

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC., AMAZON.COM SERVICES LLC,
Petitioner,

v.

NOKIA TECHNOLOGIES OY,
Patent Owner.

IPR2024-01507
Patent 8,996,693 B2

Before JAMESON LEE, STEVEN M. AMUNDSON, and
JASON M. REPKO, *Administrative Patent Judges*.

REPKO, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

I. INTRODUCTION

Amazon.com, Inc. and Amazon.com Services LLC (collectively, “Petitioner”) filed a petition requesting *inter partes* review of claims 1–5, 7–15, and 17–20 of U.S. Patent No. 8,996,693 B2 (Ex. 1001, “the ’693 patent”). Paper 3 (“Pet.”). Nokia Technologies OY (“Patent Owner”) filed a Preliminary Response. Paper 7 (“Prelim. Resp.”).

To institute an *inter partes* review, we must determine “that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a). For the reasons discussed below, Petitioner has shown a reasonable likelihood that it would prevail with respect to at least one claim challenged in the Petition. Thus, we institute an *inter partes* review.

A. Related Matters

The parties have identified *Nokia Corporation v. Amazon.com, Inc.*, 1:23-cv-01232 (D. Del. Oct. 27, 2023) as a related case. Pet. 78; Paper 5, 2 (Mandatory Notices).

B. The ’693 Patent

The ’693 patent generally relates to integrating slow-moving and real-time data processing. Ex. 1001, 1:40–42. Slow-moving data is relatively static data collected over a long time. *Id.* at 1:22–29. This type of data includes user-behavior or service-usage information. *Id.* On the other hand, “real-time,” “fast-moving,” or “dynamic” data is collected as streams, such as social-networking or location-tracking feeds. *Id.*; *see also id.* at 4:47–49, 4:5–22. According to the ’693 patent, conventional systems process slow-moving and real-time data using different, isolated data-processing architectures, which prevents a service provider from using both data types. *Id.* at 1:29–36. An integrated dataset can be more useful than a dataset with

only one type. *Id.* at 4:5–22. For example, a service provider may be able to improve its movie recommendations by integrating ratings from a user profile for two-week period (static historical data) with more current ratings from other users (dynamic data). *Id.*

In integrating the two types of data, the '693 patent's system uses encapsulated computational chains. *Id.* at 4:26–31. The encapsulated computations can be marshalled as data objects that travel with static data. *Id.* Marshalling transforms an object's memory representation to a suitable data format for transmission between processes or to storage. *Id.* at 4:34–42. Marshalling is similar to serialization. *Id.* The analytics engines can then shape the marshalled computational chains and return them to the stream-processing systems as processing elements (PEs) for processing real-time data. *Id.* The real-time processing systems detect changes or trends in real-time data and send those changes or trends to the analytics engines. *Id.* at 4:42–47. The analytics engines then update the marshalled computational chains of processing elements of the stream processing system. *Id.* In this way, the system of the '693 patent purports to provide a “complete lifecycle” of slow- and fast-moving data analytics. *Id.* at 4:47–49.

C. Claims

Of those challenged claims 1 and 11 are independent. Claim 1 is reproduced below:

A method comprising:

determining via a processor at least one processing element of at least one dynamic processing mechanism;

causing, at least in part, a marshalling via the processor of the at least one processing element as at least one data object;

transferring of the at least one data object to at least one static processing mechanism;

processing of the at least one data object by the at least one static processing mechanism; and

after processing, transferring the at least one data object back to the at least one dynamic processing mechanism.

Ex. 1001, 25:33–44.

D. Evidence

Name	Reference	Exhibit No.
Foster	Foster et. al., <i>IBM InfoSphere Streams: Assembling Continuous Insight in the Information Revolution</i> (2011)	1004
Williams	Williams et al., <i>Implementing CICS Web Services</i> (2007)	1005

E. Asserted Grounds

Petitioner asserts that claims 1–5, 7–15, and 17–20 are unpatentable on the following grounds. Pet. 14.

Claims Challenged	Pre-AIA ¹ 35 U.S.C. §	Reference(s)/Basis
1–5, 7–15, 17–20	103	Foster
1–5, 7–15, 17–20	103	Foster, Williams

II. ANALYSIS

A. Level of Ordinary Skill in the Art

According to Petitioner,

A hypothetical person of ordinary skill in the art (“POSITA”) in the field of the ’693 patent would have had a (1) bachelor’s degree in computer science, electrical engineering, computer engineering, or a comparable field of study, and (2) approximately two years of practical experience with data processing and analysis. Additional experience can substitute for the level of education, and vice-versa.

¹ Congress amended § 103 when it passed the Leahy-Smith America Invents Act (AIA). Pub. L. No. 112–29, § 3(c), 125 Stat. 284, 287 (2011). Here, the pre-AIA version of § 103 applies.

Pet. 17–18.

Patent Owner applies Petitioner’s definition in its Preliminary Response. Prelim. Resp. 17.

At this stage, we adopt Petitioner’s uncontested skill-level definition because it is consistent with the challenged patent and prior art of record and supported by Mark E. Crovella’s testimony. *See* Exs. 1003,² 1004, 1005. If the parties believe that the level of skill in the art would affect this case’s outcome, the parties must brief that issue.

