Paper No. __

UNITED STATES PATENT AND TRADEMARK OFFICE

_____

BEFORE THE PATENT TRIAL AND APPEAL BOARD

_____

TESSELL, INC.,
Petitioner,

v.

NUTANIX, INC.,
Patent Owner.

_____

Case No. IPR2025-00298
Patent No. 11,860,818

_____

PETITION FOR INTER PARTES REVIEW
UNDER 35 U.S.C. §§ 311-319 AND 37 C.F.R. § 42.1 et seq

TABLE OF CONTENTS

MANDATORY NOTICES....x

A. Real Party-In-Interest....x

B. Related Matters....x

1. United States Patent & Trademark Office....x

2. U.S. District Court for the Northern District of California....x

C. Counsel and Service Information - §42.8(b)(3) and (4)....xi

I. INTRODUCTION....1

II. STANDING CERTIFICATION....5

III. UNPATENTABILITY GROUNDS....6

IV. THE '818 PATENT....6

A. Technical Background and Described Embodiments....6

B. Challenged Claims....11

C. Prosecution History....12

D. Person of Ordinary Skill in the Art ("POSA")....13

V. CLAIM INTERPRETATION....13

VI. GROUND 1: Claims 1-36 ARE UNPATENTABLE OVER SIVASUBRAMANIAN+ ZHA....13

A. Sivasubramanian (EX1005)....13

1. Overview....14

a. UIs for "Create Database" Workflow....19

i. GUI....20

ii. Command-Line....24

b. Backup "Workflow"....25

2. Motivation to Customize Sivasubramanian's UI....26

B. Zha (EX1006)....27

C. Sivasubramanian+Zha Combination....29

1. UI for retention period....35

2. UI for schedule....38

D. Claim-by-Claim Analysis 39
1. Claim 1 39
a. [1Pre]: "A non-transitory computer-readable media comprising computer-readable instructions stored thereon for provisioning a database, wherein the computer-readable instructions when executed by a processor cause the processor to:" 39
b. [1A]: "receive a first input of a database name for the database;" 42
i. GUI 42
ii. Command Line 43
iii. Order of Inputs Irrelevant 44
c. [1B]: "receive a second input of a database engine for the database;" 45
i. GUI 45
ii. Command Line 46
d. [1C]: "receive a third input indicative of a location where the database is to be provisioned" 46
i. GUI 46
ii. Command Line 48
e. [1D]: "receive a fourth input comprising a selection of a level of a Service Level Agreement (SLA) from a plurality of levels of the SLA to protect the database after provisioning" 49
i. GUI 49
ii. Command Line 52
f. [1E]: "receive a fifth input of a protection schedule comprising a first frequency of capturing a snapshot of the database after provisioning and at least one time period for capturing the snapshot;" 52
i. GUI 52
ii. Command Line 54

g. [1F]: "present at least one user interface to receive the first input, the second input, the third input, the fourth input, and the fifth input;" 54
i. GUI 54
ii. Command Line 55
h. [1G]: "determine an amount of storage for the database; and" 55
i. [1H]: "create the database at the location based on the first input, the second input, the third input, the fourth input, the fifth input, and the amount of storage" 57
2. Claim 2 58
3. Claim 3 60
4. Claim 4 61
5. Claim 5 61
6. Claim 6 62
7. Claim 7 63
8. Claim 22 64
9. Claim 23 66
10. Claim 28 67
11. Claim 29 69
12. Claim 30 72
13. Claims 8-14, 24-25, 31-33 73
14. Claims 15-21, 26-27, 34-36 74
VII. Ground 2: Claims 1-36 Are Unpatentable Over Sivasubramanian+Shekar+Zha 75
A. Shekar (EX1007) 75
B. Sivasubramanian+Shekar+Zha Combination 77
C. Claim-by-Claim Analysis 80
1. Claims 1, 3-8, 10-15, 17-36 80
2. Claims 2, 9, 16 80
VIII. NO BASIS EXISTS FOR DISCRETIONARY DENIAL 81

A. 315(a) 81
1. Stay Potential 81
2. Trial Timing 81
3. Litigation Investment 82
4. Issue overlap 82
5. Litigation Defendants 82
6. Other Circumstances 82
B. 325(d) 83
1. Part One 83
2. Part Two 83
IX. CONCLUSION 85
X. CLAIM LISTING 86

TABLE OF AUTHORITIES

CASES

Advanced Bionics, LLC v. Med-El Elektromedizinische Gerate GMBH,
IPR2019-01469, Paper 6 (Feb. 13, 2020)..... 83, 84

