Id.* at 1:34-41; 12:47-59. A shared data model was undesirable because it could require rewriting of application code. *See id.* at 1:36-38. And while table-driven code would not necessarily require rewriting applications, each special case or new behavior to be applied to some data element resulted in the addition of a new column. *Id.* at 12:60-

64. The additional columns could eventually result in a large but mostly empty table that was difficult to maintain. *Id.*

Accordingly, rather than table-driven code or a standard data model, the '575 patent uses polymorphic metadata, *e.g.*, metadata represented using the Java object-oriented programming language, with individual metadata objects having different numbers and types of attributes determined by class. *See id.* at 3:65-4:1; 4:57-64; 13:3-12; 36:25. With this method, when a special case is encountered, rather than rewriting an application or increasing the size of an already large and mostly empty table, special cases of metadata are simply represented as subclasses of more general metadata classes. *Id.* at 13:13-19.

For example, metadata for a new document can be created “by creating a new Java class...conventionally but optionally named after the document name.... [F]or creation of metadata for a native INVOICE document, the name of the Java file can be ‘NativeInvoicesjava’.” *Id.* at 38:1-7. The new Java class extends the metadata using a preexisting DocumentDefinition class. *Id.* at 38:7-8. Accordingly, “the metadata classes extensively use polymorphism.” *Id.* at 3:67-4:1.

2. “polymorphic metadata” (claims 1, 4, 28, 33)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not indefinite	Indefinite

There is nothing indefinite about “polymorphic metadata” in the context of the '575 patent. A POSITA would have understood this term to refer to metadata⁴ defined using a polymorphic mechanism such as Java’s hierarchal inheritance (Ex. 15, ¶¶ 129-130), which allows objects of

⁴ Metadata refers to data about data. Ex. 15, ¶ 128; Ex. 19 at 4:18-19 (“[M]etadadata [is] defined as data that describes various aspects of other data.”).

different types within the same class (at the bottom of the hierarchy) to be treated as instances of a common superclass (at the top of the hierarchy). For example, dogs and cats are both pets, so a *Dog* object and a *Cat* object can both be instances of a *Pet* superclass. *See* Ex. 15, ¶¶ 104-110. The ’575 patent gives similar examples, using shapes instead of pets. *See, e.g.*, Ex. 19 at 4:40-50 (“For example, circles and rectangles can be assigned to different classes of Shape with Circle having a defining attribute called diameter, and Rectangle having defining attributes called length and width. The polymorphic representation is economical because specified attributes or parameters of metadata can be limited to those that are applicable to a specific pattern under consideration.”); *see also* Ex. 15, ¶ 131 (collecting examples from the ’575 patent).

Both polymorphism and the corresponding hierarchal inheritance mechanism would have been well-known to a POSITA. *See* Ex. 15, ¶ 98-99, 128 (“Polymorphism is an important concept in computer science.”); Ex. 20 at 374 (Douglass Downing and Michael Covington, *Dictionary of Computer and Internet Terms*, Seventh Ed. (2000)) (“Polymorphism – the use of different procedures, each with the same name, which are associated with different object types.”); Ex. 21 at 1 (Thomas J. Sheffler, *A Portable MPI-Based Parallel Vector Template Library* (1995)) (“Traditionally, polymorphism is implemented through a class hierarchy using inheritance”).

For these reasons, the court should construe “polymorphic metadata” in accordance with its plain and ordinary meaning.

3. “assembler” and “disassembler” (claims 1, 4, 36)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not indefinite	Indefinite

The “assembler” and “disassembler” terms are not indefinite. The “assembler” refers to the instructions or software for getting the information out of the database and into a format to be used, and “disassembler” refers to its complement—putting information into the database. Ex. 15, ¶¶

134, 139. Both terms would have been familiar to a POSITA. *Id.* at ¶¶ 133-34, 138-39. For example, disassembly and reassembly are described in St. Laurent et al.’s *Inside XML DTDs* (1999): “[u]sing a database repository to disassemble and reassemble documents.” Ex. 22 at 426.

Moreover, both terms are fully explained by the ’575 patent. *See, e.g.*, Ex. 19 at 3:48-54:

A “document disassembler” contains prepared statements for inserting or updating various records related to a particular class of inbound documents into the data store. A “document assembler” contains prepared Statements for selecting various records related to a particular class of outbound documents from the data store.

