

**IN THE UNITED STATES PATENT AND TRADEMARK OFFICE**

In re Patent of: Hoover et al.  
U.S. Patent No.: 8,370,416 Attorney Docket No.: 39898-0049IP1  
Issue Date: February 5, 2013  
Appl. Serial No.: 11/411,732  
Filing Date: April 26, 2006  
Title: COMPATIBILITY ENFORCEMENT IN CLUSTERED  
COMPUTING SYSTEMS

**Mail Stop Patent Board**

Patent Trial and Appeal Board  
U.S. Patent and Trademark Office  
P.O. Box 1450  
Alexandria, VA 22313-1450

**PETITION FOR *INTER PARTES* REVIEW OF UNITED STATES PATENT  
NO. 8,370,416 PURSUANT TO 35 U.S.C. §§ 311–319, 37 C.F.R. § 42**

**TABLE OF CONTENTS**

|      |                                                              |    |
|------|--------------------------------------------------------------|----|
| I.   | REQUIREMENTS FOR IPR—37 C.F.R. §42.104.....                  | 1  |
| A.   | Grounds for Standing—37 C.F.R. §42.104(a).....               | 1  |
| B.   | Challenge and Relief Requested—37 C.F.R. §42.104(b) .....    | 1  |
| II.  | THE '416 PATENT .....                                        | 2  |
| A.   | Pertinent Disclosure .....                                   | 2  |
| B.   | File History .....                                           | 6  |
| III. | LEVEL OF ORDINARY SKILL .....                                | 7  |
| IV.  | CLAIM CONSTRUCTION—37 C.F.R. §42.104(b)(3) .....             | 8  |
| V.   | THE CHALLENGED CLAIMS ARE UNPATENTABLE.....                  | 8  |
| A.   | GROUND 1: Claims 1–18 Are Obvious Over Sparks and Bram ..... | 8  |
| 1.   | Sparks (FEDEX-1005) .....                                    | 8  |
| 2.   | Bram (FEDEX-1006) .....                                      | 13 |
| 3.   | Sparks-Bram Combination .....                                | 15 |
| 4.   | Claims 1, 14.....                                            | 21 |
| 5.   | Claims 2, 15.....                                            | 43 |
| 6.   | Claim 3 .....                                                | 46 |
| 7.   | Claim 4 .....                                                | 46 |
| 8.   | Claims 5, 16.....                                            | 50 |
| 9.   | Claims 6, 17.....                                            | 54 |
| 10.  | Claims 7, 18.....                                            | 58 |
| 11.  | Claim 8 .....                                                | 60 |

|      |                                                      |    |
|------|------------------------------------------------------|----|
| 12.  | Claim 9 .....                                        | 61 |
| 13.  | Claim 10 .....                                       | 67 |
| 14.  | Claim 11 .....                                       | 69 |
| 15.  | Claim 12 .....                                       | 70 |
| 16.  | Claim 13 .....                                       | 71 |
| VI.  | PAYMENT OF FEES—37 C.F.R. §42.103 .....              | 73 |
| VII. | MANDATORY NOTICES—37 C.F.R. § 42.8(a)(1) .....       | 74 |
| A.   | Real Party-In-Interest—37 C.F.R. § 42.8(b)(1).....   | 74 |
| B.   | Related Matters—37 C.F.R. § 42.8(b)(2) .....         | 74 |
| C.   | Lead And Back-Up Counsel—37 C.F.R. § 42.8(b)(3)..... | 74 |
| D.   | Service Information .....                            | 75 |

**LIST OF EXHIBITS**

|            |                                                                                                                                                                                                                                                                                                               |
|------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| FEDEX-1001 | U.S. Patent 8,370,416 to Hoover et al. (“the ’416 Patent”)                                                                                                                                                                                                                                                    |
| FEDEX-1002 | Excerpts from the File History of the ’416 Patent (“the File History”)                                                                                                                                                                                                                                        |
| FEDEX-1003 | Declaration of Dr. Chandrajit Bajaj                                                                                                                                                                                                                                                                           |
| FEDEX-1004 | Curriculum Vitae of Dr. Chandrajit Bajaj                                                                                                                                                                                                                                                                      |
| FEDEX-1005 | U.S. Patent 7,590,653 to Sparks (“Sparks”)                                                                                                                                                                                                                                                                    |
| FEDEX-1006 | U.S. Patent App. Pub. 2006/0179058 to Bram (“Bram”)                                                                                                                                                                                                                                                           |
| FEDEX-1007 | Resources for Business, Business Software Alliance, 2004<br>retrieved from <a href="https://web.archive.org/web/20040612091712/http://www.bsa.org/usa/antipiracy/Resources-for-Business.cfm">https://web.archive.org/web/20040612091712/<br/>http://www.bsa.org/usa/antipiracy/Resources-for-Business.cfm</a> |
| FEDEX-1008 | Piracy and The Law, Business Software Alliance, 2004<br>retrieved from <a href="https://web.archive.org/web/20040614221833/http://www.bsa.org/usa/antipiracy/Piracy-and-the-Law.cfm">https://web.archive.org/web/20040614221833/<br/>http://www.bsa.org/usa/antipiracy/Piracy-and-the-Law.cfm</a>             |
| FEDEX-1009 | Microsoft Computer Dictionary, Fifth Edition, 2002 (excerpts)                                                                                                                                                                                                                                                 |
| FEDEX-1010 | The New Penguin Dictionary of Computing, 2001 (excerpts)                                                                                                                                                                                                                                                      |
| FEDEX-1011 | FLEXlm End Users Guide, Version 9.5, August 2004                                                                                                                                                                                                                                                              |
| FEDEX-1012 | Administering Platform LSF, Version 5.1, February 2003                                                                                                                                                                                                                                                        |

**LISTING OF CLAIMS**

| Claim 1 |                                                                                                                                                                                                                                                                               |
|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [1pre]  | A method of creating a clustered computing system, comprising:                                                                                                                                                                                                                |
| [1a-i]  | storing license information for a computing cluster in a memory module associated with a computing node,                                                                                                                                                                      |
| [1a-ii] | wherein the licensing information includes a bundle-type parameter that identifies a size of the cluster and at least one additional bundle-type characteristic of the cluster, and one or more node license parameters that identifies a characteristic of a computing node; |
| [1b]    | initializing the computing cluster in a first computing node;                                                                                                                                                                                                                 |
| [1c]    | adding one or more available computing nodes to the computing cluster; and                                                                                                                                                                                                    |
| [1d]    | activating the computing cluster when the computing cluster includes a number of nodes that complies with the bundle-type parameter.                                                                                                                                          |
| Claim 2 |                                                                                                                                                                                                                                                                               |
| [2]     | The method of claim 1, wherein storing license information for a computing cluster in a memory module associated with a computing node comprises retrieving licensing information from an external licensing server.                                                          |
| Claim 3 |                                                                                                                                                                                                                                                                               |
| [3]     | The method of claim 1, wherein the bundle-type parameter identifies at least one of a minimum or maximum node count threshold for the computing cluster.                                                                                                                      |
| Claim 4 |                                                                                                                                                                                                                                                                               |
| [4]     | The method of claim 1, wherein: the node license parameters identify one or more bundle-types compatible with the computing node.                                                                                                                                             |

| Claim 5  |                                                                                                                                                                                                                                                                                                        |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [5]      | The method of claim 1, wherein adding one or more available computing nodes to the computing cluster comprises assigning a computing node a new node license when the computing node license is incompatible with a brand-type associated with the computing cluster.                                  |
| Claim 6  |                                                                                                                                                                                                                                                                                                        |
| [6]      | The method of claim 1, further comprising initiating a license check while the computing system is operating, wherein the license check evaluates at least one of: a license status of the computing node; a node count in the computing cluster; and a brand-type associated with the computing node. |
| Claim 7  |                                                                                                                                                                                                                                                                                                        |
| [7]      | The method of claim 1, wherein the license check comprises implementing an error routine that inhibits operation of the computing node when an incompatibility between the computing node and the computing cluster is detected.                                                                       |
| Claim 8  |                                                                                                                                                                                                                                                                                                        |
| [8]      | The method of claim 1, further comprising adding a computing node to the activated computing cluster.                                                                                                                                                                                                  |
| Claim 9  |                                                                                                                                                                                                                                                                                                        |
| [9pre]   | A computing system comprising at least two computing nodes, wherein at least one of the computing nodes comprises:                                                                                                                                                                                     |
| [9a-i]   | a cluster module stored on a non-transitory computer readable medium, the cluster module configured to manage compatibility operations associated with a computing cluster,                                                                                                                            |
| [9a-ii]  | the cluster module initializing a computing cluster in the first computing node,                                                                                                                                                                                                                       |
| [9a-iii] | the cluster module adding one or more available computing nodes to the computing cluster;                                                                                                                                                                                                              |

|                 |                                                                                                                                                                                                                                                                                                             |
|-----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [9b-i]          | a node module stored on a non-transitory computer readable medium, the node module configured to manage compatibility operations associated with the at least two computer nodes,                                                                                                                           |
| [9b-ii]         | the node module storing license information for the computing cluster,                                                                                                                                                                                                                                      |
| [9b-iii]        | wherein the licensing information includes a bundle-type parameter that identifies a size of the computing cluster and at least one additional bundle-type characteristic of the computing cluster, and one or more node license parameters that identifies a characteristic of a first computing node; and |
| [9c-i]          | a management module stored on a non-transitory computer readable medium, the management module configured to manage compatibility operations with one or more external servers,                                                                                                                             |
| [9c-ii]         | the management module activating the computing cluster when the computing cluster includes a number of nodes that complies with the bundle-type parameter.                                                                                                                                                  |
| <b>Claim 10</b> |                                                                                                                                                                                                                                                                                                             |
| [10]            | The computing system of claim 9, wherein the cluster module comprises a create cluster module to aggregate the at least two computing nodes into a clustered computing system.                                                                                                                              |
| <b>Claim 11</b> |                                                                                                                                                                                                                                                                                                             |
| [11]            | The computing system of claim 10, wherein the cluster module maintains a right to use license comprising licensing information for a computing cluster and a node license pool comprising licensing information for nodes in the computing cluster.                                                         |
| <b>Claim 12</b> |                                                                                                                                                                                                                                                                                                             |
| [12]            | The computing system of claim 10, further comprising an add node module to add a computing node to the clustered computing system.                                                                                                                                                                          |

| Claim 13 |                                                                                                                                                                                                                                                                               |
|----------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [13]     | The computing system of claim 10, wherein the management module provides a license key to and receives licensing information from an external licensing server.                                                                                                               |
| Claim 14 |                                                                                                                                                                                                                                                                               |
| [14pre]  | A computer program product comprising logic instructions stored on a non-transitory computer-readable medium which, when executed by a computer processor, configure the processor to:                                                                                        |
| [14a-i]  | store license information for a computing cluster in a memory module associated with a computing node,                                                                                                                                                                        |
| [14a-ii] | wherein the licensing information includes a bundle-type parameter that identifies a size of the cluster and at least one additional bundle-type characteristic of the cluster, and one or more node license parameters that identifies a characteristic of a computing node; |
| [14b]    | initialize the computing cluster in a first computing node;                                                                                                                                                                                                                   |
| [14c]    | add one or more available computing nodes to the computing cluster;<br>and                                                                                                                                                                                                    |
| [14d]    | activate the computing cluster when the computing cluster includes a number of nodes that complies with the bundle-type parameter.                                                                                                                                            |
| Claim 15 |                                                                                                                                                                                                                                                                               |
| [15]     | The computer program product of claim 14, further comprising logic instructions which, when executed by the processor, configure the processor to retrieve licensing information from an external licensing server.                                                           |

| Claim 16 |                                                                                                                                                                                                                                                                                                                                                                                                              |
|----------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| [16]     | The computer program product of claim 14, further comprising logic instructions which, when executed by the processor, configure the processor to assign a computing node a new node license when the computing node license is incompatible with a brand-type associated with the computing cluster.                                                                                                        |
| Claim 17 |                                                                                                                                                                                                                                                                                                                                                                                                              |
| [17]     | The computer program product of claim 14, further comprising logic instructions which, when executed by the processor, configure the processor to initiate a license check while the computing system is operating, wherein the license check evaluates at least one of: a license status of the computing node; a node count in the computing cluster; and a brand-type associated with the computing node. |
| Claim 18 |                                                                                                                                                                                                                                                                                                                                                                                                              |
| [18]     | The computer program product of claim 14, further comprising logic instructions which, when executed by the processor, configure the processor to implement an error routine that inhibits operation of the computing node when an incompatibility between the computing node and the computing cluster is detected.                                                                                         |

FedEx Corporation and FedEx Office and Print Services, Inc. petition for IPR of claims 1–18 (“Challenged Claims”) of U.S. Patent 8,370,416.

**I. REQUIREMENTS FOR IPR—37 C.F.R. §42.104**

**A. Grounds for Standing—37 C.F.R. §42.104(a)**

Petitioners certify that the ’416 Patent is available for IPR, and that Petitioner is not barred or estopped from requesting IPR.

**B. Challenge and Relief Requested—37 C.F.R. §42.104(b)**

Petitioners request IPR and cancellation of the ’416 Patent’s Challenged Claims based on the ground below. Petitioners submit that the cited prior art references are themselves sufficient to establish obviousness pursuant to the analysis in Section V. The Section V analysis further cites to the Declaration of Dr. Bajaj for complementary corroboration and explanation. *See* FEDEX-1003.

| Asserted Grounds |        |                   |
|------------------|--------|-------------------|
| ID               | Claims | Prior Art Basis   |
| 1                | 1–18   | §103: Sparks-Bram |

As shown below, each of the above identified reference pre-dates April 26, 2006 (“Critical Date”), which is the earliest date to which the ’416 Patent claims priority. However, Petitioners do not concede that the ’416 Patent is entitled to the claimed priority.

| Reference                               | Prior Art Date (at least as early as) /<br>Pre-AIA Basis <sup>1</sup> |
|-----------------------------------------|-----------------------------------------------------------------------|
| Sparks<br>US 7,590,653<br>(FEDEX-1005)  | Filed, March 3, 2005—§102(e)                                          |
| Bram<br>US 2006/0179058<br>(FEDEX-1006) | Filed, November 30, 2005—§102(e)<br>(at least as early as)            |

## II. THE '416 PATENT

### A. Pertinent Disclosure

The '416 Patent, entitled “COMPATIBILITY ENFORCEMENT IN CLUSTERED COMPUTING SYSTEMS,” issued on February 5, 2013 from U.S. Patent Application No. 11/411,732 (“the '732 application”), which was filed on April 26, 2006. FEDEX-1001, Cover.

The '416 Patent states that it “relates to electronic computing, and more particularly to compatibility enforcement in clustered computing systems.” *Id.*, 1:6-8; FEDEX-1003, ¶¶37-41. The '416 Patent describes that “[c]omputing systems and products created using clustering technology present particular technical issues.” FEDEX-1001, 1:23-29. These “[t]echnical issues include ensuring compatibility between hardware, software and firmware operating on clustered devices and

---

<sup>1</sup> See §§V.A.1, V.A.2 for explanations of these dates.