Apple Inc. v. Fintiv, Inc.,
IPR2020-00019, Paper 11 (Mar. 20, 2020)..... 81, 82

Apple v. Seven Networks,
IPR2020-00156, Paper 10 (June 15, 2020) 82

Becton, Dickinson & Co. v. B. Braun Melsungen AG ,
IPR2017-01586, Paper 8 (Dec. 15, 2017) 83, 84

Cellco P'ship v. Huawei Device Co.,
IPR2020-01117, Paper 10 (Feb. 3, 2021).....84

Group III v. Targus,
IPR2021-00371, Paper 21 (July 9, 2021).....84

Sand Revolution II v. Continental Intermodal Group-Trucking,
IPR2019-01393, Paper 24 (June 16, 2020) 81, 82

REGULATIONS

37 C.F.R. §42.100(b) 13

37 C.F.R. §42.104(a).....5

STATUTES

35 U.S.C. §102(a) 6

35 U.S.C. §103.....6

35 U.S.C. §282(b) 13

35 U.S.C. §315(a) 82

35 U.S.C. §325(d) 84

OTHER AUTHORITIES

Director's Interim Procedure for Discretionary Denials (June 21, 2022) 82, 83

PTAB Rules for 325(d) Considerations,
89 Fed. Reg. 28693 (Apr. 19, 2024).....85

APPENDIX LISTING OF EXHIBITS

Exhibit	Description
1001	U.S. Patent No. 11,860,818
1002	Prosecution History of U.S. Patent No. 11,860,818
1003	Declaration of Dr. Samrat Bhattacharjee ("Bhattacharjee")
1004	Curriculum Vitae of Dr. Bhattacharjee
1005	U.S. Patent No. 8,713,060 ("Sivasubramanian")
1006	U.S. Patent No. 8,150,808 ("Zha")
1007	U.S. Patent App. Pub. No. 2018/0232142 ("Shekar")
1008	U.S. Patent No. 6,542,893 ("Quernemoen")
1009	The American Heritage Dictionary of the English Language (5th ed. 2011) ("Heritage")
1010	U.S. Patent No. 7,873,684 ("Souder")
1011	U.S. Patent App. Pub. No. 2018/0324279 ("Barton")
1012	Using Oracle Database Cloud Service – Creating a Database Deployment (2017) ("Oracle") (Wayback Machine Capture, January 18, 2017), available at https://web.archive.org/web/20170118071307/https://docs.oracle.com/en/cloud/paas/database-dbaascloud/csdbi/create-db-deployment.html
1013	Amazon Relational Database Service User Guide (2016) ("RDS") (Wayback Machine Capture, April 12, 2016), available at https://web.archive.org/web/20160412105355/http://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/rds-ug.pdf
1014	U.S. Patent App. Pub. No. 2009/0319582 ("Simek")
1015	U.S. Patent No. 7,331,000 ("Ohno")
1016	U.S. Patent App. Pub. No. 2016/0182328 ("Bhasin")
1017	U.S. Patent App. Pub. No. 2016/0077925 ("Tekade")
1018	U.S. Patent App. Pub. No. 2019/0324865 ("Weissman")
1019	U.S. Patent App. Pub. No. 2016/0335369 ("Picard")
1020	U.S. Patent No. 11,308,114 ("Moghe")
1021	U.S. Patent App. Pub. No. 2014/0229698 ("Sivasubramanian-698")
1022	"Taking a Backup Using Snapshots" ("LVM-Howto") (Wayback Machine Capture, January 20, 2007), available at https://web.archive.org/web/20070120200004/https://tldp.org/HOWTO/LVM-HOWTO/snapshots_backup.html
1023	U.S. Patent App. Pub. No. 2004/0059878 ("Madany")
1024	U.S. Patent App. Pub. No. 2006/0161879 ("Lubrecht")
1025	PCT Pub. No. WO 2010/106578 ("Utsunomiya")