Because a POSITA would understand these terms with reasonable certainty, they are not indefinite. *Nautilus, Inc. v. Biosig Instruments, Inc.*, 572 U.S. 898, 910 (2014).

4. “the RTID that ... create at least one assembler defined according to the polymorphic metadata ...” (claim 28)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
the RTID that ... create[s] at least one assembler defined according to the polymorphic metadata...	Indefinite. <i>See</i> above.

For reasons discussed above, this term is not indefinite. However, it contains an obvious typographical error (a missing “s” after “create”) that should be corrected. *See CBT Flint Partners, LLC v. Return Path, Inc.*, 654 F.3d 1353, 1358 (Fed. Cir. 2011) (“[A] district court may correct an obvious error in a patent claim.”).

E. Disputed Terms of U.S. Patent No. 6,823,409

1. Background Concerning the ’409 Patent

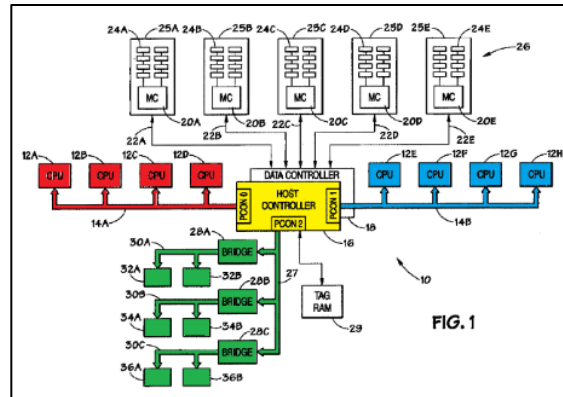
The ’409 patent is directed to systems and methods for maintaining coherency between main memory and separate memory storage areas called “caches” in multi-processor, multi-processor-bus computer systems. Ex. 1, Abstract, 1:8-12, 2:57-3:36; Ex. 3, ¶ 40.

As Dr. Brogioli explains, at the time of the ’409 patent, and even now, “software developers would ideally like as much memory as possible in a computing system for their applications to

utilize. At the same time, software developers would also like this memory to be accessible as fast as possible, such as at the rate at which the central processing unit ('CPU') of the system operates. In practice, achieving both in a computing system is not possible." Ex. 3, ¶ 53. To maximize speed and capacity, and reduce cost, computer architects use "hierarchical memory system designs whereby several levels of memory are used to implement the overall computing system. These designs generally place one or more levels of smaller . . . and faster . . . memory called 'cache' physically closer to a CPU to be accessed at lower latency and place the larger . . . and slower . . . main memory comparatively farther from the CPU to be accessed at higher latency." *Id.* "[F]requently accessed data may be stored in the cache memory. . . ." Ex. 1 at 2:40-59.

As the '409 patent acknowledges, "[t]he process for managing the caches in a multi-processor system is complex." *Id.* at 2:59-61. In such systems, "[e]ach processor may have its own memory cache" *Id.* at 3:64-4:4. The system must therefore "maintain coherency across the different cache memories and the main memory.... by [e]nsuring that any shared operand that has changed in any cache is changed throughout the entire system." *Id.* Accordingly, "'[c]ache coherence' refers to a protocol for managing the caches of a multi-processor system so that no data is lost or over-written before the data is transferred from a cache to a requesting or target memory." *Id.* at 3:61-64.

Figure 1 of the '409 patent (below) depicts an exemplary multi-processor, multi-bus system in which CPUs 12A-12D (red), CPUs 12E-12H (blue), and peripheral devices 32A-B, 34A-B, or 36A-B (green) are coupled to a chip set, which includes a host controller 16 (yellow) and a data controller 18, via processor bus 14A (red), processor bus 14B (blue), and I/O bus 27 (green), respectively. Ex. 1 at 4:3-62, Fig. 1 (annotated); Ex. 3, ¶ 108.



“[E]ach CPU 12A-12H [(red and blue)] may include a segment of cache memory for storage of frequently accessed data and code. Coherency between the caches found in each CPU 12A-12H [(red and blue)] is further complicated by the split bus configuration since coherency must be maintained between the separate buses [14A (red) and 14B (blue)]. Also, because requests may originate from or be directed to not only one of the CPUs 12A-12H [(red and blue)] but also from one of the peripheral devices 32A-B, 34A-B, or 36A-B [(green)], cache coherency must be maintained along the I/O bus 27 [(green)], as well.” Ex. 1 at 4:53-62, Fig. 1 (annotated).