B. Claim Construction

The Board is required to construe “only those terms . . . that are in controversy, and only to the extent necessary to resolve the controversy.” *Realtime Data, LLC v. Iancu*, 912 F.3d 1368, 1375 (Fed. Cir. 2019) (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)).

1. Processing Element

Claim 1 recites, in part, “determining via a processor at least one *processing element* of at least one dynamic processing mechanism.” Ex. 1001, 25:34–35 (emphasis added). Claim 11 recites a similar limitation. *Id.* at 26:45–46. Here, the term “processing element” is in controversy.

a. The Petition

According to Petitioner, “‘Processing element’ means ‘data to be processed’” because the challenged patent’s claims and specification use the term in this way. Pet. 19. For example, Petitioner argues that the claims recite determining and marshalling a processing element. *Id.* (citing Ex. 1001, 25:33–44). In Petitioner’s view, “The specification explains that

² Exhibit 1003 is the Declaration of Mark E. Crovella.

[processing elements] are destined to be processed.” *Id.* (citing Ex. 1001, 12:32–48; Ex. 1003 ¶¶ 26–27). Petitioner argues that the processing elements are not limited to any particular type of data. *Id.* (citing Ex. 1001, 3:67–4:5, 6:58–7:16, 15:54–61; Ex. 1003 ¶ 28).

b. The Preliminary Response

Patent Owner argues that “a ‘processing element’ must be a component that carries out data processing.” Prelim. Resp. 20. In Patent Owner’s view, this is consistent with the term’s well-established meaning. *Id.* at 19–20. Patent Owner argues that the challenged patent’s specification, dependent claims, and extrinsic record support its position. *See id.* at 20–27.

At this stage and on this record, we agree with Petitioner. Our reasoning follows.

c. Analysis

We construe the claims using the same claim-construction standard as in a civil action under 35 U.S.C. § 282(b) “in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent.” 37 C.F.R. § 42.100(b).

i. The Claim Language

We begin our preliminary claim construction by looking at the claim language. The claims themselves provide “substantial guidance” regarding the meaning of a term. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) (en banc).

The processing element recited in claim 1 is determined “via a processor.” Ex. 1001, 25:34–35. Next, the method recites “causing, at least in part, a marshalling via the processor of the at least one processing element as at least one data object.” *Id.* at 25:36–38. That is, according to the claims,

a processing element is something that a processor can determine and marshal.

Patent Owner argues that “‘marshalling’ can apply to data along with other objects such as code or parameters.” Prelim. Resp. 12. For example, computational closures can be marshalled. *See* Ex. 1001, 11:33–42, *cited in* Prelim. Resp. 11. Computational closures can be “a granular reflective set of instructions, data, and/or related execution context or state.” *Id.* at 11:26–32, *cited in* Prelim. Resp. 11–12.

Petitioner argues that “[m]arshalling was a generic term for gathering *data* from one process and converting it into a format for storage or for transmission to another process” and “unmarshalling was the inverse process to undo the conversion and extract the original *data*.” Pet. 9 (citing Ex. 1005, 39, § 1.4.3; Ex. 1015, 64) (emphasis added). We emphasize “data” here because it indicates that Petitioner views the marshalling process’s input, and the opposite process’s output, as data. *See id.*

Petitioner’s discussion of marshalling and processing elements omits the specific types of data that are discussed in the challenged patent and by Patent Owner. *Id.* On the other hand, Patent Owner uses the term “data” to exclude objects such as code or parameters. Prelim. Resp. 12.

We see no reason at this stage to restrict the recited processing elements to objects such as code or parameters. The claim itself does not explicitly define processing element as performing any operations. Rather, the processing element is acted upon in the determining and marshalling steps of claims 1 and 11. Because code or data can be determined or marshalled, the claim language indicates that data processing elements can be code or data.

ii. The Specification

The specification supports this view. *See Phillips*, 415 F.3d at 1315 (characterizing the specification as highly relevant and “the single best guide to the meaning of a disputed term”) (citation omitted). The challenged patent states that “a PE may include computational code and/or data.” Ex. 1001, 6:63–64 (emphasis added); *see also id.* at 13:16–21 (same). The use of “or” here indicates that PEs can include data but not code. *Id.* That is, there are embodiments in which the “PE may include data as well as executing code” (*id.* at 7:2), “the PE may include an executable code (e.g., binary code)” (*id.* at 7:36–37), or the PE can be an example of a “remote processor” (*id.* at 12:11–19). Even so, there still is at least one embodiment that includes just data. *See, e.g., id.* at 6:63–64.

Patent Owner argues that, under Petitioner’s proposed construction, claims 1 and 11 would be nonsensical. Prelim. Resp. 21. In Patent Owner’s view, limitation [1a] of claim 1 would read: “determining via a processor at least one [data to be processed] of at least one [mechanism for processing data in real-time or near real-time].” *Id.* (emphasis removed, alterations in original). Patent Owner argues that it is unclear what “at least one data to be processed” of a “mechanism for processing data” means. *Id.*

In Patent Owner’s view, the claim implies that the processing element carries out some processing because it recites that the processing element is a component or part of a dynamic processing mechanism. *Id.* at 20. Patent Owner argues that the challenged patent’s specification agrees with this because Figure 5 shows the processing elements within stream processing cluster 515 of processing platform 511. *Id.* at 20–21 (citing Ex. 1001, 16:7–14). Patent Owner argues that the challenged patent’s specification describes

processing elements as part of stream processing cluster 515, which is different from the data store. *Id.* at 24 (citing Ex. 1001, 16:7–14).