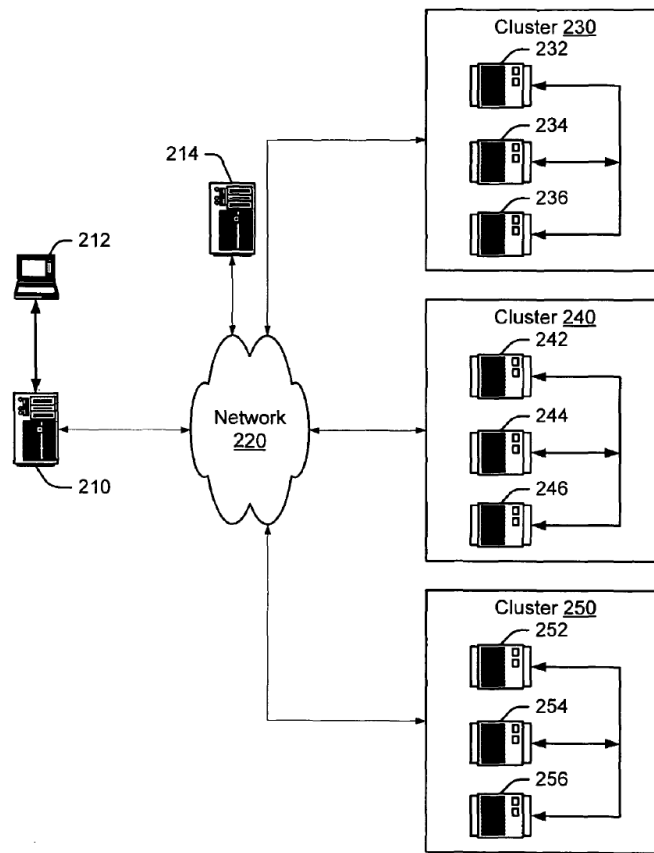
ensuring compliance with licensing terms associated with individual computing devices and with cluster configurations.” *Id.*

The ’416 Patent describes a clustered computing system “may be adapted to implement compatibility enforcement between components in the system” including “compatibility between hardware, software, firmware, and/or licensing requirements.” *Id.*, 5:40-6:3. Accordingly, the cluster implements a “licensing scheme [which] may implement a multi-tiered (or bundled) license program.” The ’416 Patent describes that:

The first part of the license scheme defines a permissible size range for a cluster. In one embodiment, clusters may be categorized into multiple size ranges, referred to herein as bundle-types. In alternate embodiments, clusters may be categorized into bundle types by one or more additional parameters such as, e.g., an array, tape, computing, translating functional bundle. For example, a bundle may refer to a sales bundle. A bundle may be sold as a set or group of node or grid types to meet a specific market or function.

*Id.*, 5:50-58. The ’416 Patent explains that “[b]undles may have a specific homogeneous purpose along with standard grid parameters such as, e.g., right to use, maximum and minimum node counts, and a specific array type functional brand.” *Id.*, 5:61-64.

A clustered computing system can include one or more clusters, where each cluster includes one or more nodes. *Id.*, 4:39-5:8.



FEDEX-1001, Fig. 2. The '416 Patent explains that "[t]he term 'clustered computing system' refers to a computing system formed from a group of loosely coupled computing devices that cooperate to provide the appearance that the devices function as a single system." FEDEX-1001, 1:11-22.

A technique for creating a clustered computing system disclosed by the '416 Patent, includes (1) "storing license information for a computing cluster in a memory module associated with a computing node," (2) "initializing the computing cluster in a first computing node," (3) "adding one or more available computing nodes to the computing cluster," and (4) "activating the computing cluster when the

computing cluster includes a number of nodes that complies with the bundle-type parameter.” *Id.*, Abstract. The stored “licensing information includes a bundle-type parameter that identifies a characteristic of the cluster and one or more node license parameters that identifies a characteristic of a computing node.” Figure 5 depicts “operations in a method for creating an initial clustered computing system.” *Id.*, 6:55-8:23.

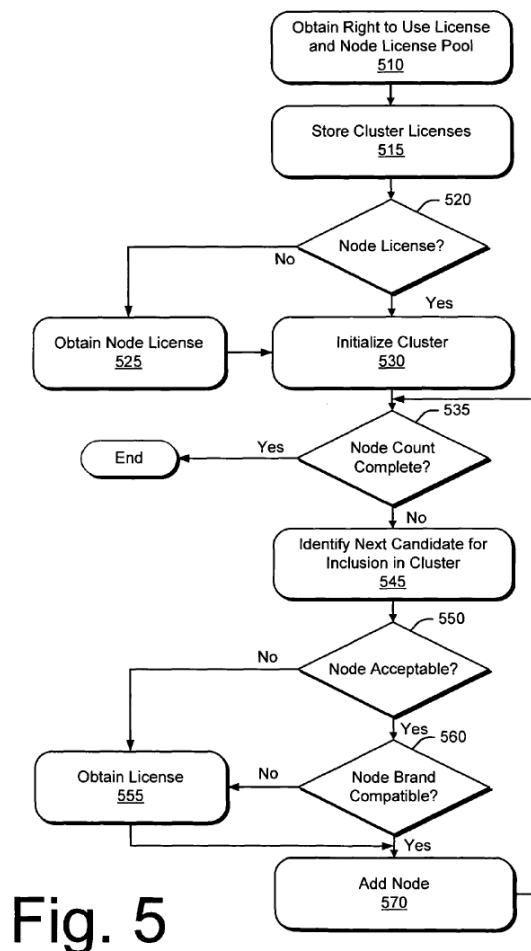


Fig. 5

FEDEX-1001, Fig. 5.

## **B. File History**

The '732 Application, which later issued as the '416 Patent, was filed on April 26, 2006. FEDEX-1001, Cover. The Examiner initially rejected the pending claims over Bauer (US 2006/0036894) or a combination of Bauer and Yellai (US 2006/0106728). FEDEX-1002, pp. 620-630. In response, the Applicant (Hewlett-Packard Company) made minor amendments and provided arguments attempting to distinguish the claims over Bauer. *Id.*, pp. 602-615. But the Examiner maintained the prior art rejections. *Id.*, pp. 586-599. So the Applicant filed another response with a request for continued examination. *Id.*, pp. 458-471.

The Examiner issued a new non-final rejection, which continued the rejections over Bauer or in combination with Yellai. *Id.*, pp. 357-369. This time, the Applicant amended the claims by adding, to each of the independent claims, “a bundle-type parameter that identifies a size of the cluster and at least one additional bundle-type characteristic of the cluster.” *Id.*, pp. 337-352 (amendment underlined). This amendment prompted the Examiner to issue a final rejection over Bauer and a new reference, Davidson (US 2005/0234846). *Id.*, pp. 321-333. The Examiner stated that “Davidson teaches a bundle-type parameter that identifies a size of the computer cluster.” *Id.* In combination with Bauer and Davidson, the Examiner continued to apply Yellai for a subset of the dependent claims. *Id.*

Facing the final rejections, and unwilling to amend the claims to distinguish the new Bauer-Davidson combination, the Applicant filed an appeal brief to the Board. *Id.*, pp. 287-312. However, the Board never adjudicated the appeal because, surprisingly, the Examiner abruptly issued a notice of allowance. *Id.*, pp. 262-274.

Had the Examiner remained diligent and submitted an answer to the Applicant's appeal brief, the Board would have had an opportunity to review—and potentially affirm—the standing rejections. Additionally, had the Examiner performed a more thorough search before issuing the notice of allowance, he would have discovered pertinent references like Sparks (FEDEX-1005) and Bram (FEDEX-1006) that make an even stronger case of obviousness against the Challenged Claims, including the “bundle-type” limitations. *See id.*, 278-275 (time stamps on “2012/11/06” from “17:20” to “17:45”); *see also infra* §V.A.4, [1a-i] and [1d]; FEDEX-1003, ¶¶42-48. Either way, the Challenged Claims should not have issued in their unjustifiably broad form.

### **III. LEVEL OF ORDINARY SKILL**

A person of ordinary skill in the art as of the Critical Date of the '416 Patent (“POSITA”) would have had (1) at least a master's of science degree in computer science, computer engineering, or a related field, and (2) at least two years of professional experience involving the study and/or development of distributed,

clustered, and/or grid computer systems. FEDEX-1003, ¶¶28-30. Additional education could substitute for experience and vice versa. *Id.*

#### **IV. CLAIM CONSTRUCTION—37 C.F.R. §42.104(b)(3)**

The analysis below in Section V proves the Challenged Claims obvious under any interpretation reasonably advanced according to the *Phillips* standard. Petitioners submit that the Board need not resolve any express constructions. *See Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011). Should any subsequent discussion of claim interpretation arise here or in the parallel district court proceeding, Petitioners reserve the right to address and respond to any construction advanced by Patent Owner, the Board, and/or the District Court.

#### **V. THE CHALLENGED CLAIMS ARE UNPATENTABLE**

##### **A. GROUND 1: Claims 1–18 Are Obvious Over Sparks and Bram**

##### **1. Sparks (FEDEX-1005)**

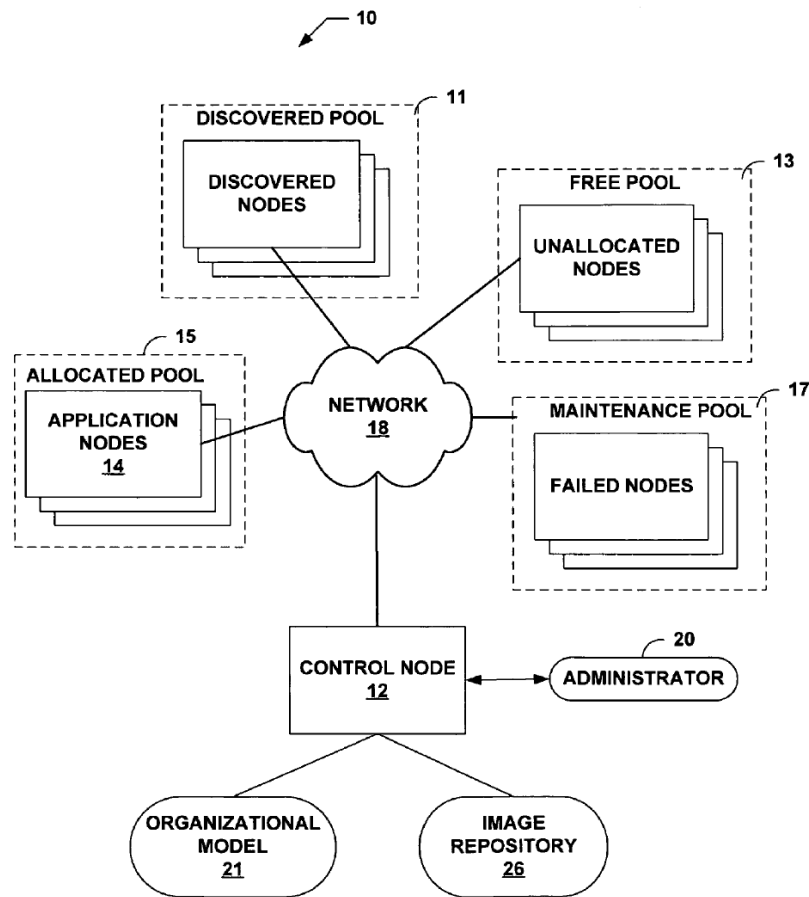
Sparks, entitled “AUTOMATED DISCOVERY AND INVENTORY OF NODES WITHIN AN AUTONOMIC DISTRIBUTED COMPUTING SYSTEM,” issued on September 15, 2009 from a patent application filed on March 2, 2005. FEDEX-1005, Cover. Accordingly, Sparks qualifies as prior art under at least pre-AIA § 102(e).

Like the '416 Patent, Sparks discloses a clustered computing system and techniques for its management. FEDEX-1003, ¶¶49-55.<sup>2</sup> For example, Sparks describes a “distributed computing system that conforms to a multi-level, hierarchical organizational model” where “[o]ne or more control nodes provide for the efficient and automated allocation and management of computing functions and resources within the distributed computing system in accordance with the organization model.” FEDEX-1005, 1:33–50. “[T]he model includes four distinct levels: fabric, domains, tiers and nodes that provide for the logical abstraction and containment of the physical components as well as system and service application software of the enterprise.” *Id.*

Sparks further describes that a “[d]istributed computing system 10 may be viewed as a collection of computing nodes operating in cooperation with each other to provide high-performance processing.” FEDEX-1005, 3:7-11. These nodes can be organized into one or more tiers. *Id.*, 5:23-67. Sparks explains that “[a] tier is a logically associated group of fabric components within a domain that share a set of attributes: usage, availability model or business service mission.” *Id.*, 5:48–58.

---

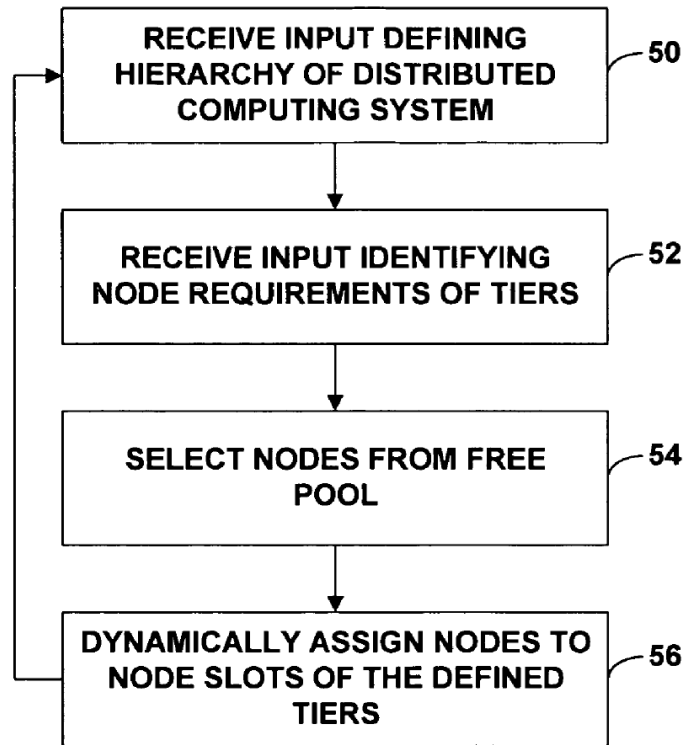
<sup>2</sup> General descriptions provided for this and other references and combinations thereof are incorporated into each subsection and mapping of the claims that includes citations to these references.



FEDEX-1005, Fig. 1. As discussed further below, a POSITA would have recognized that Sparks’s disclosure regarding the distributed computing system 10 describes a clustered computing system, and also that each tier (including its computing nodes) is likewise itself a computing cluster within the greater system. FEDEX-1003, ¶52.

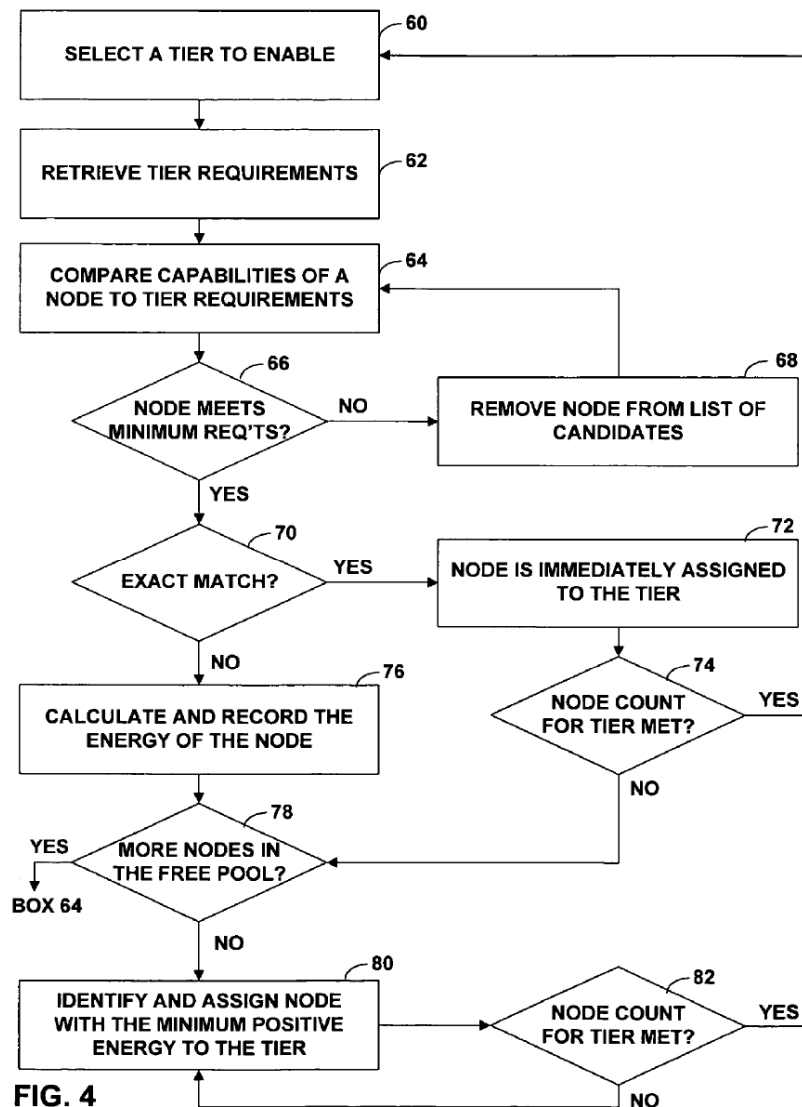
According to Sparks, a system administrator “may define policies regarding tier characteristics” and “specific node requirements for each tier of the fabric.” FEDEX-1005, 6:42-67. Sparks further describes that the “[c]ontrol node 12 dynamically assigns computing nodes to the node slots of each tier based on the node requirements specified for the tiers [by the administrator] and the identified node

attributes (56).” *Id.*, 9:3-13; *see also* 8:37-39 (Fig. 3 shows “operation of control node 12 when configuring distributed computing system 10”).