1026	U.S. Patent No. 10,855,554 (“Freeman”)
1027	U.S. Patent No. 11,182,372 (“Jain”)
1028	U.S. Patent App. Pub. No. 2016/0057077 (“Gomatam”)
1029	U.S. Patent No. 11,036,696 (“Higginson”)
1030	U.S. Patent No. 9,305,068 (“Esposito”)
1031	U.S. Patent No. 7,873,610 (“Poulsen”)
1032	“Creating Instances,” Google Cloud (“Google”) (Wayback Machine Capture, September 18, 2018), available at https://www.archive.org/web/20180918103721/https://cloud.google.com/sql/docs/mysql/create-instance
1033	U.S. Patent App. Pub. No. 2003/0046353 (“Chung”)
1034	U.S. Patent App. Pub. No. 2006/0253472 (“Wasserman”)
1035	U.S. Patent App. Pub. No. 2005/0027661 (“Lober”)
1036	U.S. Patent App. Pub. No. 2018/0173452 (“Hu”)
1037	U.S. Patent No. 10,078,465 (“Wu”)
1038	U.S. Patent No. 9,830,082 (“Srinivasan”)
1039	U.S. Patent App. Pub. No. 2016/0266913 (“Achillopoulos”)
1040	U.S. Patent App. Pub. No. 2017/0262232 (“Zhao”)
1041	U.S. Patent App. Pub. No. 2019/0116237 (“Gibson”)
1042	U.S. Patent No. 10,206,092 (“Mellquist”)
1043	U.S. Patent No. 11,243,703 (“Watkins”)
1044	U.S. Patent App. Pub. No. 2015/0207683 (“Adogla”)
1045	U.S. Patent App. Pub. No. 2014/0018947 (“Ales”)
1046	U.S. Patent App. Pub. No. 2013/0332251 (“Ionnidis”)
1047	U.S. Patent App. Pub. No. 2006/0107006 (“Green”)
1048	U.S. Patent App. Pub. No. 2008/0059894 (“Cisler”)
1049	U.S. Patent App. Pub. No. 2013/0110779 (“Taylor”)
1050	U.S. Patent App. Pub. No. 2016/0306840 (“Deshmukh”)
1051	U.S. Patent App. Pub. No. 2018/0121521 (“Bivol”)
1052	U.S. Patent No. 10,496,302 (“Gaurav”)
1053	U.S. Patent No. 10,268,551 (“Wong”)
1054	U.S. Patent App. Pub. No. 2014/0108339 (“Marsden”)
1055	U.S. Patent No. 8,335,765
1056	Scheduling Order in *Nutanix, Inc. v. Tessell, Inc.*, Case No. 3:24-cv-01729 (N.D. Cal. Mar. 20, 2024)
1057	Docket Sheet of *Nutanix, Inc. v. Tessell, Inc.*, Case No. 3:24-cv-01729 (N.D. Cal.)
1058	Affidavit of Nathaniel E. Frank-White

1059	Southwest Airlines website (Wayback Machine Capture, June 1, 2018), available at https://web.archive.org/web/20180601002749/https://www.southwest.com/
1060	U.S. Patent No. 11,120,011 ("Priebe")
1061	U.S. Patent No. 7,634,679 ("Quintanillo")

MANDATORY NOTICES

A. Real Party-In-Interest

Petitioner is the Real Party-in-Interest.

B. Related Matters

1. United States Patent & Trademark Office

The application from which U.S. Patent No. 11,860,818 issued is a Continuation of U.S. Patent Application No. 17/237,599 (U.S. Patent No. 11,604,762), filed April 22, 2021, which is a Continuation of U.S. Patent Application No. 16/234,553 (U.S. Patent No. 11,010,336), filed December 27, 2018.

The following U.S. patent applications claim the benefit of priority to U.S. Patent 11,860,818:

(i) U.S. Patent Application No. 18/489,837 (U.S. Patent No. 12,026,124), filed October 18, 2023;

(ii) U.S. Patent Application No. 18/740,432, filed June 11, 2024.

2. U.S. District Court for the Northern District of California

The ’818 patent is currently asserted in *Nutanix, Inc. v. Tessell, Inc.*, Case No. 3:24-cv-01729 (NDCA).

C. Counsel and Service Information - §42.8(b)(3) and (4)

Lead Counsel	Anant K. Saraswat, Reg. No. 76,050
Backup Counsel	Elisabeth H. Hunt, Reg. No. 67,336 Daniel D. Smith, Reg. No. 71,278
Service Information	E-mail: ASaraswat-PTAB@wolfgreenfield.com EHunt-PTAB@wolfgreenfield.com DSmith-PTAB@wolfgreenfield.com Post and hand delivery: Wolf, Greenfield & Sacks, P.C. 600 Atlantic Avenue Boston, MA 02210-2206 Telephone: 617-646-8000 Facsimile: 617-646-8646

A power of attorney is submitted with the Petition. Counsel for Petitioner consents to service of all documents via electronic mail.

I. INTRODUCTION

Petitioner requests *inter partes* review and cancellation of claims 1-36 of U.S. Patent No. 11,860,818 ("the '818 patent," EX1001).