This is the issue to which the ’409 patent is directed—improved, faster, and more efficient systems and methods for maintaining coherency across cache memories in a complex multi-processor bus architecture such as that shown in Figure 1. *See Id.* at 1:9-12.

As both the ’409 patent and Dr. Brogioli explained, at the time of the ’409 patent, there were two main mechanisms for maintaining cache coherency that POSITAs would have understood and implemented—a directory-based system or a snooping system. In a “Directory Based implementation, the sharing status, or state, of a given block of main memory, is kept in a singular location, called the directory. In a Snooping Based implementation, every cache that has a copy in its cache block of a given block of main memory also keeps a copy of the sharing status of that block. The computing system having a Snooping Based implementation does not keep centralized state information. The caches usually communicate via a shared memory bus, and when

a CPU requests data or instructions from a block of main memory, all cache controllers corresponding to the caches on this shared memory bus will themselves check, or ‘snoop,’ this shared memory bus to make decisions as to whether or not they have a copy of a block that is requested on the bus.” Ex. 3, ¶ 69; Ex. 1 at 3:4-20.

The ’409 patent focuses on a mechanism for maintaining coherency across caches in such a snooping based system. Specifically, “a mechanism is provided in the host/data controller 16, 18 to efficiently facilitate snooping of the buses in the present multi-processor/multi-bus system 10.” Ex. 1 at 4:62–65. Figure 2, below, of the ’409 patent depicts one such mechanism.

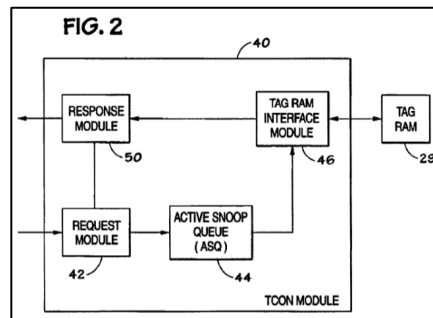


Figure 2 shows a tag control (TCON) module 40 used to “manage the snooping process efficiently.” Ex. 1 at 5:13-16, Fig. 2; *id.* at 4:62-65, 5:13-24; 7:13-15, Fig. 1; Ex. 2 at VAL-SAP00004452-VAL-SAP00004453; Ex. 3, ¶ 108. It “serves as a snoop filter to minimize the number of clock cycles required to obtain cache state information and to determine the proper snoop response on the processor or I/O buses[.]” Ex. 1 at 5:16-20. The TCON module 40 contains four functional blocks—request module 42, active snoop queue (ASQ) 44, TAG RAM interface module 46, and response module 50. Ex. 1 at 5:25-6:67, Fig. 2; Ex. 3, ¶¶ 109, 126, 149.

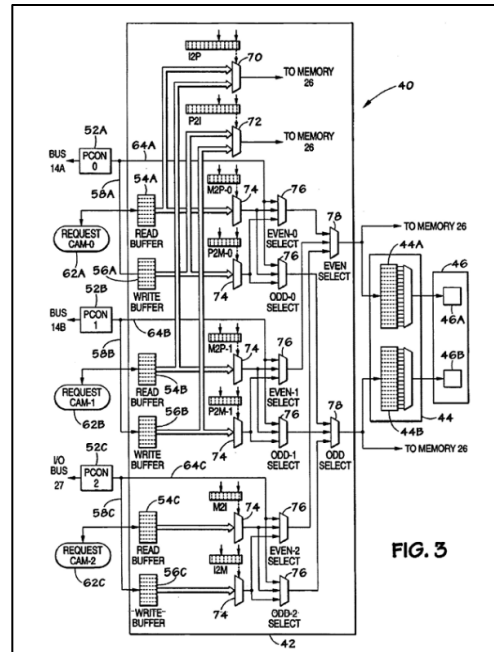


Figure 3 of the '409 patent, above, shows the request module 42, ASQ 44, and tag RAM interface module 46 of the TCON module 40 implemented in circuitry. Ex. 1 at 7:13-11:65, Fig. 3, Table 1; Ex. 3 at ¶¶ 110, 127, 150. The specification details the associated logic for implementing the TCON module as well as preferred techniques. Ex. 1 at 5:13-11:65.