We disagree. The challenged patent explains that “the processing platform 107 determines at least one PE of at least one dynamic processing mechanism (e.g., engine).” Ex. 1001, 13:11–13. The challenged patent then provides an example of what this means: “a PE may include computation code and/or data.” *Id.* at 13:16–17 (emphasis added). That is, a PE of a dynamic processing mechanism can be data. *Id.*

If the processing element is data, such as process state or parameters, we see nothing inconsistent with depicting that data as part of stream processing cluster 515, as in Figure 5, or part of a processing mechanism. For example, stream processor cluster 515 can be a dynamic process. *Id.* at 16:8. Stream processor cluster 515 includes “an S4 engine and various PEs with access to one or more data stores.” *Id.* at 16:7–11. “In one embodiment, one or more PEs are transferred between the stream processing cluster 515 and the static process 525 via the SP-MR containers *including data, code, configuration, etc.*” *Id.* at 16:34–37 (emphasis added). Thus, at this stage, we see no reason to conclude that it would be inconsistent with the specification or nonsensical for a processing element that is data to be part of a processing mechanism, as Patent Owner argues. *See* Prelim. Resp. 21.

Patent Owner also argues that the challenged patent’s specification describes the processing elements as encapsulated computational chains, and “by virtue of their name,” computational chains are related to performing computations on data—they are not mere data. *Id.* at 25 (citing Ex. 1001, 4:31–33). But the chain’s parameters and state can be part of the chains themselves. *See* Ex. 1001, 17:17–24 (discussing “updated parameters”). Also, the processing elements can be code, which could represent the

computational aspects of the chains. *See id.* (discussing “computational codes”).

Patent Owner argues that the challenged patent’s specification refers to processing elements as examples of processors and as including executable code. Prelim. Resp. 25–26 (citing Ex. 1001, 12:17–18, 7:36–38). But the same parts cited by Patent Owner confirm that the “computational codes” are one example of an updated processing element. *See* Ex. 1001, 17:17–24. The challenged patent gives other examples of updated processing elements, such as “updated parameters.” *Id.* at 17:20–24. This is consistent with the rest of the challenged patent, which explains that a processing element is not limited to code. *Id.* at 6:63–64; *see also id.* at 13:16–21.

iii. Other Claims

“Other claims of the patent in question . . . can also be valuable sources of enlightenment as to the meaning of a claim term.” *Phillips*, 415 F.3d at 1314 (citing *Vitronics Corp. v. Conceptronic Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 2003)). Here, Patent Owner argues that Petitioner did not challenge claims 6 or 16, which require a processing element that carries out processing: “wherein the updating of the at least one data object comprises, at least in part, causing a shaping of the one or more computational chains based, at least in part, on the one or more triggering events, the one or more contextual trends, or a combination thereof.” Prelim. Resp. 39–40 (quoting Ex. 1001, 26:8–12 (claim 6), 27:21–25 (claim 16)). In Patent Owner’s view, “Petitioner fails to challenge these claims because Foster fails to disclose or otherwise suggest any computational chain or other ‘element that carries out processing’ being transferred from Streams to BigInsights.” *Id.* at 40.

As discussed above, the challenged patent discloses that a processing element is not limited to code or elements that carry out processing. *See*

Section II.B.1.c.ii *supra*. Even assuming the subject matter of claims 6 and 16 requires an element that carries out processing, that does not mean independent claims 1 and 11 are limited to those embodiments. In fact, under Patent Owner’s interpretation, the processing elements of claims 1 and 11 would be limited to only those that carry out processing, which would render the language in claims 6 and 16 meaningless. That is typically disfavored. *See Ortho-McNeil Pharm. v. Mylan Labs, Inc.*, 520 F.3d 1358, 1362 (Fed. Cir. 2008) (“[T]his court strives to reach a claim construction that does not render claim language in dependent claims meaningless.”). Thus, if we were to agree with Patent’s Owner’s view of the dependent claims 6 and 16, it would only further support Petitioner’s proposed construction of processing element in claims 1 and 11. *See* Prelim. Resp. 39–40.

For the same reasons, we also disagree with Patent Owner’s argument that dependent claims 2, 9, 12, and 19 confirm that a processing element must be a component that carries out data processing. *Id.* at 22–23.

For example, Patent Owner argues that claims 2 and 12 recite “a processing of at least substantially real-time data via the at least one processing element to determine one or more triggering events.” *Id.* at 22. But the specification explains that the processing element can be code or data. *See, e.g.*, Ex. 1001, 13:16–17. So even if claim 2 encompasses at least one embodiment in which the code is used to process the real-time data, claim 1 can be broader than claim 2. In this way, claim 1 is not limited to the embodiment expressly recited in claim 2.

Similarly, Patent Owner argues that “data alone does not ‘operate’ at any frequency,” yet claims 9 and 19 require “wherein the at least one dynamic processing mechanism, the at least one processing element, or a

combination thereof operate, at least in part, in substantially real-time or at a frequency greater [than]³ the predetermined batch frequency.”