The '818 patent discloses a "database system" that receives a "user request" via a "dashboard" for "provisioning" a "database." EX1001, Abstract, 1:33-37, claims 1, 8, 15 (reciting "instructions… to provision a database"); "Bhattacharjee (EX1003), ¶27. "Provisioning" may involve either "creating" a database with user-specified parameters or "associating" an existing database with the database system. EX1001, 3:41-43; Bhattacharjee, ¶28.

The patent purportedly provides a "user friendly, intuitive user interface [UI] that solicits information from and conveniently walks the user through the various steps for creating a new database." EX1001, 4:17-20; Bhattacharjee, ¶46. The information solicited includes parameters such as the database software type ("engine"), database name, and a frequency for taking "snapshots," *i.e.*, point-in-time records of the database's state. EX1001, 14:20-31, 24:61-25:14, 6:5-8, 4:55-5:1, Figs. 4C, 4E, 4F; Bhattacharjee, ¶¶38-40. The patent's UI purportedly enables database creation "within minutes," while database creation in prior-art systems allegedly required "multiple days" of manual work by "multiple entities." EX1001, 3:41-64, 4:17-28; Bhattacharjee, ¶46. *See also* EX1001, Abstract, Figs. 4C, 4E.

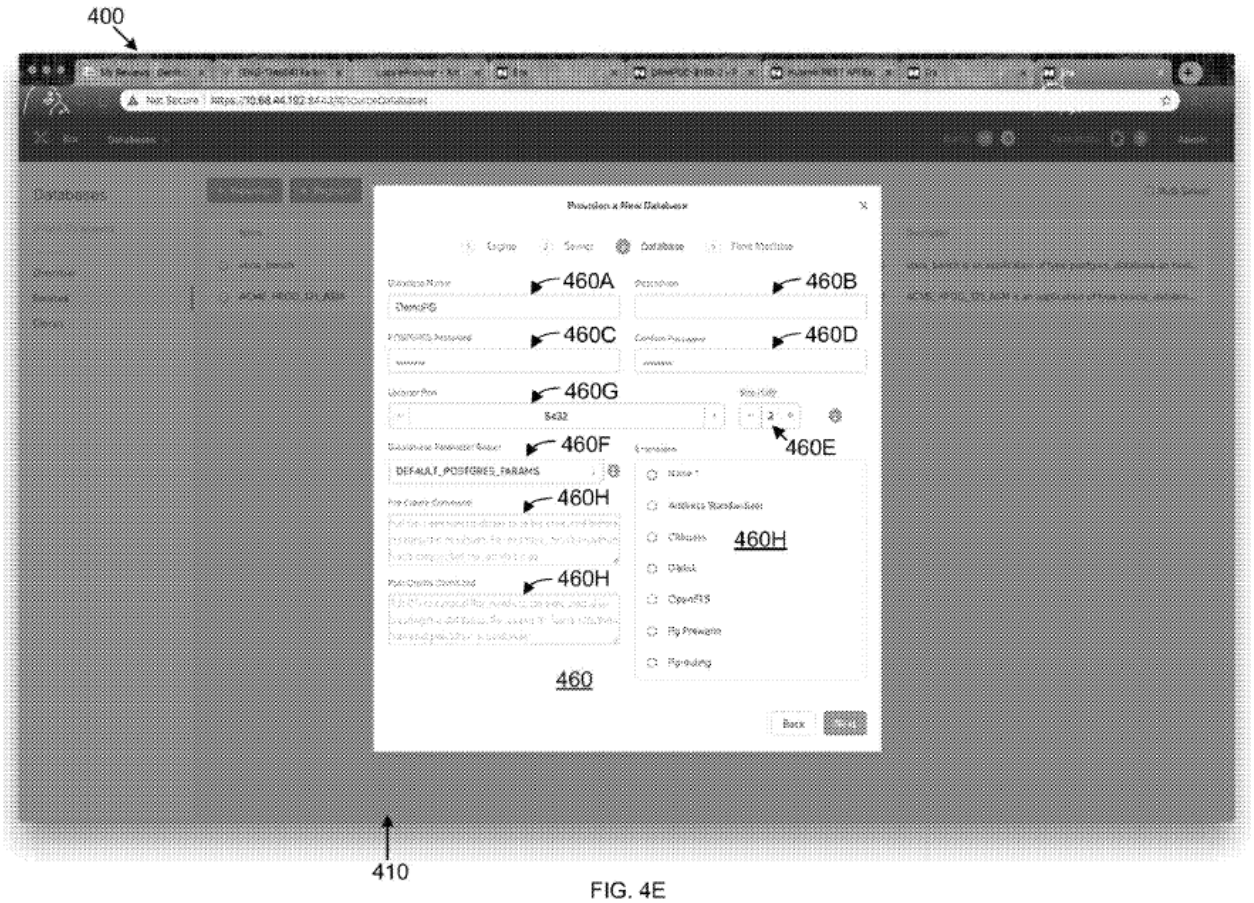


FIG. 4E

EX1001, Fig. 4E[1]