2. “coherency control module” terms (claims 1, 4)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not subject to § 112, ¶ 6. Not indefinite.	Indefinite. ⁵ Alternatively, subject to § 112(6/f) and indefinite for failing to recite corresponding structure.

SAP asserts that “coherency control module” is written in means-plus-function format and that there is no corresponding structure in the specification, such that the claim is indefinite.

⁵ SAP’s asserts that “coherency control module” is indefinite even if not subject to Section 112, ¶6. SAP has never explained the basis for its position. “Coherency control module” is described in the '409 patent and prosecution history and would not have been reasonably uncertain to a POSITA. Ex. 1 at 1:66-3:34, 4:3-65, 5:13-24, Fig. 1; Ex. 2 at VAL-SAP00004452–VAL-SAP00004453; Ex. 3, ¶¶ 74-90 (also citing numerous publications, textbooks, and cited prior art); *see also, supra* Section II.E.1.

However, as noted above, SAP offers no expert testimony in support of any position. *See supra* Section II.A. On the other hand, Dr. Brogioli explained in his declaration that a POSITA would have understood this claim term to refer to “the structural component on a multi-processor that contains the circuitry and logic that maintains coherency in cache memory.” Ex. 3, ¶ 91. Dr. Brogioli explains that cache control, and related terms, were well known concepts in the art, and were “generally recognized as specific structures in microprocessors.” *Id.* at ¶ 92. He cites numerous publications and textbooks in support, all of which describe cache coherence mechanisms using a “class of known structures.” *Id.* at ¶¶ 93-98. For example, “Hennessy (1999) introduced a hardware-supported scalable cache coherence system including a memory and general interconnect controller (‘MAGIC’) that handles coherence communications between processor nodes” Ex. 4 at VAL-SAP00016749, VAL-SAP00016758, Fig. 11; Ex. 3, ¶ 95. And “Nanda (2001) is an article describing an architecture for high-performance cache coherence” along with a depiction of the physical structures in the disclosed cache controller. Ex. 5 at VAL-SAP00016911-VAL-SAP00016912, VAL-SAP00016919, Figs. 1-2; Ex. 3, ¶ 97.

As Dr. Brogioli also explains, the ’409 patent is no different in this respect, using “coherency control module” terms “in a way that is consistent with and supporting a POSITA’s understanding of those terms as describing a structure.” Ex. 3, ¶¶ 99-111. Specifically, Dr. Brogioli explains that the TCON module 40 (*see* Section II.E.1, *supra*) is an embodiment of the coherency control module and operates to efficiently facilitate snooping of the buses. Ex. 3, ¶ 108. Figure 2 of the ’409 patent shows the layout of the various components of the TCON module 40, while Figure 3 shows the physical structures within the TCON module used for implementing cache control. Ex. 1 at 5:25-6:67, 7:13-11:65, Figs. 2 & 3, Table 1; Ex. 3, ¶¶ 108-109.

Thus, the ’409 patent is consistent with the state of the art at the time in using the term

“coherency control module” to refer to structures used in maintaining coherency in cache memories. *See* Ex. 3, ¶¶ 91-111. All of the above testimony from Dr. Brogioli—that a coherency control module refers to a known class of structures—is un rebutted. Accordingly, there is no credible basis for holding that “coherency control module” is written in means-plus-function format, particularly given the absence of the word “means.” *See Dyfan, LLC v. 28 F.4th* at 1365; *Hillman Group, Inc. v. KeyMe, LLC*, 2020 WL 3605626, at *16 (E.D. Tex. Jul. 2, 2020). Moreover, the term “module” as used here is not synonymous with “means”—when read in context of the subject patent as a whole and as explained through un rebutted expert testimony the claim term refers to a known class of structures. *See, e.g., IPCom GmbH & Co. KG v. AT&T Corp.*, 2021 WL 4191407, at *24-25 (E.D. Tex. Sept. 15, 2021) (“transmission radio module” denotes a known class of structures and is not governed by § 112, ¶ 6); *SAS Inst. Inc. v. World Programming Ltd.*, 2020 WL 569856, at *15–16 (E.D. Tex. Feb. 5, 2020) (“graph generator module” not means-plus-function based on surrounding claim language that “inform[s] the structural character” and use in the field to denote structure); *see also, e.g., Microchip Tech. Inc. v. Nuvoton Tech. Corp. Am.*, 2020 WL 978636, at *8-12 (N.D. Cal. Feb. 28, 2020) (“interface module,” “serial engine module,” and “port control module” not governed by § 112(f)).