FEDEX-1005, Fig. 3.

Figure 4 of Sparks (below) provides details of adding nodes to a tier, and in general, tier enablement. For example, Sparks describes that the control node selects and enables a tier. FEDEX-1005, 9:14-21 (discussing that “control node 12 selects a tier to enable (60)” and “[c]ontrol node 12 may, for example, initially select the tier with the highest priority and successively enable the tiers based on priority”).



FEDEX-1005, Fig. 4. Sparks further discloses that the control node “determines whether the node count for the tier is met (74).” FEDEX-1005, 9:52-62. “When the node count for the tier is met, control node 12 selects the next tier to enable, ... and repeats the process until all defined tiers are enabled, i.e., populated with application nodes (60).” Sparks additionally discloses how “a user may request that a newly created tier come online” via enabling the tier. *Id.*, 18:45–50.

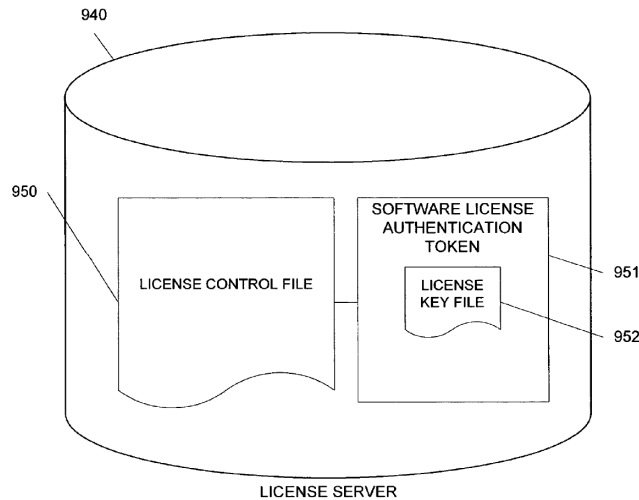
## **2. Bram (FEDEX-1006)**

Bram, entitled “METHODS AND SYSTEMS FOR LICENSING COMPUTER SOFTWARE,” published on August 10, 2006 from a U.S. patent application filed on November 30, 2005. FEDEX-1006, Cover. Accordingly, Bram qualifies as prior art under at least pre-AIA § 102(e).

Like the '416 Patent, Bram discloses a clustered computing system and techniques for its management. FEDEX-1003, ¶¶56-59. Specifically, Bram describes a system “designed to run on one or more clusters of various types of computers, including heterogeneous clusters made up of different types of computers running different operating systems.” FEDEX-1006, [0232] (“Each individual computer is called a ‘node.’”). In this clustered-computing context, Bram provides “a method of determining whether a component defined in a computer program is authorized to activate,” which includes “determining if the component is associated with at least one required license.” *Id.*, [0004].

In Bram’s system, users “obtain licenses that allow them to run particular applications 16 or components of the system 10 (e.g., the applications 16 and components of the system 10 that the system user has purchased a license for).” FEDEX-1006, [0244]. “To manage and apply the licenses,” the system includes “a license server” having a “license control file.” *Id.* The license control file “includes a license file defining a base license or license package that allows core components

of the system...to operate and one or more license files defining supplemental licenses that allow additional components of the system...to operate.” *Id.*



FEDEX-1006, Fig. 28.

Bram’s license server further includes a “software licensing authentication token” with a “license key file.” FEDEX-1006, [0245]. The license key file “provides authorization to apply the available licenses if constraints of the available licenses are satisfied.” *Id.* Accordingly, the key file “can specify parameters and/or constraints for each available license, such as an effective date, an expiration date, a total concurrent activations limit or count, a list of authorized named nodes, a list of authorized operating systems (or operating system aliases), a list of authorized MAC codes, a list of authorized JVMs, and/or an activation mode (e.g., ‘restricted’ or ‘unrestricted’) of components allowed under a license.” *Id.*

### **3. Sparks-Bram Combination**

The Sparks-Bram Combination applies Bram's teachings about licensing enforcement to Sparks's distributed computing system. FEDEX-1003, ¶60. It would have been obvious to a POSITA to modify Sparks's distributed computing system based on Bram's disclosures on licensing, such that the modified system applies and tracks licenses as appropriate for various system components to ensure compliance with license requirements. *Id.* For example, it would have been obvious to a POSITA to include the functionality provided by Bram's licensing server in the distributed computing system of Sparks. *Id.* As further explained below, a POSITA would have been motivated to incorporate the teachings of Bram into Sparks's system with a reasonable expectation of success. *Id.*

#### **(a) Motivation**

As discussed above at Sections V.A.1-2, Sparks and Bram each pertain to clustered computing systems and their management. FEDEX-1005, 1:7-8, 1:33-39; FEDEX-1006, [0004], [0232]; FEDEX-1003, ¶61. It is therefore no surprise that a POSITA would have been motivated to pursue the Sparks-Bram Combination for at least the following reasons (taken individually or collectively). FEDEX-1003, ¶61.

**First**, a POSITA equipped with basic common sense and ordinary creativity would have been motivated to seek out teachings from references like Bram to incorporate licensing support in Sparks's distributed computing system based on the

teachings and suggestions within Sparks. FEDEX-1003, ¶62 (also explaining that Bram's license controls use of a known licensing mechanism to monitor requirements/constraints that Sparks already tracks). Sparks's system includes "[o]ne or more control nodes [that] provide for the efficient and automated allocation and management of computing functions and resources within the distributed computing system in accordance with the organization model." FEDEX-1005, 1:33-39. As discussed above, "[a] tier [within the model/system] is a logically associated group of fabric components within a domain that share a set of attributes: usage, availability model or business service mission." FEDEX-1005, 5:48-58. Sparks describes that "each node slot [of a tier] represents a data set that describes specific information for a corresponding node, such as software resources for a physical node that is assigned to the node slot." FEDEX-1005, 6:22-30. It was well-known before the Critical Date that software resources typically require a license. FEDEX-1003, ¶62; FEDEX-1007; FEDEX-1008. Moreover, license managers/servers were also well-known including those for cluster-based deployments. FEDEX-1003, ¶62; FEDEX-1011, p. 11 ("FLEXlm is the most popular license manager used in the software industry"); FEDEX-1012, p. 14 (discussing "Platform LSF" for "cluster administrators"), p. 231 ("[s]oftware licenses are valuable resources that must be fully utilized"; [t]his section discusses how LSF can help manage licensed applications to maximize utilization and minimize job failure due to license

problems”). Sparks describes that a system administrator “may define policies regarding tier characteristics” as well as “specific node requirements for each tier of the fabric.” FEDEX-1005, 6:42-67. A POSITA would have understood that licensing is another common requirement to be configured for one or more nodes/tiers in Sparks. FEDEX-1003, ¶62.

**Second**, it would have been obvious to a POSITA—and the POSITA would have been motivated—to incorporate in a distributed and clustered computing system like Sparks’s a means to obtain the appropriate license(s) in order to operate a tier, including its nodes, in accordance with prescribed license terms. FEDEX-1003, ¶¶63-66. For example, Bram discloses that “[c]omputer software providers often desire to control access to and activation of their computer software” and “depending on functionality purchased, authorized, and/or needed by a specific consumer, the computer software provider can design specific computer software in order prevent the consumer from obtaining un-purchased, un-authorized, and/or un-needed functionality.” FEDEX-1006, [0002]; *see also id.*, [0003]. As the POSITA would have understood, such means can include a local license storage and/or license server. FEDEX-1006, [0244] (“the system 10 can include a license server”), [0245] (“Using the license key file 952, the license server 940 can authorize activation of applications 16 and/or components associated with available licenses and can disable (e.g., prevent activation) of applications 16 and/or components

associated with unavailable licenses or available licenses that have unsatisfied constraints”), [0247].

Thus, from the perspective of a POSITA, it would have been obvious to incorporate licensing capabilities in the distributed/clustered computing system of Sparks to prevent the use of “un-purchased, un-authorized, and/or un-needed functionality,” just as Bram described. FEDEX-1003, ¶¶63-66 (citing to FEDEX-1006, [0002]). It would have likewise been obvious to do so to “track applied licenses” within the system. *Id.* (citing to FEDEX-1006, [0255], [0268]).

**Third**, it was well publicized that the unlicensed use of computer systems and/or programs may be unlawful or result in civil penalties. Industry groups, such as the Business Software Alliance, widely warned that companies could be held liable for employees’ unauthorized software installations, face civil damages, and even criminal penalties. *See, e.g.*, FEDEX-1007 (“[M]any businesses, even without knowing it, face potential disaster because of software piracy[.]”); FEDEX-1008 (“Piracy and the Law”). Accordingly, it would have been obvious to a POSITA—and a POSITA would have been motivated—to include licensing capabilities in a computing system so that its users do not run afoul of laws or incur civil penalties. FEDEX-1003, ¶67. Integrating license based controls was therefore a straightforward, low-cost way (relative to the expensive fines/damages) to reduce legal exposure and ensure compliant use across clusters. *Id.*

**Fourth**, a POSITA would have been led to the Sparks-Bram Combination because the licensing techniques that improved Bram's clustered computing system in terms of license management and enforcement would likewise be expected to improve the clustered system of Sparks in the same way. FEDEX-1003, ¶68; *see supra* §§V.A.1-2. In short, the POSITA would have been motivated to account for license requirements and restrictions, as taught by Bram, when implementing the technique Sparks teaches for setting up and activating the tiers of a distributed/clustered computing system. *Id.* Indeed, the POSITA would have realized that the same improvements offered by the license management and enforcement techniques in Bram's system would be realizable in other systems, including the one that Sparks describes. *Id.*

**Fifth**, a POSITA would have been led to the Sparks-Bram Combination because it involves the application of a known technique (Bram's licensing technique) to a known system (the clustered/distributed computing system of Sparks) to achieve predictable results (a clustered/distributed computing system having licensing capabilities). FEDEX-1003, ¶69. As the Federal Circuit has made clear: "It's enough for [the Petition] to show that there was a known problem of [licensing in clustered computing system], that [Bram's technique] helped address that issue, and that combining the teachings of [Sparks and Bram] wasn't beyond the skill of an ordinary artisan. Nothing more is required to show a motivation to

combine under *KSR*.” *Intel Corp. v. PACT XPP Schweiz AG*, 61 F.4th 1373, 1380-81, (Fed. Cir. 2023).

**(b) Reasonable Expectation of Success**

Given Sparks’s disclosure of various node and tier requirements in the context of a distributed and clustered computing system (e.g., FEDEX-1005, 5:48–58, 6:22-30, 6:42-67), turning to a reference such as Bram to further incorporate licensing requirements in this same context would have been straightforward from a POSITA’s perspective. FEDEX-1003, ¶¶70-72; *see supra* §V.A.1-2. The Sparks-Bram Combination represents the integration of known elements, each performing the functions it had been known to perform to produce predictable results. *Id.* The Sparks-Bram Combination does not change the key aspects of either reference. *Id.* The crux of the combination simply involves adding functionality to Sparks’s system based on the licensing disclosures of Bram. *Id.* The respective teachings would work in combination just as they did apart. *Id.*

Further, integrating Bram’s licensing techniques into the system of Sparks would have been a routine task for a POSITA. FEDEX-1003, ¶72. Bram and Sparks are related to cluster computing systems and their management. FEDEX-1005, 1:7-8, 1:33-39; FEDEX-1006, [0004], [0232]; FEDEX-1003, ¶72. Given their similarities, it would have been straightforward to modify the control node of Sparks to implement the functionality of and/or communicate with a licensing server like

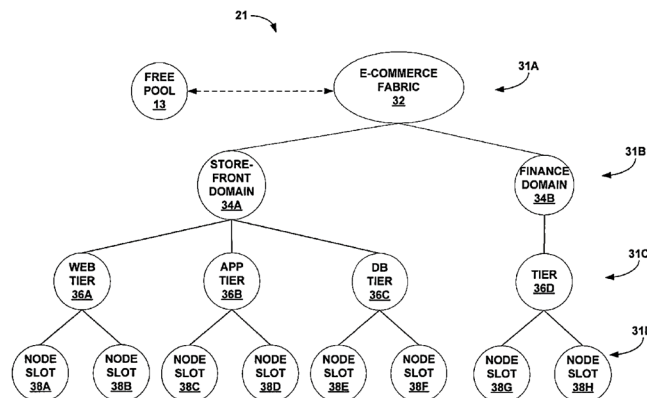
the one disclosed by Bram. *Id.* As such, a POSITA would have reasonably expected a successful outcome from combining Bram’s teachings on licensing techniques and systems with Sparks’s description of a distributed computing system. *Id.*

#### 4. Claims 1, 14

##### (a) Claim 1

**[1pre] A method of creating a clustered computing system, comprising:**

To the extent the preamble is limiting, Sparks-Bram renders obvious [1pre]. FEDEX-1003, ¶¶73-77. For example, Sparks describes methods for creating “a distributed computing system that conforms to a multi-level, hierarchical organization model.” FEDEX-1005, 1:33–50. As shown below, this model “includes four distinct levels: fabric, domains, tiers and nodes.” *Id.*; *see also id.*, 1:51-2:7, 5:23–28. “A tier is a logically associated group of fabric components within a domain that share a set of attributes: usage, availability model or business service mission.” *Id.*, 5:48–58.



FEDEX-1005, Fig. 2; *see also id.* 7:26-8:36.

Sparks goes on to explain that the “[d]istributed computing system 10 may be viewed as a collection of computing nodes operating in cooperation with each other to provide high-performance processing.” FEDEX-1005, 3:7-11; *see also id.*, Fig. 1. As described above at Section V.A.1 and below at [1a-i], [1a-ii], [1b], [1c], and [1d], among the nodes of Sparks’s distributed computing system is a control node that assigns other constituent computing nodes to the node slots within separate tiers.

A POSITA would have recognized that Sparks discloses, or at least renders obvious, a clustered computing system. FEDEX-1003, ¶76. For one, it was known to those of skill in the art that a distributed computing system is an example of a clustered computing system. FEDEX-1003, ¶76 (explaining that the computing nodes that make up a clustered computing system can be “distributed”); FEDEX-1009 (see definitions of distributed system, distributed processing, cluster, and clustering); FEDEX-1010 (see definitions of distributed, distributed processing, distributed system, cluster).

Moreover, as would have been apparent, or obvious, to a POSITA, each tier (including its nodes) in Sparks is likewise itself a computing cluster within the greater distributed system. FEDEX-1003, ¶76. This understanding, and Sparks’s description, is consistent with the ’416 Patent’s description that “[t]he term ‘clustered computing system’ refers to a computing system formed from a group of loosely coupled computing devices that cooperate to provide the appearance that the

devices function as a single system.” FEDEX-1001, 1:11-22; *see also id.*, 4:39-5:8, Fig. 2; FEDEX-1003, ¶76.