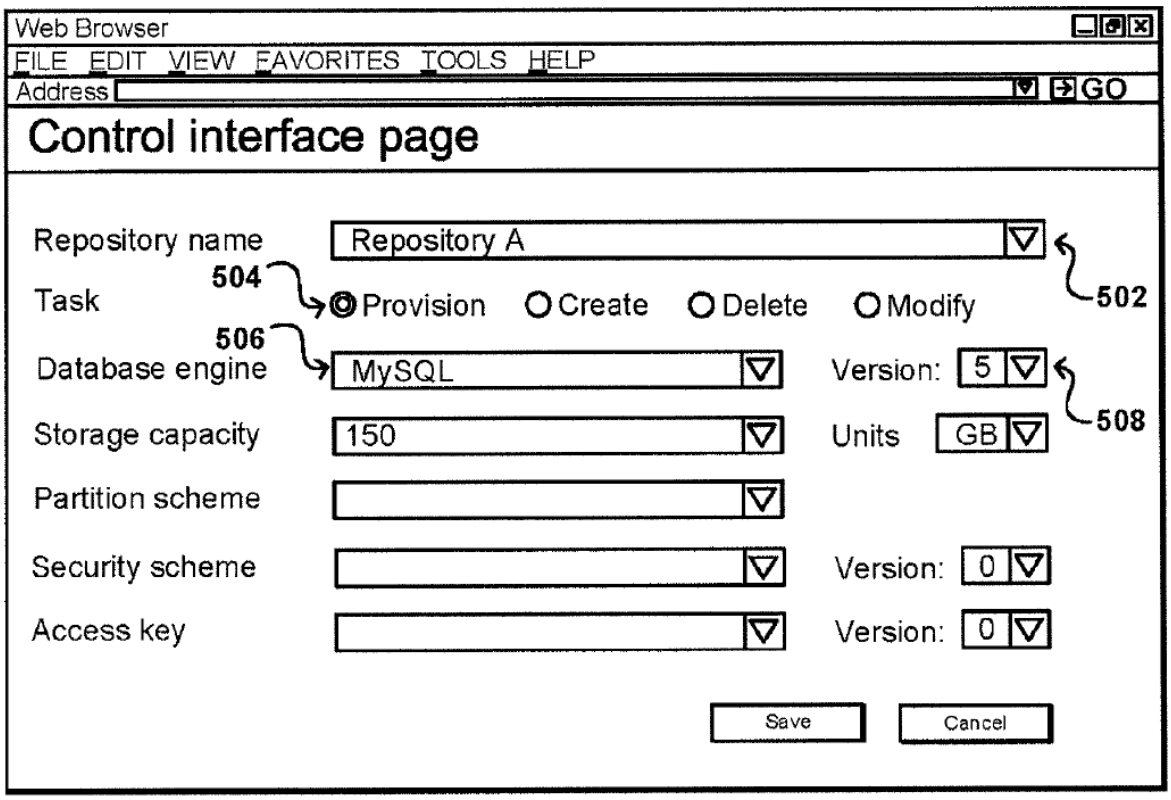
Notably, the '818 patent ***does not*** purport to disclose any novel database implementation, *e.g.*, any new database software or any new data storage technology. Bhattacharjee, ¶52. Rather, the patent's claims recite simply accepting various parameters via a UI and creating a database based on those parameters.