Prelim. Resp. 22–23 (citing Ex. 1001, 13:40–46). Patent Owner makes a similar argument for claims 10 and 20 with regard to performing “fast moving data analytics.” *Id.* at 23–24.

But claims 9, 10, 19, and 20 recite that a processing mechanism or, alternatively, a processing element operates at the recited frequency and performs the analytics. *See* Ex. 1001, 26:29 (claim 9), 26:36 (claim 10), 28:16 (claim 19), 28:23 (claim 20). Because the processing mechanism and processing elements are listed as alternatives, we disagree that claims 9, 10, 19, and 20 require a processing element to operate as recited.

Petitioner’s analysis of Foster is consistent with this interpretation of claims 9, 10, 19, and 20. For example, Petitioner argues that “Foster teaches the static processing mechanism operates . . . in a batch mode” because BigInsights is a static-processing mechanism that operates in batch mode and uses a Hadoop’s MapReduce framework. Pet. 72–73 (citing Ex. 1004, 338–39, 343; Ex. 1003 ¶¶ 176–180; Ex. 1012 § 1; Ex. 1001, 5:4–11, 16:26–34) (emphasis removed, alteration in original). Here, Petitioner’s analysis relies on the alternative involving the processing mechanism. *See id.*

Even in the embodiments in which a processing element could be code, the challenged patent explains that the code itself is in some ways data that describes an executable program. For example, a processing element could be binary code. *See* Ex. 1001, 7:36–37. The processing element can then be marshalled, resulting in a data object with a “snap shot of the state of

³ Claims 9 and 19 recite “greater than,” but Patent Owner claims this is a typo. Prelim. Resp. 23 n.2. For this Decision, we adopt Patent Owner’s interpretation.

the process” or “values of variables.” *Id.* at 7:36–43. Thus, the challenged patent’s specification better aligns with Petitioner’s proposed construction, which encompasses all the different types of processing elements described in the challenged patent: data, code, parameters, and other things.

iv. Extrinsic Evidence

Patent Owner argues that Petitioner has failed to show that the patentee expressed an intent to redefine the term that would be necessary to justify a construction that deviates from the plain and ordinary meaning of “processing element.” Prelim. Resp. 30–31 (citing *Thorner v. Sony Computer Ent. Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012); *K-fee Sys. GmbH v. Nespresso USA, Inc.*, 89 F.4th 915, 923 (Fed. Cir. 2023)). Patent Owner relies on the extrinsic record to support its position, arguing that Foster supports the well-established meaning of “processing element.” *Id.* at 26. According to Patent Owner, Foster describes a processing element (1) as a custom shared library that can be deployed and (2) as a process running the database adapter’s code. *Id.* at 26–27 (citing Ex. 1004, 63, 89, 375).

But extrinsic evidence is “less reliable” and “is unlikely to result in a reliable interpretation of patent claim scope unless considered in the context of the intrinsic evidence.” *Phillips*, 415 F.3d at 1318–19. Here, the specification consistently uses the term “processing element” to mean data or code. *See, e.g.*, Ex. 1001, 6:63–64; *id.* at 13:16–21. Interpreting “processing element” in the way that Foster uses the term would exclude the embodiments in which the processing element is data. Claims are not normally interpreted “in a way that excludes embodiments disclosed in the specification.” *Oatey Co. v. IPS Corp.*, 514 F.3d 1271, 1276 (Fed. Cir. 2008). Also, Patent Owner’s extrinsic evidence is inconsistent with the challenged patent (Prelim. Resp. 20), which explains that processing

elements can be data and not code (*see, e.g.*, Ex. 1001, 6:63–7:1). By contrast, Petitioner’s proposed construction (Pet. 19) is more consistent with the entire patent, which describes various types of processing elements. *See Phillips*, 415 F.3d at 1313 (explaining that it is important to read a claim in the context of the entire patent).

Patent Owner argues that the claimed marshalling does not require the recited processing element to be data because marshalling is not unique to data. Prelim. Resp. 28. According to Patent Owner, it was well known to marshal components used for processing, such as models or code. *Id.* But Patent Owner’s argument here suggests that Petitioner reads the claim as excluding code. *Id.* at 29–30 (citing Pet. 19). Not so. Petitioner acknowledges that processing elements can be either “*code and/or data.*” *See* Pet. 19 (quoting Ex. 1001, 6:63–7:1) (emphasis in original). Likewise, in the Foster-Williams combination, Petitioner asserts that “Williams teaches marshalling a processing element (e.g., data, code, or both) as a data object.” *Id.* at 34 (emphasis removed). These statements indicate that Petitioner views the claim as encompassing both data and code (*id.*), contrary to Patent Owner’s argument (Prelim. Resp. 28–30).

v. *Petitioner’s Expert*

Patent Owner argues that Mark E. Crovella’s testimony should be given little to no weight in connection with the proposed claim construction because it merely repeats the Petition’s argument. Prelim. Resp. 31–32 (citing Pet. 19; Ex. 1003 ¶¶ 26–28). We disagree. The testimony may be entitled to little weight if it restates the petition’s unsupported, conclusory assertions without adding supporting evidence or technical reasoning, for example. *Xerox Corp. v. Bytemark, Inc.*, IPR2022-00624, Paper 9 at 15 (PTAB Aug. 24, 2022) (precedential). By contrast, the evidence here

adequately supports the Petition’s argument (Pet. 19) and the corresponding testimony (Ex. 1003 ¶¶ 26–28) for all the reasons discussed above.

vi. *Conclusion*

Thus, at this stage, we construe processing element to mean “data to be processed” (Pet. 19), including data, binary code, configuration, parameters, variables, process state, among other things (*see, e.g.*, Ex. 1001, 7:36–43; *id.* at 13:16–17; *id.* at 16:34–37). By adopting this construction, we disagree with Patent Owner that the claimed processing elements should be limited to one of those embodiments, such as code that “carries out data processing.” Prelim. Resp. 20.