Bram similarly describes that a system can be “designed to run on *one or more clusters of various types of computers*, including heterogeneous clusters made up of different types of computers running different operating systems” and explains that “[e]ach individual computer is called a ‘node.’” FEDEX-1006, [0232]; FEDEX-1003, ¶77. Bram’s confirmatory teachings reinforce how a POSITA would have understood Sparks’s above-discussed similar disclosures regarding a collection of coordinated computing nodes making up a distributed and clustered computing system, leaving no doubt that the Sparks-Bram combination provides the claimed limitation of “creating a clustered computing system.” FEDEX-1006, [0232], [0252], [0304]; FEDEX-1003, ¶77.

**[1a-i] storing license information for a computing cluster in a memory module associated with a computing node,**

Sparks-Bram renders obvious [1a-i]. FEDEX-1003, ¶¶78-86. To start, and as established above at [1pre], the combined teachings of Sparks and Bram provide a clustered computing system (*e.g.*, Sparks’s distributed computing system 10) including a collection of computing nodes, including a control node, where each tier of nodes within the greater system likewise provides a cluster. As explained below,

Sparks and Bram further provide the claimed step of “storing license information for a computing cluster in a memory module associated with a computing node.”

*storing information in a memory module associated with a computing node*

Sparks describes “administration software executing on control node 12 to logically define the hierarchical organization of distributed computing system 10,” and further explains that “a model for the enterprise [] includes a number of domains, tiers, and node slots hierarchically arranged within a single enterprise fabric.” FEDEX-1005, 6:1-9. The control node “maintain[s] a database having entries for each node slot, where the entries identify the node requirements for each of the tiers” and “retrieves the node requirements for the selected tier from the database.” FEDEX-1005, 9:22-27; *see also id.*, 1:51-56 (discussing “***updating a model to store the attributes*** for the detected node, wherein the model defines an organization of a distributed computing system having a plurality of computing nodes”), 23:15-20 (“A computer-implemented method comprising: ***storing a model within a database of a control node***, wherein the model defines an organization of a distributed computing system having a plurality of computing nodes”).

Sparks makes clear that the computing nodes—including the control node and the application nodes—have a memory module to store information. FEDEX-1005, 3:7-25 (discussing that “a distributed computing system 10 [is] constructed from a collection of computing nodes,” which “includes at least one control node 12”),

3:49-59 (discussing that a node includes “computing resources (e.g., processors and *memory*)”), Fig. 1; FEDEX-1003, ¶¶81-82. Accordingly, a POSITA would have understood, or it would have been obvious to a POSITA, that information for Sparks’s distributed computing system, including its model and requirements for each tier and each node thereof, is stored by the control node in a memory module (e.g., memory containing some or all of the information from Sparks’s above-discussed database). FEDEX-1003, ¶¶83. As one example, Sparks teaches that a control node’s rule engine may correspond to a tier, and that it stores expected state information (e.g., max nodes) in its working *memory*. FEDEX-1005, 16:36-40, 17:1-7, 17:65-18:2, 18:47-50, 14:11-14, Figs. 17, 18; FEDEX-1003, ¶¶83-84 (discussing this and other examples).

*storing license information*

Bram’s disclosure, incorporated with Sparks as discussed above at Section V.A.3, provides for obtaining and storing *licensing information* such as a license control file, license key file, and token. FEDEX-1006, [0232], [0244], [0245], [0282], [0005]; FEDEX-1003, ¶¶85. Bram’s system is “designed to run on *one or more clusters of various types of computers*.” FEDEX-1006, [0232]. In this cluster-computing context, Bram explains that system users “obtain licenses that allow them to run particular applications 16 or components of the system 10.” *Id.*, [0244]. Accordingly, Bram’s system includes a license server, which has a license control

file that “can be used to map activation authorization requests generated by applications 16 and/or components of the system 10 to licenses that are required in order for the license server 940 to authorize the activation of the applications 16 and/or components.” *Id.* The license server “includes a software licensing authentication token (‘token’) 951.” *Id.*, [0245]. As Bram explains, “the token 951 can include a license key file 952 that defines the licenses available to a particular system user” and “provides authorization to apply the available licenses if constraints of the available licenses are satisfied by the current state of the system 10.” *Id.* Bram’s system components can also obtain and store updated licensing information, such as “updated license control file 950, an updated token 951, and/or an updated license key file 952.” FEDEX-1006, [0282].

As discussed at Section V.A.3, a POSITA would have been motivated to account for license requirements, as taught by Bram, when implementing the techniques Sparks teaches for setting up and activating one or more tiers of a distributed computing system. FEDEX-1003, ¶86. It would have been obvious to a POSITA to store license information, such as that disclosed by Bram, as part of the above-discussed information stored in the memory module of a node for a tier (computing cluster) in Sparks so that each node and/or the number of nodes of a tier comport with license requirements. FEDEX-1003, ¶86; §V.A.3. Additional details

regarding the stored information, including license information, are provided below in [1a-ii].

**[1a-ii] wherein the licensing information includes a bundle-type parameter that identifies a size of the cluster and at least one additional bundle-type characteristic of the cluster, and one or more node license parameters that identifies a characteristic of a computing node;**

Sparks-Bram renders obvious [1a-ii]. FEDEX-1003, ¶¶87-99.

“wherein the licensing information includes a bundle-type parameter that identifies a size of the cluster and at least one additional bundle-type characteristic of the cluster”

As discussed, Sparks-Bram provides licensing information.<sup>3</sup> See §V.A.4, [1a-i]. Further, Sparks describes a computing node—a control node—that provides a user interface to define *properties of a tier (computing cluster)*, as explained above for [1pre] and [1a-i]. FEDEX-1005, 6:1-9 (“control node 12 to logically define the hierarchical organization of distributed computing system 10”), 6:42–54 (“define specific node requirements for each tier of the fabric,” *i.e.*, “creates a list of attributes that the tier requires of its candidate computing nodes”), 9:22-32 (discussing “the node requirements associated with the selected tier”), 7:1-14 (discussing same), 11:50–12:5, 12:19-24, Figs. 7-8.

---

<sup>3</sup> A POSITA would have understood that “the licensing information” in [1a-ii] refers to “license information” in [1a-i]. FEDEX-1003, ¶88.

One of the tier properties *identifies a size of the tier (computing cluster)*. FEDEX-1003, ¶90; FEDEX-1005, 6:55-67 (“System administrator 20 may define policies regarding tier characteristics, such as a *minimum number of nodes a tier requires*, ...”), 6:22-30 (“control node 12 creates a number of ‘node slots’ equal to the *maximum number of application nodes* 14”; “each node slot represents a data set that describes specific information for a corresponding node”), 9:52-62, 10:15-35 (“Control node 12 determines whether the *minimum node count for the tier* is met (82)”).

A POSITA would have understood that a tier property such as a “minimum number of nodes” (or “maximum number”) connotes a *bundle-type parameter* because this number defines a size requirement for a *tier (computing cluster)*. FEDEX-1003, ¶91. This understanding is consistent with the ’416 Patent which discloses “clusters may be categorized into multiple size ranges, referred to herein as bundle-types” and that “[b]undles may have a specific homogeneous purpose along with standard grid parameters such as, e.g., [] maximum and minimum node counts.” FEDEX-1001, 5:48-64; FEDEX-1003, ¶91.

Another tier property identifies *at least one additional bundle-type characteristic of the tier (cluster)*. FEDEX-1003, ¶92. For example, Sparks discloses that tier properties include “a name for the tier, a description of the tier, and a image associated with the tier.” FEDEX-1005, 11:66-12:3. Additional node

requirements for a tier can include “a central processing unit (CPU) count, a CPU speed, an amount of memory (e.g., RAM), local disk characteristics and other hardware characteristics.” FEDEX-1005, 6:42–54. Tier properties can include a common functional aspect or attribute such as “usage, availability model or business service mission.” FEDEX-1005, 5:48–58, 6:10-21 (“System administrator 20 further defines one or more tiers within each domain that represent the functional subcomponents of applications and services provided by the domain,” e.g., “define a storefront domain within the enterprise fabric that includes a web tier, an application tier and a database tier.”). A POSITA would have understood that these additional properties (taken individually or collectively) provide a ***bundle-type characteristic*** because each additional property defines an operative aspect for a ***tier (computing cluster)***. FEDEX-1003, ¶¶93-94. This understanding is consistent with the ’416 Patent, which discloses that “clusters may be categorized into bundle types by one or more additional parameters such as, e.g., an array, tape, computing, translating functional bundle” and that “[a] bundle may be sold as a set or group of node or grid types to meet a specific market or function.” FEDEX-1001, 5:48-64; FEDEX-1003, ¶¶93-94.

“wherein the licensing information includes...one or more node license parameters that identifies a characteristic of a computing node”

The Sparks-Bram Combination further includes Bram’s disclosures relating to the claimed one or more node license parameters that identifies a characteristic of a computing node. FEDEX-1003, ¶¶95-99. As discussed, Bram describes a system “designed to run on one or more clusters of various types of computers.” FEDEX-1006, [0232]. “[S]ystem users obtain complete copies of the system...and obtain licenses that allow them to run particular applications.” *Id.*, [0244]. “To manage and apply the licenses,” the system includes “a license server” that includes a “license control file.” *Id.* The license control file “includes a license file defining a base license or license package that allows core components of the system [] to operate and one or more license files defining supplemental licenses that allow additional components of the system [] to operate.” *Id.*

The license server further includes a “software licensing authentication token” with a “license key file.” FEDEX-1006, [0245]. The license key file “provides authorization to apply the available licenses if constraints of the available licenses are satisfied.” *Id.* Accordingly, the key file “can specify parameters and/or constraints for each available license, such as an effective date, an expiration date, a total concurrent activations limit or count, a list of authorized named nodes, a list of authorized operating systems (or operating system aliases), a list of authorized MAC

codes, a list of authorized JVMs, and/or an activation mode (e.g., ‘restricted’ or ‘unrestricted’) of components allowed under a license.” *Id.* Bram further discloses that “the license server 940 can check static constraints (e.g., host restrictions) and dynamic constraints (e.g., counts of the number of instances of a component in the system 10 and/or on a specific node) for a required license associated with a component.” *Id.*, [0275]. Bram includes several figures showing examples of a license key file.

952a

| LICENSE KEY FILE            |                                                       |
|-----------------------------|-------------------------------------------------------|
| LICENSE NAME:               | CORE                                                  |
| LICENSE MODE:               | FULL (UNRESTRICTED)                                   |
| LICENSE AUTHORIZED HOST(S): | NODE1, NODE2                                          |
| LICENSE EFFECTIVE DATE:     | N/A                                                   |
| LICENSE EXPIRATION DATE:    | N/A                                                   |
| EFFECTS:                    |                                                       |
|                             | -CORE COMPONENTS ACTIVATED IN UNRESTRICTED MODE       |
|                             | -CORE COMPONENTS ONLY ALLOWED TO ACTIVATE ON ON       |
|                             | NODE1 AND NODE2                                       |
|                             | -ATTEMPTS TO ACTIVATE CORE COMPONENTS ON NODES OTHER  |
|                             | THAN NODE1 AND NODE2 REJECTED                         |
|                             | -CONFIGURED NON-CORE COMPONENTS ACTIVATED IN          |
|                             | UNRESTRICTED MODE                                     |
|                             | -AUTHORIZATION COMPONENTS NOT ALLOWED TO ACTIVATE     |
| LICENSE NAME:               | FRAUD MANAGER                                         |
| LICENSE MODE:               | EVALUATION (RESTRICTED)                               |
| LICENSE AUTHORIZED HOST(S): | NODE1, NODE2                                          |
| LICENSE EFFECTIVE DATE:     | FIRST ACTIVATION                                      |
| LICENSE EXPIRATION DATE:    | 60 DAYS FROM FIRST ACTIVATION                         |
| LICENSED COPIES:            | 1                                                     |
| EFFECTS:                    |                                                       |
|                             | -FRAUD MANAGEMENT COMPONENTS ACTIVATED IN RESTRICTED  |
|                             | MODE                                                  |
|                             | -FRAUD MANAGEMENT COMPONENTS ONLY ALLOWED TO          |
|                             | ACTIVATE ON NODE1 AND NODE2                           |
|                             | -ON FIRST ACTIVATION OF FRAUD MANAGEMENT COMPONENTS   |
|                             | EXPIRATION DATE SET IN TOKEN                          |
|                             | -FRAUD MANAGEMENT COMPONENTS VALID FOR 60 DAYS FROM   |
|                             | FIRST ACTIVATION                                      |
|                             | -ATTEMPTS TO ACTIVATE FRAUD MANAGEMENT COMPONENTS     |
|                             | AFTER EXPIRATION DATE REJECTED                        |
|                             | -ATTEMPTS TO ACTIVATE FRAUD MANAGEMENT COMPONENTS     |
|                             | ON NODES OTHER THAN NODE1 AND NODE 2 REJECTED         |
|                             | -ONE FRAUD MANAGEMENT COMPONENT (ONE COPY) ALLOWED    |
|                             | TO BE ACTIVATE AT ONE TIME                            |
| LICENSE NAME:               | ATM HANDLING                                          |
| LICENSE MODE:               | EVALUATION (RESTRICTED)                               |
| LICENSE AUTHORIZED HOST(S): | NODE1                                                 |
| LICENSE EFFECTIVE DATE:     | N/A                                                   |
| LICENSE EXPIRATION DATE:    | N/A                                                   |
| EFFECTS:                    |                                                       |
|                             | -ATM HANDLING COMPONENTS ACTIVATED IN RESTRICTED MODE |
|                             | -ATM HANDLING COMPONENTS ONLY ALLOWED TO ACTIVATE ON  |
|                             | NODE1                                                 |
|                             | -ATTEMPTS TO ACTIVATE ATM HANDLING COMPONENTS ON      |
|                             | NODES OTHER THAN NODE1 REJECTED                       |

FEDEX-1006, Fig. 28C. *See also id.*, [0030], [0245], Figs. 28A, 28B, 28D.

Bram goes on to explain that a client, to activate license covered components, broadcasts a get-key request in order to obtain a decryption key from a license server (conditioned on a license check) to decrypt encrypted files (encrypted because they are covered by a license). FEDEX-1006, [0263]-[0266]; FEDEX-1003, ¶98. The license server can “check that the application 16 that generated the get-key request meets the license constraints of the basic license” and can “verify that the node executing the application 16 that generated the get-key request is a valid node.” FEDEX-1006, [0265]. A POSITA would have understood Bram to teach the claimed one or more node license parameters that identifies a characteristic of a computing node because such parameters are checked by a license server in response to receiving a get-key request. FEDEX-1003, ¶99. Examples of such node license parameters include “a list of authorized named nodes, a list of authorized operating systems (or operating system aliases), a list of authorized MAC codes, a list of authorized JVMs, and/or an activation mode (e.g., ‘restricted’ or ‘unrestricted’) of components allowed under a license.” FEDEX-1006, [0245]; *see also* FEDEX-1003, ¶99 (explaining that the node name/identity of a get-key request can be matched against “a list of authorized named nodes,” citing FEDEX-1006, [0245], [0265], [0275]).