Accordingly, Section 112, ¶ 6 does not apply.⁶

3. “request module” terms (claims 1, 7)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not subject to § 112, ¶ 6. Not indefinite.	Subject to § 112(6/f) and indefinite for failing to recite corresponding structure.

⁶ Even if the Court finds that “coherency control module” is governed by § 112, ¶ 6, the ’409 patent discloses corresponding structure for its function of maintaining coherency in cache memory, with an example. *See, e.g.,* Ex. 1 at 5:13-8:39, 7:13-11:65, Figs. 2, 3; Ex. 3, ¶¶ 91, 112, 113.

Similar to “coherency control module,” “request module” is not subject to § 112, ¶ 6. *See Dyfan*, 28 F.4th 1365; *Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1349 (Fed. Cir. 2015).

As set out above in Section II.E.1, *supra*, the TCON module 40 (an example of a “coherency control module” claimed in the ’409 patent) contains four functional blocks. Ex. 2 at VAL-SAP00004452-VAL-SAP00004453; Ex. 3, ¶ 88. One is the “request module,” shown, for example, in Figure 2 as request module 42. Ex. 1 at 5:25-6:2, Fig. 2; Ex. 3, ¶ 126.

Dr. Brogioli explained that a POSITA would understand “request module” to refer to known structural components in the context of the ’409 patent. Ex. 3, ¶ 117. Specifically, “a POSITA would understand request module to describe definite structures implemented in a hardware architecture, not just functions performed by a generic computing device.” *Id.* This is consistent with the state of the art at the time, as well as the description in the ’409 patent. *Id.* at ¶¶ 118-128. Dr. Brogioli explained that Figure 3 depicts the components of a request module 42 and its layout within the TCON module 40. *Id.* at ¶ 127. This includes a number of components “that a POSITA would recognize as known physical structures, for example, arbiters/multiplexors . . . connections to memory 26 . . . [and] buffers.” *Id.*; Ex. 1 at 7:13-11:65, Fig. 3. A POSITA reading these disclosures would understand that “request module” refers to a class of definite structures used in receiving requests. Ex. 3, ¶ 128.

Given Dr. Brogioli’s unrebutted expert testimony, the disclosure of the ’409 patent, and the fact that “request module” does not include the term “means,” the Court should hold that § 112, ¶ 6 does not apply.⁷

⁷ Even if the Court finds that “request module” is governed by § 112, ¶ 6, the ’409 patent discloses corresponding structure for its function of receiving requests to access cache memory, with an example. *See, e.g.*, Ex. 1 at 5:13-6:2, 7:13-8:15, 10:30-56, 11:9-65, Figs. 2, 3; Ex. 3, ¶¶ 116, 129, 130.

4. “static RAM interface module” terms (claims 1, 7, 9)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not subject to § 112, ¶ 6. Not indefinite.	Subject to § 112(6/f) and indefinite for failing to recite corresponding structure.

As with “coherency control module” and “request module,” the “static RAM interface module” is not subject to § 112, ¶ 6. *See Dyfan*, 28 F.4th 1365; *Williamson*, 792 F.3d at 1349.

It is Dr. Brogioli’s un rebutted testimony that a “POSITA would readily understand the static RAM interface module is the structural component in a multiprocessor that contains the circuitry and logic that accesses static RAM (e.g., cache memory).” Ex. 3, ¶ 133. Per Dr. Brogioli, the word “interface” is generally recognized as specific structure in microprocessors (e.g., the structure that “facilitates interactions between two or more components or between the microprocessor and an external device, including with different types of memory.”) *Id.* at ¶ 134. Numerous publications and patents from the time of the ’409 patent invention describe mechanisms for deploying such an interface. *Id.* at ¶¶ 135-142. The ’409 patent’s use of “static RAM interface module” in the context of the claim language is consistent with how a POSITA would have understood an “interface.” Figures 2 and 3, for example, detail the positioning of the components of tag RAM interface module 46, an example of the static RAM interface module claimed in the ’409 patent, relative to the other modules within the TCON module 40 (an example of the coherency control module) and with known physical component tag RAM 29. Ex. 1 at 5:25-11:65; Ex. 3, ¶ 149-150; *see supra* Section II.E.1.