2. *Dynamic and Static Processing Mechanisms*

According to Petitioner, “[d]ynamic processing mechanism’ means a ‘mechanism for processing data in real-time or near real-time’” (Pet. 20), and “[s]tatic processing mechanism’ means a ‘mechanism for processing historical or slow-moving data’” (*id.* at 21).⁴

Apart from the arguments discussed above, Patent Owner does not present any argument specifically directed to Petitioner’s constructions of these mechanisms or propose alternative constructions. *See* Prelim. Resp.

At this stage, we adopt Petitioner’s proposed constructions. Pet. 20–21. Those constructions are consistent with the challenge patent’s specification. For example, the challenged patent explains that dynamic processing mechanisms operate in real-time (Ex. 1001, 17:15–17), receiving data from the feeds:

⁴ Petitioner’s definition of “static processing mechanism” repeats the phrase “for processing.” Pet. 21. We treat this as a typographical error and alter the quotation here and throughout this decision. *Id.*

In one embodiment, the dynamic processing mechanism 601 *receives real-time and/or near real-time data* from one or more data feeds 605 (e.g., one or more data and/or service providers), where the data is processed by a stream processing engine 607 (e.g., stream processing framework, context mapping algorithms and data-structures, etc.), wherein processing results may be kept (e.g., stored at an intermediate component such as database 115) *and/or output in real-time (e.g., stream) to one or more applications*.

Id. at 16:48–56, *cited in* Pet. 20 (emphasis added). By contrast, static processing mechanism 603 can operate in a batch-mode with a predetermined batch frequency, such as daily or weekly. *Id.* at 17:12–17. Thus, one of the primary differences between the mechanisms is their operating frequencies. *Id.* at 17:15–17.

In this decision, we construe “dynamic processing mechanism” as “a mechanism for processing data in real-time or near real-time,” and “static processing mechanism” as a “mechanism for processing historical or slow-moving data.” *See* Pet. 20–21.

C. *Obviousness over Foster Alone and Foster with Williams*

Petitioner asserts that claims 1–5, 7–15, and 17–20 are unpatentable as obvious over Foster alone and Foster with Williams. Pet. 14. Patent Owner’s arguments focus on claims 1 and 11. *See* Prelim. Resp. 33–42. Patent Owner does not provide a separate argument for the remaining claims. *See id.* at 42–43. Instead, Patent Owner argues that Petitioner’s challenges to the remaining claims are deficient for the same reasons as given in connection with the challenges to claims 1 and 11. *Id.* Thus, the guidance that follows focuses on claims 1 and 11.

1. *Foster*

Foster relates to big-data analysis using IBM's InfoSphere Streams and BigInsights. Ex. 1004, 37.⁵ According to Foster, standard databases offer static models and data. *Id.* at 56. In these databases, data persists on a hard disk, and queries are executed against historical data. *Id.* By contrast, a Streams environment does not need to store data before analysis. *Id.* at 57. Instead, the environment ingests, filters, analyzes, and correlates data “while it is in motion.” *Id.* For example, in a Streams environment, data is selected as it flows past the application, which sends the selected data to a consumer. *Id.* at 58. In fact, Streams can work with data volumes so large that it would be impractical to make them persistent. *Id.* at 57. Also, Streams applications can answer always-on or continuous types of questions: What is the rolling average? How many parts have been built so far today? How does production at this point in time today compare with yesterday? *Id.* at 58.

2. *Williams*

Williams discusses encodings used in distributed-computing environments. Ex. 1005, 39. Of relevance to Petitioner's challenge, Williams describes marshalling and unmarshalling. *Id.* According to Williams, data values can be defined in an application, and encodings define how those data values can be translated to and from a protocol format. *Id.* The translation steps are called serialization and deserialization, which is synonymous with marshalling and unmarshalling. *Id.* For example, a SOAP⁶ encoding enables

⁵ The pages of Exhibits 1004 and 1005 are numbered using six-digits with leading zeros—e.g., 000037. This decision's citations omit the leading zeros.

⁶ According to Williams, the authors of SOAP 1.1 declared that SOAP stands for Simple Object Access Protocol, but the authors of SOAP 1.2 did not describe any such meaning. Ex. 1005, 32.

marshalling and unmarshalling of data-type values from the SOAP data model. *Id.*

3. Claims 1 and 11

a. Processing Element

Claim 1 recites, in part, “determining via a processor at least one *processing element* of at least one dynamic processing mechanism.”