As this Petition demonstrates, all the input parameters recited in the claims were well-known database parameters, as disclosed in, *e.g.*, Sivasubramanian

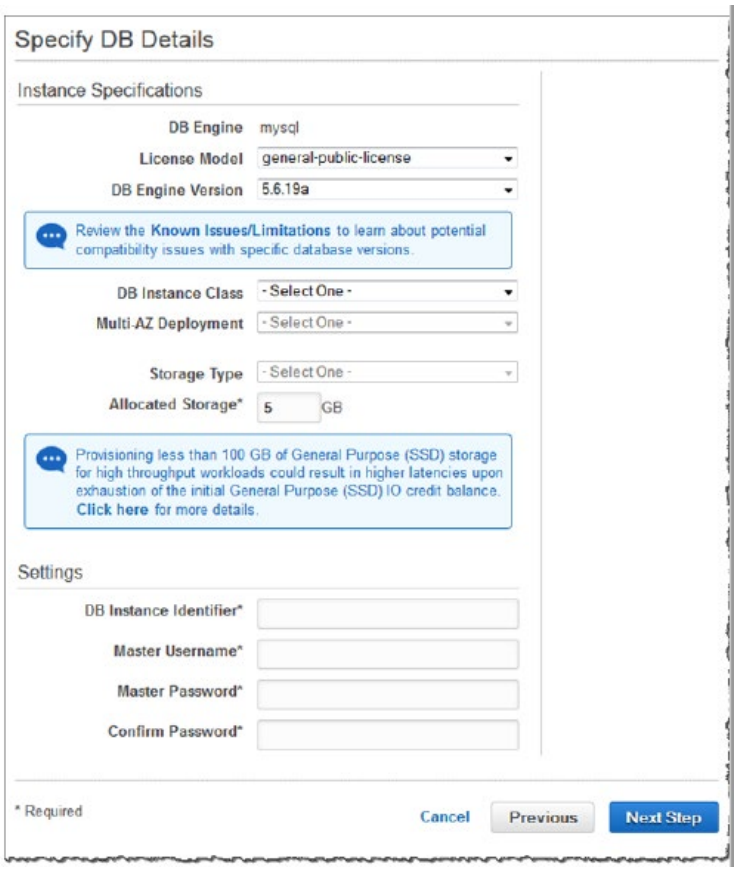
[1] The blurry text is in the '818 patent's original figures.

(EX1005) and Zha (EX1006). Nor was there anything novel about receiving the recited parameters as inputs and creating a database based on them. As discussed below, years before the claimed 2018 priority date, UIs providing for convenient and quicker parameter input and database creation were known.

For example, ***in 2009***, Sivasubramanian, assigned to Amazon, disclosed a "control plane" that automates various "workflows" to replace "tedious manual procedures" previously required for tasks such as "provisioning and scaling data storage." Sivasubramanian, Abstract, 1:46-56, 2:17-21; Bhattacharjee, ¶47. In contrast to the previously-used manual procedures, which "[took] a significant amount of time and effort," a user could use a UI to "submit a call including information such as a type of hardware and a version of a database product" to Sivasubramanian's control plane, which then created the database; thus, Sivasubramanian provided the same functionality ***9 years earlier*** than the '818 patent. Sivasubramanian, 9:49-61, Fig. 5; Bhattacharjee, ¶48. Similar database-creation UIs were used in commercial products such Amazon's Relational Database Service and Google's Cloud SQL. Amazon Relational Database Service User Guide ("RDS," EX1013), 13-18, 22; Google (EX1032), 2-3; Bhattacharjee, ¶¶49-50.



Sivasubramanian, Fig. 5



RDS, 22

As this Petition demonstrates, Sivasubramanian's database-creation UI was customizable to accept any suitable user-desired parameters, such as by supplementing Sivasubramanian's disclosed parameters with parameters disclosed in Zha. All the '818 patent's claims would have been obvious as demonstrated below.

II. STANDING CERTIFICATION

Petitioner certifies that the '818 patent is available for *inter partes* review and that Petitioner is not barred or estopped from requesting *inter partes* review of the challenged claims. 37 C.F.R. §42.104(a).

III. UNPATENTABILITY GROUNDS

The table below identifies the references, applicable claims, and basis for each unpatentability ground. Sivasubramanian, Zha, and Shekar (EX1007) are prior art under post-AIA §§102(a)(1)-(2). Zha and a descendant of Sivasubramanian were of record during prosecution, but as discussed *infra* §§IV.B, VIII.B, neither was substantively discussed or applied in any rejection, and the Sivasubramanian+Zha combination was never considered.

Ground Number and Reference(s)		Claims	Basis
1	Sivasubramanian, Zha	1-36	§103
2	Sivasubramanian, Shekar, Zha	1-36	§103

IV. THE '818 PATENT

A. Technical Background and Described Embodiments

The '818 patent discloses a "database system." EX1001, Abstract. Databases (which long predated the '818 patent) are collections of organized data. Bhattacharjee, ¶¶27-30; Quernemoen (EX1008); Heritage (EX1009), 462. Databases can be created and managed by different types of database software, which the patent refers to as "database engine types," examples of which include Oracle and MySQL. Bhattacharjee, ¶30 ; EX1001, 14:20-24; Sivasubramanian, 2:44-47.