As with “coherency control module” and “request module,” the “static RAM interface module” terms are presumed not to be means-plus-function terms because they do not use the word “means.” *See Dyfan*, 28 F.4th at 1365; *Williamson*, 792 F.3d at 1349; *Hillman*, 2020 WL 3605626, at *16. Neither use of the word “module” nor its modifier “static RAM interface” can overcome

that presumption. *See Zeroclick, LLC v. Apple Inc.*, 891 F.3d 1003, 1008 (Fed. Cir. 2018). Also, the claim language surrounding “static RAM interface module” in the claims of the ’409 patent informs the structural character of the terms. *See SAS Inst.*, 2020 WL 569856, at *15-16; *see also*, *e.g.*, *Microchip Tech.*, 2020 WL 978636, at *8-12.

Accordingly, Section 112, ¶ 6 does not apply.⁸

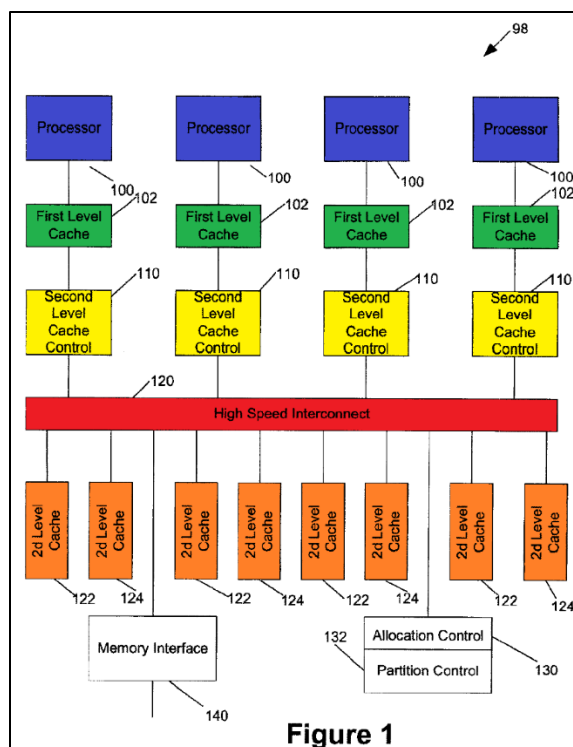
F. Disputed Term of U.S. Patent No. 6,871,264

1. Background of the ’264 Patent

The ’264 patent, like the ’409 patent, relates to the use of cache memory. The ’264 patent, however, is directed to the dynamic allocation of a cache shared by different processors on a multi-processor integrated circuit, so as to provide an amount of cache that is appropriate for a particular thread being executed on a particular processor on the system. Ex. 6 at 3:9-13.

For example, Figure 1 of the ’264 patent (reproduced below), shows a partitionable multiple-processor integrated circuit 98 with several processors 100 (blue). Each processor 100 (blue) performs memory references for instructions data through a first level cache 102 (green). Those references that miss (*i.e.*, where the reference is not found in the cache) are directed to a second level cache controller 110 (yellow), which are themselves coupled to a high-speed interconnect 120 (red). This high-speed interconnect 120 (red) allows the second level cache controllers 110 (yellow) to each access one or more of the second level cache memory blocks 122 and 124 (orange). At least one is an allocable cache memory block 124 (orange) that may be allocated to any cache controller 110 (yellow). Ex. 6 at 4:10-21; Ex. 7 at VAL-SAP00007195; Ex. 3, ¶ 162.

⁸ Even if the Court finds that “static RAM interface module” is governed by § 112, ¶ 6, the ’409 patent discloses corresponding structure for its function of accessing static RAM, with an example. *See, e.g.*, Ex. 1 at 6:3-7:12, 10:20-11:8, Figs. 2, 3; Ex. 3, ¶¶ 133, 152, 153.



2. “high-speed interconnect coupling” (claim 1)

<u>Valtrus’s Proposed Construction</u>	<u>SAP’s Proposed Construction</u>
Plain and ordinary meaning. Not indefinite.	Indefinite.

“[H]igh-speed interconnect coupling” appears in claim 1 of the ’264 patent, which recites:

1. A processor integrated circuit capable of executing more than one instruction stream comprising:

- a first processor, coupled to . . . a first cache controller;
- a second processor, coupled to . . . a second cache controller;
- a plurality of cache memory blocks;

a high-speed interconnect coupling the plurality of cache memory blocks to the first and second cache controllers such that at least one allocable cache memory block is capable of being used by the first and second cache controllers . . .