Ex. 1001, 25:34–38 (emphasis and numbering added). Claim 11 recites a similar limitation. *Id.* at 26:45–48. The parties refer to these limitations as [1a] and [11b]. *See, e.g.*, Pet. 81, 84; Prelim. Resp. 33.

According to Petitioner, Foster’s Streams sends data to BigInsights, and this data is a processing element “because it is earmarked to be further processed (e.g., bootstrapped and enriched) at BigInsights.” Pet. 24 (citing Ex. 1004, 340).

Petitioner relies on the embodiment shown in Foster’s Figure 7-1, below. *Id.* at 26 (citing Ex. 1004, 339, Fig. 7-1).

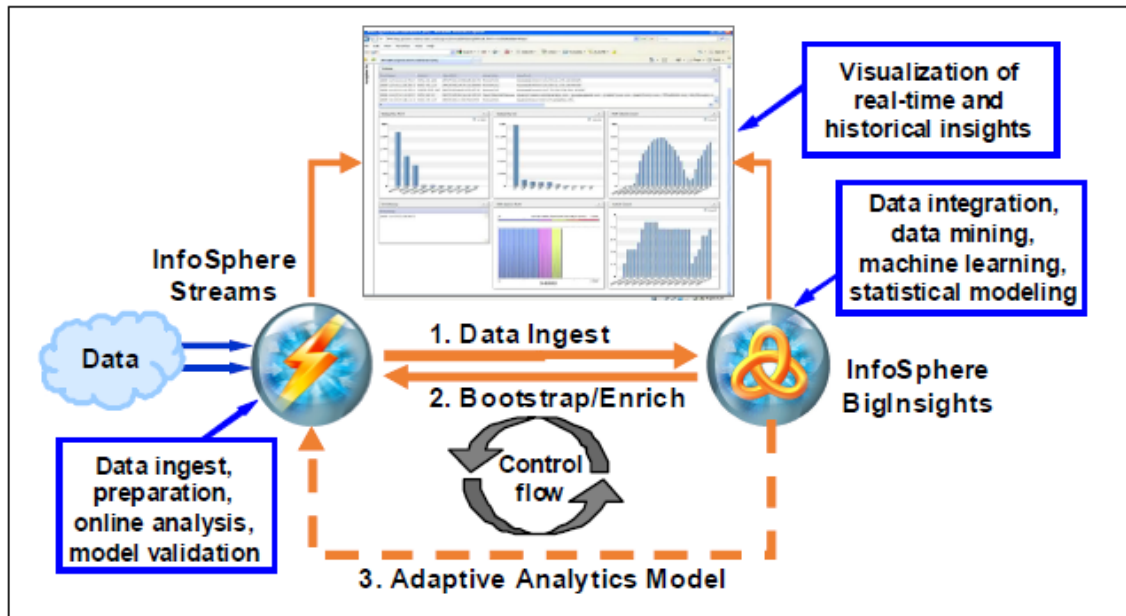


Figure 7-1 Big Data application scenarios

Foster's Figure 7-1 shows the integration of data-in-motion (InfoSphere Streams) and data-at-rest (BigInsights) platforms for three main scenarios. Ex. 1004, 338. In the first scenario (orange arrow labeled "1. Data Ingest"), data is continuously ingested through Streams into BigInsights. *Id.* In the second scenario (orange arrow labeled "2. Bootstrap/Enrich"), historical context generated from BigInsights is used to bootstrap analytics and enrich incoming data on Streams. *Id.* at 339. In the third scenario (orange arrow labeled "3. Adaptive Analytics Model"), data mining, machine learning, and statistical modeling on BigInsights generate models, and these models form the basis for analytics on incoming data on Streams. *Id.* The top of the figure shows how real-time and historical insights are visualized as reports. *Id.*

Petitioner argues that the reports in Figure 7-1 show that real-time data ingestion in Streams contributes to the data-in-motion (current data) and data-at-rest (historical data) analysis. Pet. 25 (citing Ex. 1004, 338–39, 341; Ex. 1003 ¶ 58). In Petitioner's view, Foster displays real-time insights. *Id.* at 26 (citing Ex. 1004, 339). Petitioner argues that Streams and its data-ingest function is a "dynamic processing mechanism," as claimed, because Streams processes data in real-time. *Id.* at 24.

At this stage, we agree with Petitioner's argument. *See id.* Specifically, we preliminarily construe "dynamic processing mechanism" as "a mechanism for processing data in real-time or near real-time." *See* Section II.B.2 *supra*. And according to Foster, Streams analyzes and acts upon data "with low latency in close to or in real time." Ex. 1004, 56–57, *quoted in* Pet. 25. Thus, Petitioner has shown that Streams is a "dynamic processing mechanism." Pet. 24–25.