**[1b] initializing the computing cluster in a first computing node;**

Sparks-Bram renders obvious [1b]. FEDEX-1003, ¶¶100-106. Indeed, Sparks describes multiple computing cluster initialization steps. *Id.*

As one example, Sparks describes initializing the computing cluster in a first computing node such as the control node.<sup>4</sup> FEDEX-1003, ¶¶101-104. The control node can be used “to logically define the hierarchical organization of [the] distributed computing system,” including “a number of domains, tiers, and node slots.” FEDEX-1005, 6:1–9. “For each of the tiers [(computing cluster)], [the] control node [] creates a number of ‘node slots’ equal to the maximum number of application nodes.” *Id.*, 6:22–31.<sup>5</sup> Moreover, the control node “maintain[s] a database having entries for each node slot, where the entries identify the node

---

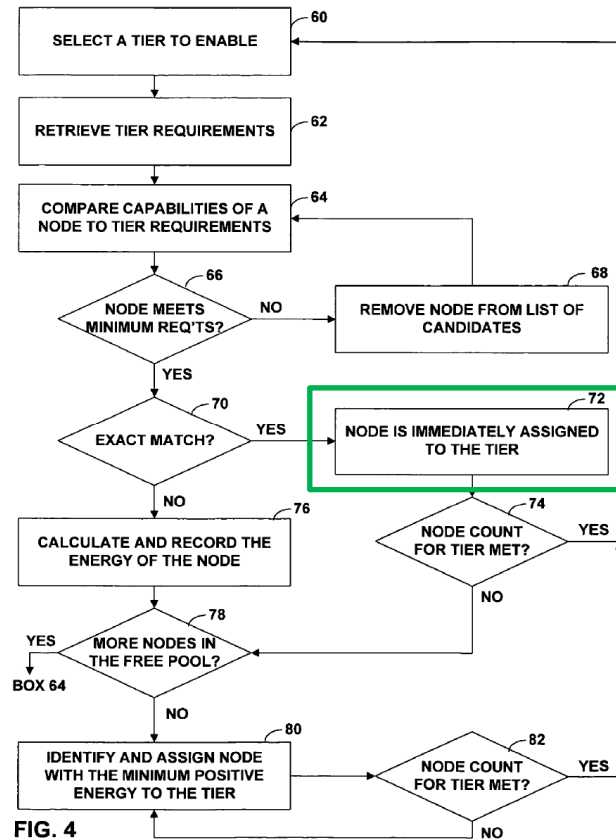
<sup>4</sup> Sparks’s control node is a type of computing node in Sparks. FEDEX-1005, 3:7-25 (discussing that “a distributed computing system 10 [is] constructed from a collection of computing nodes,” which “includes at least one control node 12”), 3:49-59 (“Application nodes 14 typically have more substantial I/O capabilities than control node 12, and are typically configured with more computing resources (e.g., processors and memory)”); FEDEX-1003, ¶101.

<sup>5</sup> A tier (including its computing nodes) is an example of a computing cluster. *See* §V.A.1; §V.A.4, [1pre], [1a-i], [1a-ii]; FEDEX-1003, ¶102.

requirement for each of the tiers” and “retrieves the node requirements for the selected tier from the database.” *Id.*, 9:22–27. A POSITA would have recognized that operations such as defining a tier (computing cluster), creating the node slots of a tier, and maintaining/retrieving a database entry for each of the nodes in a tier satisfy the claimed “initializing” step because of their role in the sequence of steps that leads to an activated computing cluster (see *infra*, [1c] and [1d]). FEDEX-1003, ¶104 (explaining that these operations can be viewed as creating (initializing) the “blueprints” from which the computing cluster is initialized). For example, the tier’s node slots are required to be created (initialized) before computing nodes called application nodes are assigned to them. FEDEX-1003, ¶104; §V.A.4, [1c]. Moreover, these operations are performed in Sparks’s control node. FEDEX-1005, 1:33-39, 6:1–9, 6:22–31, 9:22–27.

As another independent example, Sparks’s assignment of a computing node to a node slot of a tier also satisfies the claimed “initializing” step. FEDEX-1003, ¶105. Based on checking a computing node against the node requirements of the tier, Sparks’s control node can assign a computing node from the free pool to a node slot of the tier. FEDEX-1005, 9:43-51 (discussing that “[i]f the node attributes of the selected computing node and the node requirements of the tier are a perfect match (YES of 70), the computing node is immediately assigned from the free pool to a node slot of the tier and the image instance for the slot is associated with the

computing node for deployment (72)”). *See also id.*, 9:3-13, 9:14-21, 9:52-62, 10:15-26, 10:27-34, Figs. 3-4.



FEDEX-1005, Fig. 4 (annotated).

A POSITA would have recognized that Sparks teaches initializing the computing cluster in a first computing node based on Sparks’s assignment of a first computing node to a tier (computing cluster). FEDEX-1003, ¶106. This understanding, and Sparks’s description (e.g., Sparks’s step 72 which assigns a node that meets minimum tier requirements to the tier), is consistent with the ’416 Patent’s disclosure that “the cluster may be initialized with as few as one member

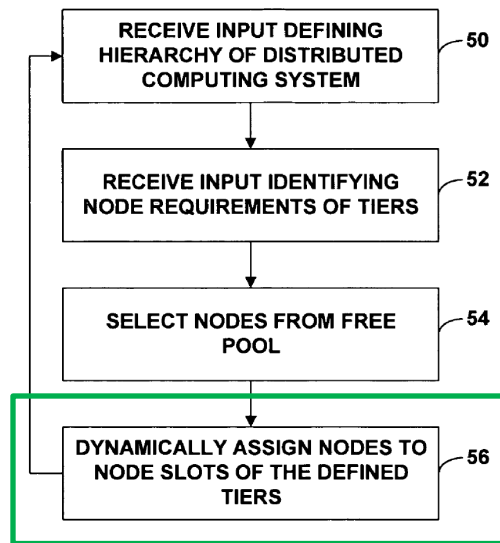
[(computing node)].” FEDEX-1005, 7:33-41 (“if at operation 520 the node has a node license that matches one or more required parameters, then control passes to operation 530 and the cluster may be initialized”; “the cluster may be initialized with as few as one member”), Fig. 5 (step 530, “Initialize Cluster”); FEDEX-1003, ¶106.

**[1c] adding one or more available computing nodes to the computing cluster;  
and**

Sparks-Bram renders obvious [1c]. FEDEX-1003, ¶¶107-111. For example, Sparks discloses that the “[p]opulation of the node slots of the tier may be performed on a tier-by-tier basis” and that “the computing nodes may be assigned [to a tier] using a ‘best fit’ algorithm.”<sup>6</sup> FEDEX-1005, 9:3-13. “Control node 12 dynamically *assigns computing nodes to the node slots of each tier* based on the node requirements specified for the tiers and the identified node attributes (56),” which is shown in Fig. 3. *Id.*

---

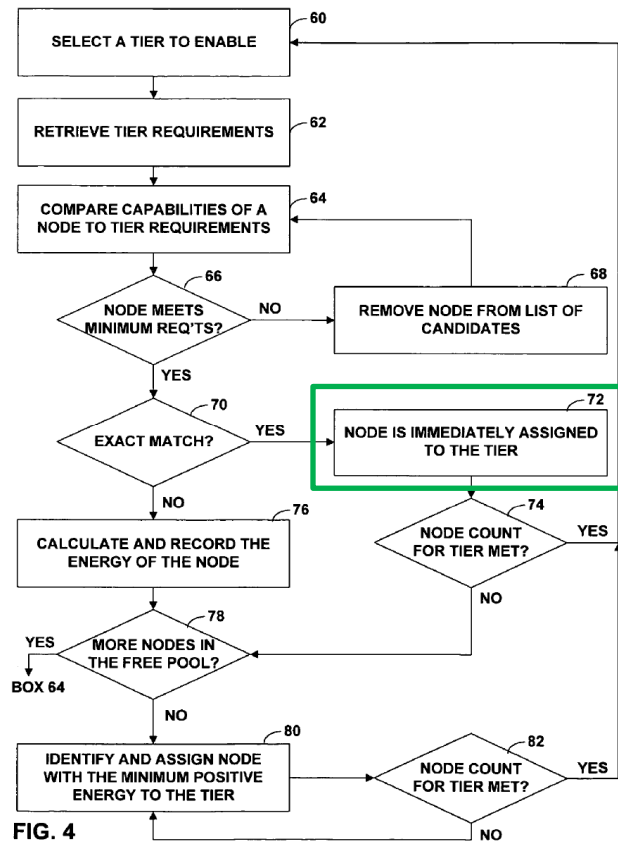
<sup>6</sup> To be clear, this analysis at [1c] is fully consistent with the prior analysis at [1b]. Where the [1b] analysis, as an alternative basis, explains that Sparks’s teaching on assigning a very *first* computing node to a tier renders obvious the claimed initializing step, the analysis here at [1c] relies on Sparks’s further teachings that *additional nodes* are subsequently added to a tier that has been initialized.



FEDEX-1005, Fig. 3 (annotated), 8:37-39 (Fig. 3 of Sparks shows the “operation of control node 12 when configuring distributed computing system 10”).

Sparks further describes that a control node adds an available computing node, which Sparks refers to as an unallocated node or a node from free pool, to a tier (*computing cluster*). FEDEX-1005, 7:1-14 (discussing that “[c]ontrol node 12 automatically *assigns unallocated nodes from free pool 13 to respective tier node slots of a tier*”), 7:15-25 (“control node 12 may automatically *add unallocated nodes from free pool 13 to a tier*”), 8:14-25 (“control node 12 may also *add nodes from free pool 13 to a tier* when more processing capacity is needed within the tier”), 3:40-48 (“free pool 13 includes *a set of unallocated nodes that are available for use within distributed computing system 10*”; “control node 12 may use unallocated nodes from free pool 13...to *add an application node to allocated pool 15* to increase processing capacity”), 8:61-9:2, 9:28-30.

Sparks provides additional details of node assignment. Figure 4 of Sparks, for example, shows the “operation of control node 12 when assigning computing nodes to node slots of tiers.” FEDEX-1005, 9:14-21. Based on checking a computing node against the node requirements of the tier, the control node can assign a computing node from the free pool to a node slot of the tier. FEDEX-1005, 9:43-51 (discussing that “[i]f the node attributes of the selected computing node and the node requirements of the tier are a perfect match (YES of 70), the computing node is immediately assigned from the free pool to a node slot of the tier and the image instance for the slot is associated with the computing node for deployment (72)”). *See also id.*, 9:52-62, 10:15-26, 10:27-34.



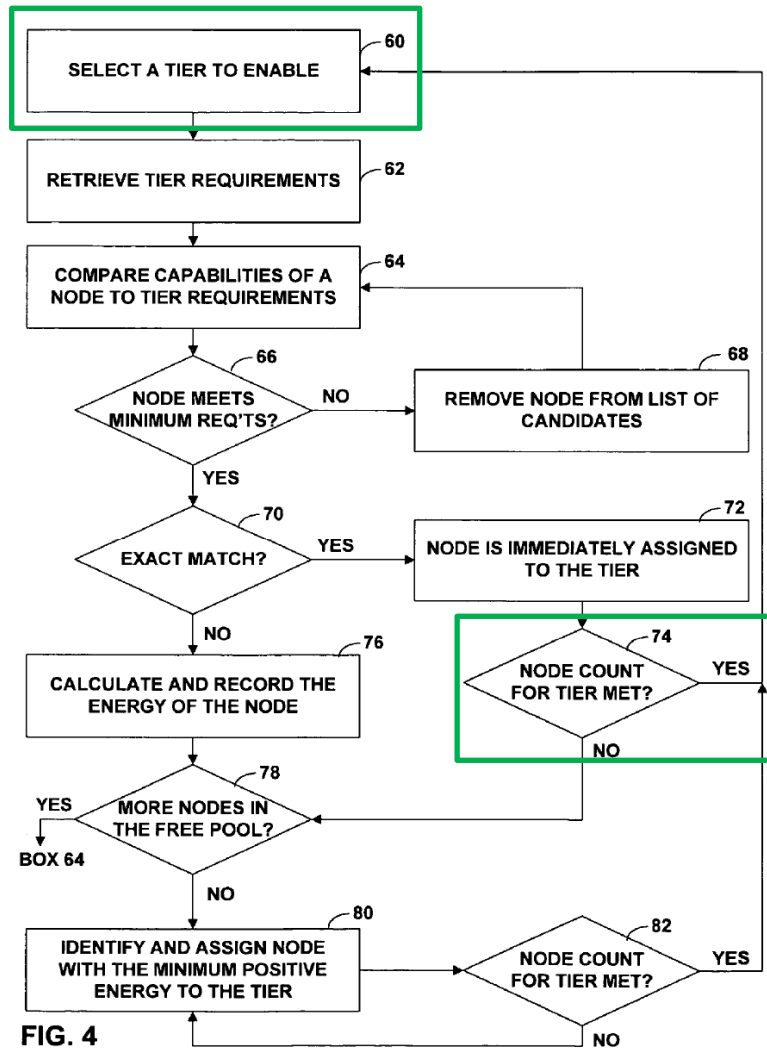
FEDEX-1005, Fig. 4 (annotated).

In sum, a POSITA would have recognized that Sparks discloses adding (see, e.g., assigning steps in Figs. 3 and 4) one or more available computing nodes (“unallocated nodes that are available for use”) to the computing cluster (tier in Sparks’s distributed computing system). FEDEX-1003, ¶111.

**[1d] activating the computing cluster when the computing cluster includes a number of nodes that complies with the bundle-type parameter.**

Sparks-Bram renders obvious [1d]. FEDEX-1003, ¶¶112-116. For example, Sparks describes the “operation of control node 12 when assigning computing nodes to node slots of tiers,” which is shown in Fig. 4. FEDEX-1005, 9:14-21. In this

context, Sparks discloses that the control node selects and enables a tier. FEDEX-1005, 9:14-21 (discussing that “control node 12 selects a tier to enable (60)” and “may, for example, initially select the tier with the highest priority and successively enable the tiers based on priority”).



FEDEX-1005, Fig. 4 (annotated). During this process, the control node “determines whether the node count for the tier is met (74).” FEDEX-1005, 9:52-62. “When the node count for the tier is met, control node 12 selects the next tier to enable, e.g., the

tier with the next largest priority, and repeats the process until all defined tiers are enabled, i.e., populated with application nodes (60).” *Id.* Sparks additionally discloses that “a user may request that a newly created tier come online” via enabling the tier. FEDEX-1005, 18:45–50.

A POSITA would have recognized that Sparks discloses activating the computing cluster (e.g., “enable” a tier) when the computing cluster includes a number of nodes that complies with the bundle-type parameter (e.g., “[w]hen the node count for the tier is met”). FEDEX-1003, ¶114; *see supra* [1a-ii] (explaining that Sparks’s disclosure of minimum and/or maximum node counts renders obvious the claimed bundle-type parameter). Moreover, a POSITA would have understood, or it would have been obvious to a POSITA, that the computing cluster is activated when for example, the node count for the tier is met because there would be enough nodes in a tier to perform the tier’s assigned function. *Id.* This is consistent with how a cluster is activated in the specification of the ’416 Patent: “if ...the node count for the cluster has not been reached[] (i.e., if the cluster has not yet activated a number of nodes authorized by the license terms), then ... a next candidate node for inclusion in the cluster may be identified.” FEDEX-1001, 7:42-51.

In addition, compliance with the at least one additional bundle-type characteristic of the cluster is evidenced through Sparks’s disclosure of requirement checks before a node is added to the tier (cluster). FEDEX-1003, ¶115. Sparks

discloses, for example, that “control node 12 determines whether the node attributes of the computing node meets the minimum *node requirements of the tier* (66)” and if they do, the computing node is assigned to a node slot of the tier. FEDEX-1005, 9:22-51; *see also* §V.A.4, [1c]. Like Sparks, Bram describes that components are activated if constraints are satisfied. FEDEX-1006, [0244] (“The license control file 950 specifies what licenses are required in order to activate a particular application 16 or component of the system 10”), [0245]; FEDEX-1003, ¶116.

**(b) Claim 14**

**[14pre] A computer program product comprising logic instructions stored on a non-transitory computer-readable medium which, when executed by a computer processor, configure the processor to:**

To the extent the preamble is limiting, Sparks-Bram renders obvious [14pre]. FEDEX-1003, ¶¶191-192; §V.A.4, [1pre]. Spark discloses that each computing node, including a control node, includes “computing resources (e.g., *processors* and memory).” FEDEX-1005, 3:49-59. Spark further discloses that “[e]ach computing node includes one or more programmable *processors for executing software instructions stored on one or more computer-readable media*.” FEDEX-1005, 3:23-25. *See also id.*, 1:65-2:7 (discussing that a “*computer-readable medium* comprises instructions that cause a processor to” perform operations).

**[14a-i] store license information for a computing cluster in a memory module associated with a computing node,**

Sparks-Bram renders obvious [14a-i]. FEDEX-1003, ¶193; §V.A.4, [1a-i].