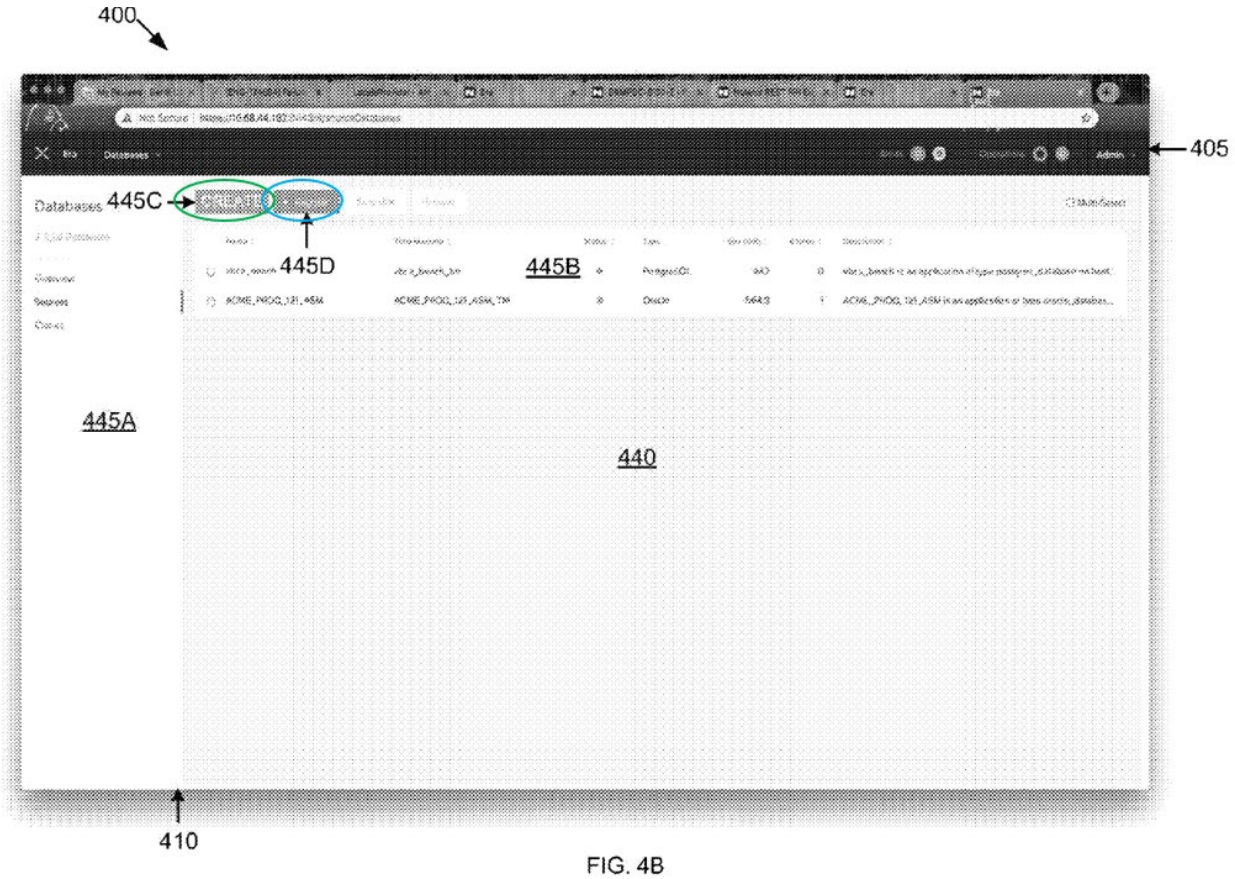
The '818 patent's "database system" includes a "database provisioning system" providing "database provisioning services." EX1001, 10:51-64;

Bhattacharjee, ¶31. POSAs understood that "provisioning" was a term generally used in many technical contexts, including databases, to refer to configuring and providing resources to an entity that needed them. EX1010, 2:47-50 (assignee Oracle, explaining in "Provisioning for Database Grid" section that "provisioning refers to providing and configuring the computational resources and data needed to provide a service."); EX1011, [0004]; Bhattacharjee, ¶32. POSAs understood that databases required resources like (1) processors and memory to run the above-mentioned database engine software, and (2) storage for the underlying data. Bhattacharjee, ¶33; Quernemoen, Abstract; Oracle (EX1012), 2-4; RDS, 100-102.