As for the mechanism's processing element, we disagree with Patent Owner's proposed construction for "processing element" (*see* Section II.B.1.

supra) and the arguments based on that proposed construction (Prelim. Resp. 33–35). For example, Patent Owner argues that Foster does not disclose a processing element that carries out processing (Prelim. Resp. 33) or that includes code or another computational element (*id.* at 34). But the claimed processing element is not limited in this way under the construction adopted in this decision. Rather, we construe processing element to mean “data to be processed.” *See* Section II.B.1 *supra*. Thus, we disagree with Patent Owner’s arguments about the claimed processing element. Prelim. Resp. 33–35.

b. Marshalling

Claim 1 recites, in part, “causing, at least in part, a *marshalling* via the processor of the at least one processing element as at least one data object.” Ex. 1001, 25:36–38 (emphasis added). Claim 11 recites a similar limitation. *Id.* at 26:47–48. The parties refer to these limitations as [1b] and [11c]. *See, e.g.*, Pet. 81, 84; Prelim. Resp. 35.

i. Foster Alone

Petitioner argues that “Streams marshals the input data by processing it in various methods to convert its memory representation into a form that BigInsights can use.” Pet. 28 (citing Ex. 1004, 319). Petitioner also argues that Foster transforms streams of data records, represented in memory as tuples, to JSON for storage in BigInsights, and Streams supports sink operators to marshal data into other output formats. *Id.* at 30 (citing Ex. 1004, 64, 95–97, 343–44, 346). According to Petitioner, these transformations create a data object, as claimed. *Id.* at 31 (citing Ex. 1001, 4:29–47).

Patent Owner argues that Petitioner cited Foster’s Figure 2-3 showing a processing element as an application running on application hosts, but

Petitioner does not rely on that processing element to teach the one claimed. Prelim. Resp. 34. In Patent Owner’s view, Petitioner does this because there is no evidence that Foster marshals the processing element shown in Figure 2-3, as limitation [1b] and [11c] require. *Id.*

We disagree with this argument because, here, we construe processing element to mean “data to be processed.” *See* Section II.B.1 *supra*. Under our preliminary construction, Petitioner’s analysis is sufficient for institution because Petitioner explains how Foster transforms data. *See* Pet. 28–31. In fact, Patent Owner does not dispute that Foster does this. *See* Prelim. Resp. 34. Rather, Patent Owner argues that marshalling data is not the same as marshalling “at least one processing element.” *Id.* at 35–36. For example, Patent Owner argues that Foster’s tuple data is not a processing element, so converting the tuple data to JSON format is not the recited marshalling. *Id.* at 35 (citing Pet. 30). Patent Owner also argues that, even if the sink operators in Streams marshal data into other output formats, this does not read on the claimed marshalling because the processing units are not data alone. *Id.* at 35–36 (citing Pet. 30–31). In sum, we disagree with Patent Owner’s arguments (*id.*) because they do not align with our preliminary construction (*see* Section II.B.1 *supra*).

In addition to the transformation to JSON, Petitioner argues that Foster also discloses sending information over a TCP network connection. Pet. 32. According to Petitioner,

A POSITA would have understood that Foster’s disclosure of Streams sending information to BigInsights via, e.g., TCP connection teaches marshalling the data before it is sent over the connection because it was well-known that sending data via TCP involves marshalling the data into the TCP packet format.

Id. at 33 (citing Ex. 1003 ¶ 70).

As for this argument, Patent Owner argues that Petitioner’s reasoning in the grounds based on Foster alone is deficient because it relies on an unsupported inherency theory. Prelim. Resp. 36 (citing Pet. 14, 22, 32–33). Alternatively, Patent Owner argues that, as an obviousness theory, Petitioner’s analysis lacks the requisite factual findings. *Id.* at 37 (citing *Prometheus Labs., Inc. v. Roxane Labs., Inc.*, 805 F.3d 1092, 1097–98 (Fed. Cir. 2015)).

We disagree with Patent Owner’s characterization of Petitioner’s rationale. In the TCP-connection rationale, Petitioner argues that a person of ordinary skill in the art would have understood that a TCP connection involves marshaling data, not that it would have been obvious to do so. Pet. 33. Notably, Patent Owner has not provided evidence or identified anything on the current record that indicates Petitioner’s assertion about TCP connections is incorrect. *See* Prelim. Resp. 36. Because Petitioner’s reasoning based on Foster’s JSON format is sufficient, we need not reach the issue of whether TCP connections marshal data. Rather, this is a factual issue best resolved on a record fully developed during trial.

ii. Foster and Williams

Petitioner alternatively argues that the Foster-Williams combination teaches the marshalling recited in limitations [1b] and [11c]. Pet. 33–40. According to Petitioner, Williams teaches marshalling data, code, or both as a data object. *Id.* at 34 (citing Ex. 1005, 39). Petitioner argues that it would have been obvious to one of ordinary skill in the art to marshal Foster’s data processed by Streams (the processing elements) into a portable form as at least one data object to be transferred to and used by other processes or applications, such as BigInsights. *Id.* at 35. Petitioner provides multiple

reasons why one of ordinary skill in the art would have done so. *See id.* at 35–40.

Patent Owner argues that Petitioner relies on a single passage from Williams, which “merely describes marshalling at a high-level of generality, and does not disclose or otherwise suggest marshalling of an object such as code that is capable of performing processing (as opposed to marshalling of mere data).” Prelim. Resp. 41–42. In Patent Owner’s view, Williams’s teachings do not remedy Foster’s alleged deficiencies. *Id.*

We disagree with Patent Owner’s arguments (*id.*) because they do not align with our preliminary construction (*see* Section II.B.1 *supra*). At this stage, Petitioner shows that Williams discloses marshalling data and code, and the Petition contains a sufficient rationale for combining Williams with Foster. *See* Pet. 33–40.

c. Data Object

Claim 1 recites, in part,

[1c] transferring of the at least one data object to at least one static processing mechanism;

[1d] processing of the at least one data object by the at least one static processing mechanism; and

[1e] after processing, transferring the at least one data object back to the at least one dynamic processing mechanism.