**[14a-ii] wherein the licensing information includes a bundle-type parameter that identifies a size of the cluster and at least one additional bundle-type characteristic of the cluster, and one or more node license parameters that identifies a characteristic of a computing node;**

Sparks-Bram renders obvious [14a-ii]. FEDEX-1003, ¶194; §V.A.4, [1a-ii].

**[14b] initialize the computing cluster in a first computing node;**

Sparks-Bram renders obvious [14b]. FEDEX-1003, ¶195; §V.A.4, [1b].

**[14c] add one or more available computing nodes to the computing cluster; and**

Sparks-Bram renders obvious [14c]. FEDEX-1003, ¶196; §V.A.4, [1c].

**[14d] activate the computing cluster when the computing cluster includes a number of nodes that complies with the bundle-type parameter.**

Sparks-Bram renders obvious [14d]. FEDEX-1003, ¶197; §V.A.4, [1d].

## **5. Claims 2, 15**

**[2] The method of claim 1, wherein storing license information for a computing cluster in a memory module associated with a computing node comprises retrieving licensing information from an external licensing server.**

Sparks-Bram renders obvious [2]. FEDEX-1003, ¶117-122. *See also* §V.A.4, [1a-i], [1a-ii]. That is, it would have been apparent or at least obvious to a POSITA that Bram's teachings, as integrated with Sparks, provides retrieving licensing information from an external licensing server, as claimed. FEDEX-1003, ¶118. For

example, Bram describes that “[t]o manage and apply the licenses associated with a particular system user, the system 10 can include a *license server*.” FEDEX-1006, [0244]. *See also id.*, [0005] (“a license server configured to receive an activation authorization request from the component, to determine if the component is associated with at least one required license based on the plurality of mappings, and to provide a response to the component”), [0006], [0245] (discussing that “the license server 940 can authorize activation of applications 16 and/or components associated with available licenses”).

A POSITA would have recognized, or it would have been obvious to a POSITA, that Bram’s *license server* is an external licensing server. FEDEX-1003, ¶120. In fact, Bram discloses multiple license servers including primary and backup license servers, which are external to the computing cluster. FEDEX-1006, [0252] (discussing that the system “can include multiple license servers”; “each version of the system 10 can include a separate primary license server and one or more separate backup license servers”), [0304] (“The system 10 can address disaster recovery by supporting a geographically distributed configuration that can run in multiple processing sites simultaneously”; “the system 10 uses various approaches to provide multi-site processing, such as hot backups, cold backups, and variations in between, which are commonly called warm backups”), [0232] (“Multiple clusters can run

simultaneously at multiple locations in order to provide mutual backup options for a single system”), [0257]-[0259], [0282], Fig. 28G.

It would have been obvious to a POSITA—and a POSITA would have been motivated—to modify Sparks’s control node based on Bram’s disclosures on primary and backup licensing servers, such that the modified control node, acting as a backup licensing server, receives licensing information from Bram’s primary licensing server. FEDEX-1003, ¶121; FEDEX-1006, [0258] (“the backup license server receives floating license information and updated floating license information from the primary license server”), [0257] (“the license server 940 can set itself as a backup server and can subscribe (e.g., create a subscriber process or thread) to the license server that has published itself as the primary license server”), [0259], Fig. 28G. In addition to the reasons in Section V.A.3, a POSITA would have been motivated to modify Sparks’s control node to communicate with one or more external servers for backup, redundancy, and/or disaster recovery reasons. FEDEX-1003, ¶122; FEDEX-1006, [0232], [0252], [0304].

**[15] The computer program product of claim 14, further comprising logic instructions which, when executed by the processor, configure the processor to retrieve licensing information from an external licensing server.**

Sparks-Bram renders obvious [15]. FEDEX-1003, ¶198; §V.A.4, [14]; §V.A.5, [2].

## 6. Claim 3

**[3] The method of claim 1, wherein the bundle-type parameter identifies at least one of a minimum or maximum node count threshold for the computing cluster.**

Sparks-Bram renders obvious [3]. FEDEX-1003, ¶¶123-124. As discussed for [1a-ii], Sparks-Bram provides a bundle-type parameter that identifies a size of the cluster. *See* §V.A.4, [1a-ii]. Sparks describes that this parameter identifies at least one of a minimum or maximum node count threshold for the computing cluster. FEDEX-1005, 6:55-67 (“System administrator 20 may define policies regarding tier characteristics, such as a *minimum number of nodes a tier requires*, ...”), 6:22-30 (“control node 12 creates a number of ‘node slots’ equal to the *maximum number of application nodes* 14”; “each node slot represents a data set that describes specific information for a corresponding node”), 9:52-62 (discussing that “[w]hen the node count for the tier is met, control node 12 selects the next tier to enable”), 10:15-35 (“Control node 12 determines whether the minimum node count for the tier is met (82)”), Fig. 4. *See also* §V.A.4, [1d].

## 7. Claim 4

**[4] The method of claim 1, wherein: the node license parameters identify one or more bundle-types compatible with the computing node.**

Sparks-Bram renders obvious [4]. FEDEX-1003, ¶¶125-130. For example, Bram discloses activation of applications and/or components if license constraints are satisfied by the current state of the system. FEDEX-1006, [0245] (“The license

key file 952 provides authorization to *apply the available licenses if constraints of the available licenses are satisfied by the current state of the system 10*"; "Using the license key file 952, the license server 940 can authorize activation of applications 16 and/or components associated with available licenses and can disable (e.g., prevent activation) of applications 16 and/or components associated with unavailable licenses or available licenses that have unsatisfied constraints"), [0255], [0265], [0273] ("using the basic license included in the floating license information, the license server 940 verifies that the core constraints of the system are satisfied").

Bram further discloses various examples of constraints including "a list of authorized named nodes, a list of authorized operating systems (or operating system aliases), a list of authorized MAC codes, a list of authorized JVMs, and/or an activation mode (e.g., 'restricted' or 'unrestricted') of components allowed under a license." FEDEX-1006, [0245]; *see supra* analysis at [1a-ii] (identifying node license parameters). Bram further discloses that "the license server 940 can check static constraints (e.g., host restrictions) and dynamic constraints (e.g., counts of the number of instances of a component in the system 10 and/or on a specific node) for a required license associated with a component." *Id.*, [0275].

Bram includes several figures showing examples of a license key file which provides node license parameters (e.g., constraints) for multiple licenses (named,

e.g., “CORE,” “FRAUD MANAGER,” “ATM HANDLING,” and “BASIC GATEWAY”).

|      |                                                                                                 |  |
|------|-------------------------------------------------------------------------------------------------|--|
| 952a | LICENSE KEY FILE                                                                                |  |
|      | LICENSE NAME: CORE                                                                              |  |
|      | LICENSE MODE: FULL (UNRESTRICTED)                                                               |  |
|      | LICENSE AUTHORIZED HOST(S): NODE1, NODE2                                                        |  |
|      | LICENSE EFFECTIVE DATE: N/A                                                                     |  |
|      | LICENSE EXPIRATION DATE: N/A                                                                    |  |
|      | EFFECTS:                                                                                        |  |
|      | -CORE COMPONENTS ACTIVATED IN UNRESTRICTED MODE                                                 |  |
|      | -CORE COMPONENTS ONLY ALLOWED TO ACTIVATE ON ON NODE1 AND NODE2                                 |  |
|      | -ATTEMPTS TO ACTIVATE CORE COMPONENTS ON NODES OTHER THAN NODE1 AND NODE2 REJECTED              |  |
|      | -CONFIGURED NON-CORE COMPONENTS ACTIVATED IN UNRESTRICTED MODE                                  |  |
|      | -AUTHORIZATION COMPONENTS NOT ALLOWED TO ACTIVATE                                               |  |
|      |                                                                                                 |  |
|      | LICENSE NAME: FRAUD MANAGER                                                                     |  |
|      | LICENSE MODE: EVALUATION (RESTRICTED)                                                           |  |
|      | LICENSE AUTHORIZED HOST(S): NODE1, NODE2                                                        |  |
|      | LICENSE EFFECTIVE DATE: FIRST ACTIVATION                                                        |  |
|      | LICENSE EXPIRATION DATE: 60 DAYS FROM FIRST ACTIVATION                                          |  |
|      | LICENSED COPIES: 1                                                                              |  |
|      | EFFECTS:                                                                                        |  |
|      | -FRAUD MANAGEMENT COMPONENTS ACTIVATED IN RESTRICTED MODE                                       |  |
|      | -FRAUD MANAGEMENT COMPONENTS ONLY ALLOWED TO ACTIVATE ON NODE1 AND NODE2                        |  |
|      | -ON FIRST ACTIVATION OF FRAUD MANAGEMENT COMPONENTS EXPIRATION DATE SET IN TOKEN                |  |
|      | -FRAUD MANAGEMENT COMPONENTS VALID FOR 60 DAYS FROM FIRST ACTIVATION                            |  |
|      | -ATTEMPTS TO ACTIVATE FRAUD MANAGEMENT COMPONENTS AFTER EXPIRATION DATE REJECTED                |  |
|      | -ATTEMPTS TO ACTIVATE FRAUD MANAGEMENT COMPONENTS ON NODES OTHER THAN NODE1 AND NODE 2 REJECTED |  |
|      | -ONE FRAUD MANAGEMENT COMPONENT (ONE COPY) ALLOWED TO BE ACTIVATE AT ONE TIME                   |  |
|      |                                                                                                 |  |
|      | LICENSE NAME: ATM HANDLING                                                                      |  |
|      | LICENSE MODE: EVALUATION (RESTRICTED)                                                           |  |
|      | LICENSE AUTHORIZED HOST(S): NODE1                                                               |  |
|      | LICENSE EFFECTIVE DATE: N/A                                                                     |  |
|      | LICENSE EXPIRATION DATE: N/A                                                                    |  |
|      | EFFECTS:                                                                                        |  |
|      | -ATM HANDLING COMPONENTS ACTIVATED IN RESTRICTED MODE                                           |  |
|      | -ATM HANDLING COMPONENTS ONLY ALLOWED TO ACTIVATE ON NODE1                                      |  |
|      | -ATTEMPTS TO ACTIVATE ATM HANDLING COMPONENTS ON NODES OTHER THAN NODE1 REJECTED                |  |

FEDEX-1006, Fig. 28C.

952a

| LICENSE KEY FILE                                                                                     |  |
|------------------------------------------------------------------------------------------------------|--|
|                                                                                                      |  |
| LICENSE NAME: CORE                                                                                   |  |
| LICENSE MODE: FULL (UNRESTRICTED)                                                                    |  |
| LICENSE AUTHORIZED OS(S): SOLARIS                                                                    |  |
| LICENSE EFFECTIVE DATE: N/A                                                                          |  |
| LICENSE EXPIRATION DATE: N/A                                                                         |  |
| EFFECTS:                                                                                             |  |
| -CORE COMPONENTS ACTIVATED IN UNRESTRICTED MODE                                                      |  |
| -CORE COMPONENTS ONLY ALLOWED TO ACTIVATE ON NODES<br>SUPPORTED BY SOLARIS OPERATING SYSTEM          |  |
|                                                                                                      |  |
| LICENSE NAME: BASIC GATEWAY                                                                          |  |
| LICENSE MODE: FULL (UNRESTRICTED)                                                                    |  |
| LICENSE AUTHORIZED OS(S): SOLARIS                                                                    |  |
| LICENSE EFFECTIVE DATE: N/A                                                                          |  |
| LICENSE EXPIRATION DATE: N/A                                                                         |  |
| LICENSED COPIES: 4                                                                                   |  |
| EFFECTS:                                                                                             |  |
| -BASIC GATEWAY COMPONENTS ACTIVATED IN UNRESTRICTED MODE                                             |  |
| -BASIC GATEWAY COMPONENTS ONLY ALLOWED TO ACTIVATE ON<br>NODES SUPPORTED BY SOLARIS OPERATING SYSTEM |  |
| -UP TO 4 BASIC GATEWAY COMPONENTS (4 COPIES) ALLOWED TO BE<br>ACTIVATE AT ONE TIME                   |  |

FEDEX-1006, Fig. 28D. *See also id.*, [0030], [0244], [0245], Figs. 28A, 28B.

A POSITA would have understood that Bram's constraints (node license parameters, per [1a-ii] analysis) identify one or more bundle-types compatible with the computing node. FEDEX-1003, ¶¶129-130. The "CORE" license provides core functionality, "ATM" provides ATM functionality (as in automated teller machine), the "BASIC GATEWAY" provides gateway functionality. FEDEX-1006, [0052], [0076], Figs. 28C and 28D. As shown in Bram's Fig. 28C, the bundle-type of "CORE" is compatible with nodes 1 and 2. FEDEX-1003, ¶129. As shown in Bram's Fig. 28D, the bundle-type of "BASIC GATEWAY" is compatible with any node

configured to run the operation system (OS) named SOLARIS. *Id.* This understanding is consistent with the '416 Patent which discloses that “clusters may be categorized into bundle types by one or more additional parameters such as, e.g., an array, tape, computing, translating functional bundle” and that “[a] bundle may be sold as a set or group of node or grid types to meet a specific market or function.” FEDEX-1001, 5:48-64; *see also id.*, 5:43-45 (“compatibility enforcement may include compatibility between hardware, software, firmware, and/or licensing requirements”); FEDEX-1003, ¶130.

## 8. Claims 5, 16

**[5] The method of claim 1, wherein adding one or more available computing nodes to the computing cluster comprises assigning a computing node a new node license when the computing node license is incompatible with a brand-type associated with the computing cluster.**

Sparks-Bram renders obvious [5]. FEDEX-1003, ¶¶131-139. As discussed, Sparks-Bram provides for adding one or more available computing nodes to the computing cluster. *See* §V.A.4, [1c]. Sparks-Bram further renders obvious assigning a computing node a new node license when the computing node license is incompatible (*e.g.*, license requirement/constraint is not satisfied) with a brand-type associated with the computing cluster (*e.g.*, license requirement/constraint customized to a specific node/cluster), as claimed. FEDEX-1003, ¶132.

First, Bram discloses how one or more requirements/constraints of a computing node license can be or can become unsatisfied. FEDEX-1006, [0245] (“the license server 940 ... can disable (e.g., prevent activation) of applications 16 and/or components associated with unavailable licenses or *available licenses that have unsatisfied constraints*”), [0293] (“a component can terminate if a *license required by the component has expired, is no longer available, has been changed, or includes constraints that are no longer satisfied*”), [0283] (discussing that the license server can “recheck the state of the system 10, the license control file 951, the token 951, and/or the license key file 952”), [0287] (“the primary license server can determine if any licenses have become effective or expired”), [0254], [0278], [0279]; FEDEX-1003, ¶133. A POSITA would have understood that a computing node license is *incompatible* as claimed if one or more of the license’s requirements/constraints are not satisfied. FEDEX-1003, ¶133.

Second, in the Sparks-Bram Combination, a computing node license assigned to a computing node may be *incompatible* with its requirements/constraints if, for example, the computing node (and its license) was previously assigned to a different tier (computing cluster). FEDEX-1003, ¶134; *see also* §V.A.4, [1a-ii], [1b], [1c]. Sparks discloses that (1) computing nodes can be assigned to different tiers such as a web tier, an application tier, and a database tier and (2) computing nodes may be transferred between tiers, e.g., tier to tier, or tier to free pool to another tier, *etc.*

FEDEX-1005, 5:48–58, 6:10-21, 7:15-25 (discussing that the control node can “remove nodes from a tier to the free pool when the tier has excess capacity” or “transfer nodes from tier to tier to meet processing demands”), 7:1-14 (discussing that “[c]ontrol node 12 automatically assigns unallocated nodes from free pool 13 to respective tier node slots of a tier”), 3:40-48, 8:14-25, 8:61-9:2, 9:28-30; FEDEX-1003, ¶135.