Consistent with claim 1, the ’264 patent specification explains that the high-speed interconnect “permits use of the dynamically allocable cache units by more than one real or virtual processor[.]” Ex. 6 at 3:22-25; 4:10-21; Ex. 7 at VAL-SAP00007195; Ex. 3, ¶ 161. The ’264 patent also explains that “[c]ache memory offers substantially faster access than main memory.” Ex. 6 at

2:11-14; Ex. 3, ¶ 160.

As Dr. Brogioli explained, a POSITA would understand from the '264 patent and the prosecution history that high-speed interconnect 120, an example of the claimed high-speed interconnect, supports data transfer between each of the processors 100, via its corresponding second level cache controller 110, and cache memory blocks 122 and 124. Ex. 3, ¶ 162. Dr. Brogioli also explained that a POSITA would understand that the high-speed interconnect is high-speed relative to other parts of a computer system, like main memory, that operate slower than the cache memory block, citing various books and publications. Ex. 3, ¶¶ 163–167; Ex. 8 at VAL-SAP00016665-VAL-SAP00016666.

Patents cited during prosecution of the '264 patent and dictionaries available at the time similarly described cache memory as a “high-speed” memory or “very fast.” Ex. 3, ¶¶ 165-167; *see also* Exs. 9-13. The cited patents also described how accessing main memory was “relatively slow” compared to accessing cache memory, and how the difference as “drastic” enough that effective cache management can be a “dominant factor” in determining the speed of a microprocessor. Ex. 3, ¶¶ 165, 166; *see also* Ex. 11-13.

Accordingly, the '264 patent provides the required objective boundary for this term—namely that the high-speed interconnect must facilitate use of cache memory by one or more processors, relative to other portions of the system that operate more slowly, like main memory. Ex. 3, ¶¶ 160-168. *See Niazi Licensing Corp. v. St. Jude Med. S.C., Inc.*, 30 F.4th 1339, 1347 (Fed. Cir. 2022) (“[A] patentee need not define his invention with mathematical precision in order to comply with the definiteness requirement.”); *STA Group LLC v. Motorola Sol’s, Inc.*, 2024 WL 3843731 at *11–13 (E.D. Tex. Jul. 10, 2024) (“a low bandwidth WAN” and “a high bandwidth LAN” not indefinite because specification, without numerically specific bandwidths, provided

examples of their capabilities).

Accordingly, “high-speed interconnect coupling” is not indefinite.

III. CONCLUSION

For the foregoing reasons, Valtrus respectfully requests that the Court enter an order adopting Valtrus’s proposed constructions and rejecting the positions SAP advances in its constructions.

Date: January 31, 2025

Respectfully submitted,

/s/ Matthew G. Berkowitz by permission Andrea L. Fair

Matthew G. Berkowitz – LEAD ATTORNEY

Navid C. Bayar

Savannah H. Carnes

Reichman Jorgensen Lehman & Feldberg LLP

100 Marine Parkway, Suite 300

Redwood Shores, CA 94065

Tel: (650) 623-1401

mberkowitz@reichmanjorgensen.com

nbayar@reichmanjorgensen.com

scarnes@reichmanjorgensen.com

Khue V. Hoang

Patrick Colsher

Reichman Jorgensen Lehman & Feldberg LLP

400 Madison Avenue, Suite 14D

New York, NY 10017

Tel: (212) 381-1965

khoang@reichmanjorgensen.com

pcolsher@reichmanjorgensen.com

Philip J. Ekle

Reichman Jorgensen Lehman & Feldberg LLP

1909 K Street NW, Suite 800

Washington DC, 20006

Tel: (202) 894-7310

peklem@reichmanjorgensen.com

Of Counsel:

Andrea L. Fair

Texas Bar No. 24078488

Claire Abernathy Henry

Texas State Bar No. 24053063

MILLER FAIR HENRY, PLLC

1507 Bill Owens Pkwy.

Longview, TX 75604

Tel: (903) 757-6400

andrea@millerfairhenry.com

claire@millerfairhenry.com

Attorneys for Plaintiffs

CERTIFICATE OF SERVICE

The undersigned hereby certifies that, on January 31, 2025 the foregoing document was served on all counsel of record who have consented to electronic service via CM/ECF.

/s/ Andrea L. Fair

Andrea L. Fair