Ex. 1001, 25:39–44 (numbering from the Petition added); *see* Pet. 81. Claim 11 recites similar limitations, which the parties refer to as [11d], [11e], and [11f]. Ex. 1001, 26:49–54; Pet. 84; Prelim. Resp. 38.

Petitioner argues that the data transferred from Streams to BigInsights—either in the JSON or TCP format, for example—corresponds to the data object. *See* Pet. 40 (citing Ex. 1004, 339); *see also* Section II.C.3.b *supra* (discussing the marshaling process in [1b] and [11c]).

According to Petitioner, BigInsights corresponds to the static processing mechanism because it processes historical or slow-moving data (Pet. 42), and Streams is the dynamic processing mechanism (*id.* at 24). According to Petitioner, Foster’s data object is transferred from Streams to BigInsights, as in [1c] (*id.* at 40), BigInsights processes the data object, as in [1d], by bootstrapping and enriching the data (*id.* at 43), and BigInsights transfers the bootstrapped-enriched data back to Streams, as in [1e] (*id.* at 46).

Patent Owner argues that Foster lacks the claimed data object. Prelim. Resp. 38–39. According to Patent Owner, the claims require that the data object is formed from a marshalled processor element, but Streams transfers data to BigInsights, not any processing elements. *Id.* (citing Ex. 1001, 25:36–38; Ex. 1005, 339; Pet. 28, 43). Patent Owner argues that the data marshalled in Streams is not a processing element under its proposed construction. *Id.* at 39.

We disagree with Patent Owner’s arguments for the reasons discussed in Section II.C.3.b. In that section, we explain why Petitioner has sufficiently shown that Foster and the Foster-Williams combination teach or suggest the marshalled data object under the Board’s preliminary construction of “processing element.” In sum, we disagree with Patent Owner’s argument (Prelim. Resp. 38–39) because we construe processing element to mean “data to be processed,” which is not limited to the specific embodiments that are the basis for Patent Owner’s proposed construction. *See* Section II.B.1 *supra*. Apart from the argument about unchallenged dependent claims 6 and 16 addressed in Section II.B.1.c.iii of this decision (Prelim. Resp. 39–40), Patent Owner does not provide further arguments about limitations [1c] through [1e] and [11d] through [11f].

Under our preliminary constructions for “processing element” and “static processing mechanism” (*see* Section II.B), Petitioner has sufficiently shown that Foster and the Foster-Williams combination teach or suggest [1c] through [1e] and [11d] through [11f]. Specifically, we preliminary construe “static processing mechanism” as a “mechanism for processing historical or slow-moving data.” *See* Section II.B.2 *supra*. Petitioner has shown that BigInsights is a static processing mechanism for at least the reasons that BigInsights is a “data-at-rest” platform (Ex. 1004, 338, *cited in* Pet. 42), generates historical context (*id.* at 339, *cited in* Pet. 42), and carries out deep analysis “to detect patterns in data collected over a long period of time” (*id.* at 341, *cited in* Pet. 42). Foster’s Figure 7-1 shows three main scenarios in which the data is transferred to and from Streams and BigInsights. *See* Section II.C.3.a *supra* (discussing the scenarios in Figure 7-1).

Thus, Petitioner has shown that Foster and the Foster-Williams combination teach or suggest “[1c] transferring of the at least one data object to at least one static processing mechanism; [1d] processing of the at least one data object by the at least one static processing mechanism; and [1e] after processing, transferring the at least one data object back to the at least one dynamic processing mechanism,” and the corresponding subject matter recited in limitations [11d] through [11f] of claim 11.

d. Conclusion

Thus, Petitioner is reasonably likely to show that claims 1 and 11 are unpatentable. Because Petitioner has shown a reasonable likelihood of prevailing as to the unpatentability of at least one of the challenged claims, we institute an *inter partes* review on all challenged claims and on all grounds in the Petition.

Although the above analysis states that we disagree with Patent Owner's arguments or agree with Petitioner's, a trial could produce additional evidence that may support a determination in the Final Written Decision that unpatentability has not been adequately proven. Everything in this decision should be read in that context and should not be understood to be a final determination.

III. ORDER

It is

ORDERED that, under 35 U.S.C. § 314(a), an *inter partes* review of claims 1–5, 7–15, and 17–20 of the '693 patent is instituted for all grounds in the Petition; and

FURTHER ORDERED that, under 35 U.S.C. § 314(a), an *inter partes* review of the '693 patent is instituted on this Decision's entry date, and under 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is given of the trial's institution.

IPR2024-01507
Patent 8,996,693 B2

FOR PETITIONER:

Harper Batts
Christopher Ponder
Jeffrey Liang
SHEPPARD, MULLIN, RICHTER & HAMPTON LLP
hbatts@sheppardmullin.com
cponder@sheppardmullin.com
jliang@sheppardmullin.com

FOR PATENT OWNER:

Scott Hejny
Eric Hansen
MCKOOL SMITH, P.C.
shejny@mckoolsmith.com
ehansen@mckoolsmith.com