Moreover, Bram discloses that licenses can be customized to a specific configuration. FEDEX-1006, [0247] (“[l]icenses can also be tied to a system user’s specific configuration so that system users cannot share licenses”; “[b]y customizing licenses and separating licenses for individual components, licenses can be created that expire on various schedules and have different prices”). Accordingly, it would have been obvious to a POSITA to customize licenses to specific nodes/tiers such that they can be priced accordingly, e.g., “have different prices.” FEDEX-1003, ¶¶136-137; FEDEX-1006, [0247]; *see also* §V.A.3.

As an example, a license can be customized to a specific computing node and to a specific tier, e.g., web tier, such that it allows a component to operate on the computing node only if the node is assigned to the web tier. FEDEX-1003, ¶137. However, the constraints/requirements of that license would not be satisfied or may become unsatisfied, *i.e.*, ***incompatible***, if the node is assigned to a non-web tier (*e.g.*, application or database tier). *Id.* This node/license customization to a specific tier

would have been recognized by the POSITA as a *brand-type associated with the tier (computing cluster)*. This understanding is consistent with the '416 Patent, which discloses “the node may be ‘branded’ as a licensed node having a specific bundle type” and “[t]he node license may then be assigned specifically to the node (and may not be assigned to any other node).” FEDEX-1001, 7:19-32; *see also* 7:62-8:8 (discussing to “determine[] whether the node’s brand is compatible with the cluster as defined by the Right-To-Use characteristics”); FEDEX-1003, ¶138.

Lastly, a POSITA would have recognized that, in the tier reassignment example discussed above, a *computing node license assigned to the computing node* may become *incompatible with a brand-type associated with the computing cluster* if reassigned. FEDEX-1003, ¶139. For example, a new node license may be required, and accordingly assigned, if license requirements of the new tier are different than those of the old tier. *Id.* As such, it would have been obvious to a POSITA to *assign*, as needed, *a new node license to a computing node* that is switching between tiers (computing clusters) to ensure that the computing node continues to provide computing resources when added to the new tier. FEDEX-1003, ¶139; §V.A.3.

**[16] The computer program product of claim 14, further comprising logic instructions which, when executed by the processor, configure the processor to assign a computing node a new node license when the computing node license is incompatible with a brand-type associated with the computing cluster.**

Sparks-Bram renders obvious [16]. FEDEX-1003, ¶199; §V.A.4, [14]; §V.A.8, [5].

## **9. Claims 6, 17**

**[6] The method of claim 1, further comprising initiating a license check while the computing system is operating, wherein the license check evaluates at least one of: a license status of the computing node; a node count in the computing cluster; and a brand-type associated with the computing node.**

Sparks-Bram renders obvious [6]. FEDEX-1003, ¶¶140-147.

*“initiating a license check while the computing system is operating”*

Bram discloses that the “the primary license server can initialize a timer, and once the timer reaches zero or another predetermined count, the timer can alert the license server 940 that it should *recheck the state of the system 10, the license control file 951, the token 951, and/or the license key file 952.*” FEDEX-1006, [0283]. Bram discloses that “[u]pdating or refreshing the license control file 950, the token 951, and/or the license key file 952 can also cause the primary license server to automatically recheck the state of the system 10, the license control file 950, the token 951, and/or the license key file 952.” *Id. See also id.*, [0254], [0279], [0287], [0293]. As such, a POSITA would have understood that Bram discloses initiating a

license check while the computing system is operating, as claimed. FEDEX-1003, ¶141.

“the license check evaluates at least one of: a license status of the computing node”<sup>7</sup>

Bram discloses evaluating a license status of the computing node, as claimed. FEDEX-1006, [0287] (“the primary license server can *determine if any licenses have become effective or expired*”), [0293] (“a component can terminate if a *license required by the component has expired, is no longer available, has been changed, or includes constraints that are no longer satisfied*”), [0247] (“the license server 940 can also provide current *license status information* to a system user, a system provider, and/or a system manager for tracking and/or billing purposes”), [0245] (discussing various parameters and/or constraints for each available license); FEDEX-1003, ¶142.

---

<sup>7</sup> Claim 6 recites “the license check evaluates at least one of [A], [B], and [C].” This petition presents arguments for each of the license check evaluations (*i.e.*, [A], [B], and [C]). However, any one of the arguments is sufficient to satisfy this limitation of claim 6.

“the license check evaluates ... a node count in the computing cluster”

Bram discloses evaluating a node count in the computing cluster, as claimed. FEDEX-1006, [0245] (“each license key file 952 can specify parameters and/or constraints for each available license, such as ... ***a total concurrent activations limit or count, a list of authorized named nodes***”), [0275] (“the license server 940 can check static constraints (e.g., host restrictions) and dynamic constraints (e.g., counts of the number of instances of a component in the system 10 and/or on a specific node) for a required license associated with a component”); FEDEX-1003, ¶143.

“the license check evaluates ... brand-type associated with the computing node”

Sparks-Bram renders obvious a license check that evaluates a brand-type associated with the computing node, as claimed. FEDEX-1003, ¶¶144-147. Sparks discloses that computing nodes can be assigned to different tiers such as a web tier, an application tier, and a database tier. FEDEX-1005, 5:48–58, 6:10-21; *see also* §§V.A.1, V.A.4. Moreover, Bram discloses that “[l]icenses can also be tied to a system user’s specific configuration so that system users cannot share licenses.” FEDEX-1006, [0247].

Bram further discloses “[b]y customizing licenses and separating licenses for individual components, licenses can be created that expire on various schedules and have different prices.” *Id.* It would have been obvious to a POSITA to implement a

license check that evaluates a brand-type associated with the computing node because licenses can be customized to specific tiers/nodes such that they can be priced accordingly. FEDEX-1003, ¶146; FEDEX-1006, [0247]; §V.A.3. For example, a license can be customized (branded) to a specific computing node and to a specific tier, e.g., web tier, such that it allows a component to operate on the computing node only if the node is assigned to the web tier. FEDEX-1003, ¶146. However, the constraints/requirements of that license would not be satisfied or may become unsatisfied if the node is assigned to a non-web tier (e.g., application or database tier). *Id.*; FEDEX-1006, [0293] (“a component can terminate if a license required by the component has expired, is no longer available, has been changed, or includes constraints that are no longer satisfied”).

This is consistent with the '416 Patent which discloses “the node may be ‘branded’ as a licensed node having a specific bundle type” and “[t]he node license may then be assigned specifically to the node (and may not be assigned to any other node).” FEDEX-1001, 7:19-32; *see also* 7:62-8:8 (discussing to “determine[] whether the node’s brand is compatible with the cluster as defined by the Right-To-Use characteristics”); FEDEX-1003, ¶147.

**[17] The computer program product of claim 14, further comprising logic instructions which, when executed by the processor, configure the processor to initiate a license check while the computing system is operating, wherein the license check evaluates at least one of: a license status of the computing node; a node count in the computing cluster; and a brand-type associated with the computing node.**

Sparks-Bram renders obvious [17]. FEDEX-1003, ¶200; §V.A.4, [14]; §V.A.9, [6].

## **10. Claims 7, 18**

**[7] The method of claim 1, wherein the license check comprises implementing an error routine that inhibits operation of the computing node when an incompatibility between the computing node and the computing cluster is detected.**

Sparks-Bram renders obvious [7]. FEDEX-1003, ¶¶148-151; §V.A.4, [1]; §V.A.9, [6].<sup>8</sup> For example, Bram discloses that the “the primary license server can initialize a timer, and once the timer reaches zero or another predetermined count, the timer can alert the license server 940 that it should *recheck the state of the system*

---

<sup>8</sup> Claim 7 recites “the license check comprises...” Claim 7 is a dependent claim of claim 1 which does not recite “license check.” However, claim 6 recites a “license check.” To the extent Patent Owner is allowed to change the dependency of claim 7 to claim 6 or argues that claim 7 requires a license check that comprises the limitations recited therein, claim 7 is unpatentable over the Sparks-Bram Combination for the reasons presented herein.

***10, the license control file 951, the token 951, and/or the license key file 952.”***

FEDEX-1006, [0283]. According to Bram, “[u]pdating or refreshing the license control file 950, the token 951, and/or the license key file 952 can also cause the primary license server to ***automatically recheck the state of the system 10, the license control file 950, the token 951, and/or the license key file 952.”*** *Id.* See also *id.*, [0293] (“Each component can ***recheck*** its current license permission and can react accordingly”; “***a component can terminate if a license required by the component has expired, is no longer available, has been changed, or includes constraints that are no longer satisfied***”), [0254], [0273], [0279], [0287].

Bram further discloses that “the license server 940 can authorize activation of applications 16 and/or components associated with available licenses and ***can disable (e.g., prevent activation)*** of applications 16 and/or components associated with unavailable licenses or available licenses that have unsatisfied constraints.” *Id.*, [0245]. See also *id.*, [0273] (“using the basic license included in the floating license information, the license server 940 verifies that the core constraints of the system are satisfied”; “[i]f the core constraints are not satisfied (step 1026), the license server 940 can create a ***reject reply*** (step 1028) and can send the reject reply to the application 16 (step 1030)”).

A POSITA would have understood that Bram discloses, or at least renders obvious, a license check comprising implementing an error routine that inhibits

operation (e.g., “disable”, “prevent activation,” send a “reject reply”) of the computing node when an incompatibility (e.g., “constraints are not satisfied” / “constraints that are no longer satisfied”) between the computing node and the computing cluster is detected. FEDEX-1003, ¶¶150-151; FEDEX-1006, [0245], [0273], [0293].

**[18] The computer program product of claim 14, further comprising logic instructions which, when executed by the processor, configure the processor to implement an error routine that inhibits operation of the computing node when an incompatibility between the computing node and the computing cluster is detected.**

Sparks-Bram renders obvious [18]. FEDEX-1003, ¶201; §V.A.4, [14]; §V.A.10, [7].

## **11. Claim 8**

**[8] The method of claim 1, further comprising adding a computing node to the activated computing cluster.**

Sparks-Bram renders obvious [8]. FEDEX-1003, ¶¶152-153. Sparks discloses “*adding an additional computing node to a tier* to meet increased processing demands” and that “[w]hen there are one or more nodes that meet the minimum node requirements of the tier, control node 12 selects the node from the free pool based the node requirements of the tier, as described above, (94) and *assigns the node to the tier* (95).” FEDEX-1005, 10:35-62; Fig. 5. *See also id.*, 3:40-48 (discussing that “control node 12 may use unallocated nodes...to *add an application node to*

*allocated pool 15* to increase processing capacity of distributed computing system 10”), 1:51-59 (“a method comprises detecting a node added to a network and automatically identifying attributes for the detected node”; “automatically deploying the detected node within the distributed computing system in accordance with the model based on the attributes”).

## 12. Claim 9

**[9pre] A computing system comprising at least two computing nodes, wherein at least one of the computing nodes comprises:**

To the extent the preamble is limiting, Sparks-Bram renders obvious [9pre]. FEDEX-1003, ¶¶154-156. Sparks’s “[d]istributed computing system 10 may be viewed as *a collection of computing nodes* operating in cooperation with each other to provide high-performance processing.” FEDEX-1005, 3:7-25. The system “may include *8 to 512 computing nodes or more.*” *Id.* Sparks discloses that these computing nodes include one or more control nodes and application nodes, and that each include “one or more programmable processors for executing software instructions stored on one or more computer-readable media.” FEDEX-1005, 3:7-25; *see also id.*, 1:12-21, 3:49-59, Fig. 1.

Further, claim 9 recites multiple modules including a cluster module ([9a-i]), node module ([9b-i]), and management module ([9c-i]), each having a group of functionalities. As discussed more below, a POSITA would have recognized that

Sparks and/or Bram discloses, or at least renders obvious, each of these claimed modules. FEDEX-1003, ¶156. Further, Sparks’s control node has multiple software modules that provide for respective control node functionalities. *E.g.*, FEDEX-1005, 13:52-57 (discussing that the control node includes a “monitoring subsystem 202, a service level automation infrastructure (SLAI) 204, and a business logic tier (BLT) 206”), Fig. 15; *see also id.*, 14:52-55 (“modular architecture”), 15:37-41 (same). Alternatively, and additionally, it would have been obvious to a POSITA to implement Sparks’s control node as having different modules for the claimed functions. FEDEX-1003, ¶156. It would have been obvious to a POSITA to implement different groups of functionalities as separate software modules given the advantages of modular programming in terms of more streamlined development, improved code debugging, and improved testing. *Id.*

**[9a-i] a cluster module stored on a non-transitory computer readable medium, the cluster module configured to manage compatibility operations associated with a computing cluster,**

Sparks-Bram renders obvious [9a-i]. FEDEX-1003, ¶¶157-159. Sparks discloses a non-transitory computer readable medium. FEDEX-1005, 3:23-25 (“[e]ach computing node includes one or more programmable processors for executing software instructions stored on *one or more computer-readable media*”); 1:65-2:7 (discussing that a “*computer-readable medium* comprises instructions that cause a processor to” perform operations). A POSITA would have recognized that

Sparks discloses, or at least renders obvious, a cluster module stored on a non-transitory computer readable medium because instructions corresponding to the relevant functionality of Sparks's control node in the Sparks-Bram Combination can be stored on Sparks's computer-readable media. FEDEX-1003, ¶158; §V.A.4, [1c], [1d]; §V.A.12, [9pre] (addressing "modules").

Sparks further discloses that the cluster module is configured to manage compatibility operations (*e.g.*, check node/tier requirements) associated with a computing cluster (*e.g.*, tier), as claimed. FEDEX-1003, ¶159. Sparks describes for example that "[o]ne or more control nodes provide for the efficient and automated allocation and management of computing functions and resources within the distributed computing system in accordance with the organization model." FEDEX-1005, 1:33–39. Sparks discloses that the "[c]ontrol node 12 dynamically assigns computing nodes to the node slots of each tier based on the node requirements specified for the tiers and the identified node attributes." FEDEX-1005, 9:3-13. A POSITA would have understood or it would have been obvious to a POSITA that Sparks's control node checks the compatibility (*e.g.*, checking node/tier requirements) between a node and a tier before assigning the node to the tier. FEDEX-1003, ¶159; *see also* §V.A.4, [1a-i], [1b], [1c].

**[9a-ii] the cluster module initializing a computing cluster in the first computing node,**

Sparks-Bram renders obvious [9a-ii]. FEDEX-1003, ¶160; §V.A.4, [1b].

**[9a-iii] the cluster module adding one or more available computing nodes to the computing cluster;**

Sparks-Bram renders obvious [9a-iii]. FEDEX-1003, ¶161; §V.A.4, [1c].

**[9b-i] a node module stored on a non-transitory computer readable medium, the node module configured to manage compatibility operations associated with the at least two computer nodes,**

Sparks-Bram renders obvious [9b-i]. FEDEX-1003, ¶¶162-164. As discussed above for [9a-i], Sparks discloses a non-transitory computer readable medium. FEDEX-1005, 1:65-2:7, 3:23-25. A POSITA would have recognized that Sparks discloses, or at least renders obvious, a node module stored on a non-transitory computer readable medium because instructions corresponding to the relevant functionality of Sparks's control node in the Sparks-Bram Combination can be stored on Sparks's computer-readable media. FEDEX-1003, ¶163; §V.A.4, [1a-i], [1a-ii]; §V.A.12, [9pre] (addressing "modules").

Sparks discloses, or at least renders obvious, that the node module is configured to manage compatibility operations (*e.g.*, storing node/tier requirements/constraints) associated with the at least two computer nodes (*e.g.*, one or more control nodes and application nodes). FEDEX-1003, ¶164; §V.A.12, [9pre]. Sparks's control node "maintain[s] a database having entries for each node slot,

where the entries identify the node requirements for each of the tiers” and “retrieves the node requirements for the selected tier from the database.” FEDEX-1005, 9:22-27. *See also id.*, 1:51-56, 23:15-20. A POSITA would have understood, or it would have been obvious to a POSITA, that storing node/tier requirements/constraints are examples of compatibility operations because such requirements/constraints are used to check compatibility. FEDEX-1003, ¶164. Additional details regarding the stored information, including license information, are provided at [9b-ii]-[9b-iii].

**[9b-ii] the node module storing license information for the computing cluster,**

Sparks-Bram renders obvious [9b-ii]. FEDEX-1003, ¶165; §V.A.4, [1a-i].

**[9b-iii] wherein the licensing information includes a bundle-type parameter that identifies a size of the computing cluster and at least one additional bundle-type characteristic of the computing cluster, and one or more node license parameters that identifies a characteristic of a first computing node; and**

Sparks-Bram renders obvious [9b-iii]. FEDEX-1003, ¶166; §V.A.4, [1a-ii].

**[9c-i] a management module stored on a non-transitory computer readable medium, the management module configured to manage compatibility operations with one or more external servers,**

Sparks-Bram renders obvious [9c-i]. FEDEX-1003, ¶¶167-171. As discussed above for [9a-i], Sparks discloses a non-transitory computer readable medium. FEDEX-1005, 3:23-25, 1:65-2:7. A POSITA would have recognized that Sparks discloses, or at least renders obvious, a management module stored on a non-transitory computer readable medium because instructions corresponding to the

relevant functionality of Sparks's control node in the Sparks-Bram Combination can be stored on Sparks's computer-readable media. FEDEX-1003, ¶168; §V.A.4, [1d]; §V.A.12, [9pre] (addressing "modules").

Sparks-Bram renders obvious that the management module is configured to manage compatibility operations with one or more external servers, as claimed. FEDEX-1003, ¶169. Bram discloses multiple servers including primary and backup license servers. FEDEX-1006, [0252] ("The system 10 can include multiple license servers" which "can include a primary license server and one or more backup license servers"), [0257] ("the license server 940 can set itself as a backup server and can subscribe (e.g., create a subscriber process or thread) to the license server that has published itself as the primary license server"), [0232] ("Multiple clusters can run simultaneously at multiple locations in order to provide mutual backup options for a single system"), [0258], [0259], Fig. 28G; *see also* §V.A.5, [2].

It would have been obvious to modify Sparks's control node to manage compatibility operations by communicating with one or more external servers, such as those taught by Bram, to obtain information pertaining to compatibility such as licensing information. FEDEX-1003, ¶¶169-171. In addition to the reasons in Section V.A.3, a POSITA would have been motivated to modify Sparks's control node to communicate with one or more external servers for backup, redundancy,

and/or disaster recovery reasons. FEDEX-1003, ¶171; FEDEX-1006, [0232], [0252], [0304].

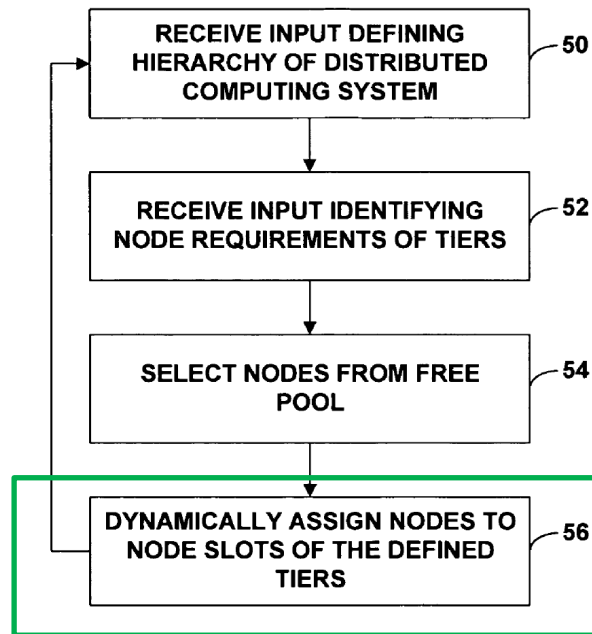
**[9c-ii]the management module activating the computing cluster when the computing cluster includes a number of nodes that complies with the bundle-type parameter.**

Sparks-Bram renders obvious [9c-ii]. FEDEX-1003, ¶172; §V.A.4, [1d].

### 13. Claim 10

**[10] The computing system of claim 9, wherein the cluster module comprises a create cluster module to aggregate the at least two computing nodes into a clustered computing system.**

Sparks-Bram renders obvious [10]. FEDEX-1003, ¶¶173-177. For example, Sparks discloses a create cluster module (*e.g.*, control node) to aggregate the at least two computing nodes into a clustered computing system (*e.g.*, tier), as claimed. *Id.* Sparks discloses that the “[c]ontrol node 12 dynamically *assigns computing nodes to the node slots of each tier* based on the node requirements specified for the tiers and the identified node attributes (56),” which is shown in Fig. 3. FEDEX-1005, 9:3-13; §V.A.12, [9pre] (addressing “modules”).



FEDEX-1005, Fig. 3 (annotated), 8:37-39 (Fig. 3 of Sparks shows the “operation of control node 12 when configuring distributed computing system 10”). *See also* §V.A.4, [1c].

Sparks describes that the control node aggregates unallocated computing nodes from a free pool into a clustered computing system. FEDEX-1003, ¶176; FEDEX-1005, 7:1-14 (“[c]ontrol node 12 automatically *assigns unallocated nodes from free pool 13 to respective tier node slots of a tier*”), 7:15-25 (“control node 12 may automatically *add unallocated nodes* from free pool 13 *to a tier*”), 8:14-25 (“control node 12 may also *add nodes from free pool 13 to a tier* when more processing capacity is needed within the tier”), 3:40-48, 8:61-9:2, 9:28-30.

Spark discloses that a tier can be enabled when the node count for that tier is met. FEDEX-1003, ¶177; FEDEX-1005, 9:14-21 (discussing that “control node 12

selects a tier to enable (60)”), 9:52-62 (“determines whether the node count for the tier is met (74)”), Fig. 4. For example, if the minimum node count is two, then when two computing nodes have been added to a tier, it can be enabled. FEDEX-1003, ¶177. *See also* §V.A.4, [1c], [1d].

#### 14. Claim 11

**[11] The computing system of claim 10, wherein the cluster module maintains a right to use license comprising licensing information for a computing cluster and a node license pool comprising licensing information for nodes in the computing cluster.**

Sparks-Bram renders obvious [11]. FEDEX-1003, ¶¶178-180. Bram discloses maintaining a right to use license comprising licensing information for a computing cluster, as claimed. FEDEX-1003, ¶179; §V.A.4, [1a-i], [1a-ii]. Bram’s license key file “provides authorization to apply the available licenses if constraints of the available licenses are satisfied.” FEDEX-1006, [0245]. Accordingly, the key file “can specify parameters and/or constraints for each available license, such as an effective date, an expiration date, a total concurrent activations limit or count, a list of authorized named nodes, a list of authorized operating systems (or operating system aliases), a list of authorized MAC codes, a list of authorized JVMs, and/or an activation mode (e.g., ‘restricted’ or ‘unrestricted’) of components allowed under a license.” *Id.* Bram further discloses that “the license server 940 can check static constraints (e.g., host restrictions) and dynamic constraints (e.g., counts of the

number of instances of a component in the system 10 and/or on a specific node) for a required license associated with a component.” *Id.*, [0275].

Bram further discloses a node license pool comprising licensing information for nodes in the computing cluster. FEDEX-1003, ¶180. Bram discloses, for example, a pool of floating licenses that can be assigned to one or more nodes. FEDEX-1006, [0255] (“[t]he floating license information can include all licenses and license constraint information currently available to the system 10 (e.g., those licenses purchased and/or obtained by a system user and the related constraints as defined in the license key file 952)”; “[t]he license server 940 can use the floating license information to authorize or decline activation authorization requests generated by components of the system 10 and track applied licenses”), [0258], [0264]; §V.A.4, [1a-i], [1a-ii].

## 15. Claim 12

**[12] The computing system of claim 10, further comprising an add node module to add a computing node to the clustered computing system.**

Sparks-Bram renders obvious [12]. FEDEX-1003, ¶¶181-182. Sparks discloses an add node module (e.g., control node) to add a computing node to the clustered computing system (e.g., tier). Sparks discloses “*adding an additional computing node to a tier* to meet increased processing demands” and that “[w]hen there are one or more nodes that meet the minimum node requirements of the tier,

control node 12 selects the node from the free pool based the node requirements of the tier, as described above, (94) and *assigns the node to the tier* (95).” FEDEX-1005, 10:35-62; Fig. 5. *See also* §§V.A.11-13.

## 16. Claim 13

**[13] The computing system of claim 10, wherein the management module provides a license key to and receives licensing information from an external licensing server.**

Sparks-Bram renders obvious [13]. FEDEX-1003, ¶¶183-190. For example, Bram discloses external licensing servers including primary and backup license servers. FEDEX-1006, [0252] (“The system 10 can include multiple license servers” which “can include a primary license server and one or more backup license servers”), [0257] (“the license server 940 can set itself as a backup server and can subscribe (e.g., create a subscriber process or thread) to the license server that has published itself as the primary license server”), [0232] (“Multiple clusters can run simultaneously at multiple locations in order to provide mutual backup options for a single system”), [0258], [0259], Fig. 28G. *See also* §V.A.5, [2].

Bram further discloses that a client, to activate license covered components, broadcasts a get-key request in order to obtain a decryption key from a license server (conditioned on a license check) to decrypt encrypted files (encrypted because they are covered by a license). FEDEX-1006, [0005], [0263]-[0266]; FEDEX-1003, ¶185. The request can include a node name/identity and/or enciphered control data.

FEDEX-1006, [0270] (“the activation authorization request also includes environmental parameters of the application 16, such as *the name or identifier of a node* executing the application 16, which may be needed to determine whether a required license associated with a particular component is available”; “[t]he activation authorization request can also include *enciphered control data* that the license server 940 can use to authenticate the activation authorization request”), [0271]. A POSITA would have recognized that the node name/identity and the enciphered control data are examples of the claimed license key because each can be used to “unlock” a license, *i.e.*, activate a license covered component. FEDEX-1003, ¶¶185-187. In addition to the reasons in Section V.A.3, a POSITA would have been motivated to modify Sparks’s control node to communicate with an external licensing server to activate a license covered component for backup, redundancy, and/or disaster recovery reasons. FEDEX-1003, ¶187; FEDEX-1006, [0232], [0252], [0304].

Bram’s license server can “check that the application 16 that generated the get-key request meets the license constraints of the basic license” and can “verify that the node executing the application 16 that generated the get-key request is a valid node.” FEDEX-1006, [0265]. After a successful verification, the license server sends a response authorizing the activation of the component --the response includes a decryption key that is used to “decrypt encrypted files (e.g., encrypted JAR files)”

associated with the component. *Id.*, see also [0243], [0276]; FEDEX-1003, ¶¶188-189 (explaining that encryption, *e.g.*, Bram’s encrypted JAR files, of a software component prevents it from running; and that the decryption key allows the encrypted JAR files to be decrypted so that they can run). Bram explains that the decryption keys are stored in the software licensing authentication token. FEDEX-1006, [0245]-[0246]. As such, a POSITA would have understood that a decryption key is an example of licensing information. FEDEX-1003, ¶189.

Moreover and alternatively, a POSITA would have understand that other information included in Bram’s disclosed response are also examples of licensing information such as a “digital signature to verify that the reply is from an authorized license server” or a restriction associated with the activation of a component. FEDEX-1003, ¶190; FEDEX-1006, [0273] (“the license server 940 can digitally sign the reply, and the application 16 and/or the application licenser component can use the digital signature to verify that the reply is from an authorized license server”), [0276] (“[t]he restricted activation authorized response can indicate that a particular component can be activated, but must be activated with limited functionality”).

## **VI. PAYMENT OF FEES—37 C.F.R. §42.103**

Petitioners authorize the Patent and Trademark Office to charge any fees to Deposit Account No. 06-1050.

**VII. MANDATORY NOTICES—37 C.F.R. § 42.8(a)(1)**

**A. Real Party-In-Interest—37 C.F.R. § 42.8(b)(1)**

Petitioners, FedEx Corporation and FedEx Office and Print Services, Inc., are the real parties-in-interest. Additionally, even though Petitioners do not believe that these additional entities are real parties-in-interest, Petitioners further identify Federal Express Corporation and FedEx Dataworks in the interest of completeness.

**B. Related Matters—37 C.F.R. § 42.8(b)(2)**

Petitioners are not aware of any disclaimers, reexamination certificates or petitions for inter partes review for the '416 Patent. The '416 Patent is the subject of civil action: *Starbucks Corporation v. Valtrus Innovations Limited et al.*, 2-25-cv-01859 (WDWA) Sep. 25, 2025; *NetApp, Inc. v. Valtrus Innovations Ltd. et al.*, 1-25-cv-06138 (SDNY) July 25, 2025; *Valtrus Innovations Ltd. et al. v. FedEx Corporation et al.*, 2-25-cv-00516 (EDTX), May 9, 2025 and *Valtrus Innovations Ltd. et al v. NetApp, Inc.*, 2-25-cv-00517 (EDTX) May 9, 2025.

**C. Lead And Back-Up Counsel—37 C.F.R. § 42.8(b)(3)**

Petitioners provide the following designation of counsel.

| Lead Counsel                                                                                                                                                                                                                                            | Backup counsel                                                                                                                                                                                                                                                                  |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Kenneth W. Darby, Jr., Reg. No. 65,068<br>Fish & Richardson P.C.<br>60 South Sixth Street, Suite 3200<br>Minneapolis, MN 55402<br>Tel: 202-783-5070<br>Fax: 877-769-7945<br>Email: <a href="mailto:IPR39898-0049IP1@fr.com">IPR39898-0049IP1@fr.com</a> | David Hoffman, Reg. No. 54,174<br>John-Paul Fryckman, Reg. No. 62,880<br>Fish & Richardson P.C.<br>60 South Sixth Street, Suite 3200<br>Minneapolis, MN 55402<br>Tel: 202-783-5070<br>Fax: 877-769-7945<br><a href="mailto:IPR39898-0049IP1@fr.com">IPR39898-0049IP1@fr.com</a> |

**D. Service Information**

Please address all correspondence and service to the address listed above.

Petitioner consents to electronic service by email at [IPR39898-0049IP1@fr.com](mailto:IPR39898-0049IP1@fr.com).

Respectfully submitted,

Dated: October 14, 2025

/Kenneth Wayne Darby Jr./  
Kenneth W. Darby, Jr., Reg. No. 65,068  
David Hoffman, Reg. No. 54,174  
John-Paul Fryckman, Reg. No. 62,880  
Fish & Richardson P.C.  
60 South Sixth Street, Suite 3200  
Minneapolis, MN 55402  
T: 202-783-5070  
F: 877-769-7945

(Control No. IPR2026-00060)

*Counsel for Petitioners*

**CERTIFICATION UNDER 37 CFR § 42.24**

Under the provisions of 37 CFR § 42.24(d), the undersigned hereby certifies that the word count for the foregoing Petition for *Inter partes* Review totals 13,822 words, which is less than the 14,000 allowed under 37 CFR § 42.24.

Dated: October 14, 2025

/Kenneth Wayne Darby Jr./

Kenneth W. Darby, Jr., Reg. No. 65,068

David Hoffman, Reg. No. 54,174

John-Paul Fryckman, Reg. No. 62,880

Fish & Richardson P.C.

60 South Sixth Street, Suite 3200

Minneapolis, MN 55402

T: 202-783-5070

F: 877-769-7945

*Counsel for Petitioners*

**CERTIFICATE OF SERVICE**

Pursuant to 37 CFR §§ 42.6(e)(4)(i) *et seq.* and 42.105(b), the undersigned certifies that on October 14, 2025, a complete and entire copy of this Petition for *Inter partes* Review, Power of Attorney, and all supporting exhibits were provided via Federal Express, to the Patent Owner by serving the correspondence address of record as follows

Botos Churchill IP Law LLP - Malik  
430 Mountain Avenue  
Suite 401  
New Providence, NJ 07974  
(908) 738-1770

/Crena Pacheco/  
\_\_\_\_\_  
Crena Pacheco  
Fish & Richardson P.C.  
60 South Sixth Street, Suite 3200  
Minneapolis, MN 55402  
[pacheco@fr.com](mailto:pacheco@fr.com)