In the '818 patent, "[d]atabase provisioning services involve creating and/or associating databases with the database system for management and use." EX1001, 3:41-43; *id.*, 15:62-65 (Figure 3's "process," which can "implement the database provisioning service…[,] may be used to create a new database or register an existing database"); Bhattacharjee, ¶34. In other words, the patent's "provisioning system" can either create a new database and provision it with resources, or provision resources to an already-existing database. Accordingly, the patent describes two different "example [UIs]"—one for "creating" and one for "registering" a "database." EX1001, 22:30-32; Bhattacharjee, ¶¶35-36.

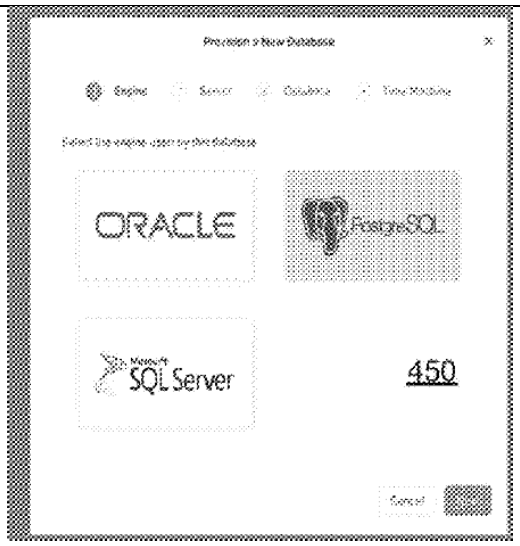
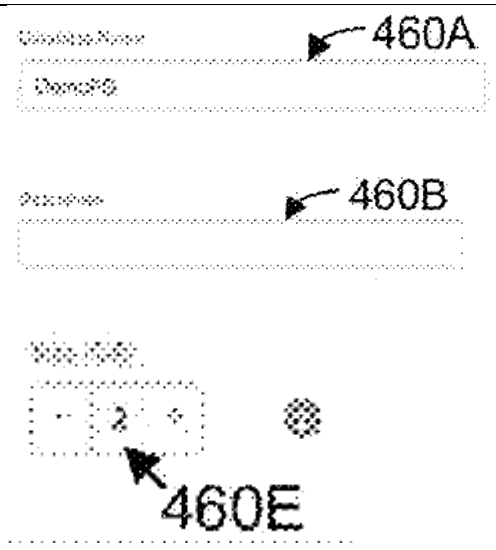
"[T]o initiate a database provisioning service," the user navigates to a "database page" having a "create button" and a "register button." EX1001, 23:50-

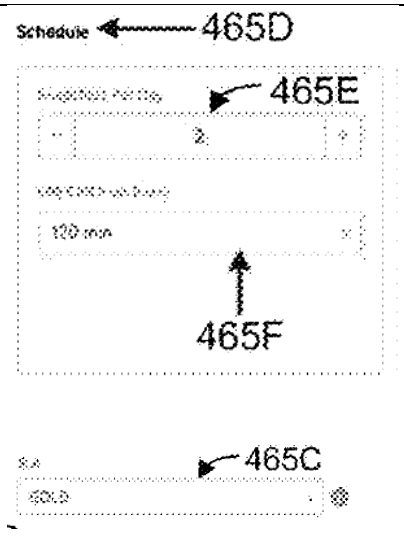
24:10, Fig. 4B (annotated below); Bhattacharjee, ¶37. The exemplary database-creation interface is shown in Figs. 4C-4G. EX1001, 22:32-33.



EX1001, Fig. 4B (annotated)

Once the user presses the "create" button, the user can enter various parameters on subsequent screens. EX1001, 24:11-16; Bhattacharjee, ¶38. Examples are shown below with excerpts from relevant figures.

selection of a “database engine type” from among those “supported by the database system,” such as “Oracle, PostgreSQL, etc.” (EX1001, 14:28-30, 24:21-29, FIG. 4C);	 FIG. 4C (excerpt)
“name 460A,” “description 460B,” “size 460E,” (EX1001, 24:61-25:2, Fig. 4E);	 FIG. 4E (excerpts)

"protection schedule 465D" and "SLA [service level agreement] level 465C" (EX1001, 25:2-14, Fig. 4F)	 FIG. 4F (excerpts)

Once the parameters are entered, the user presses a button to send the parameters to a "database creation system" to "start the database creation." EX1001, 25:23-28; Bhattacharjee, ¶39.

The above-referenced "protection schedule" may define a daily schedule for capturing "snapshots" from the database. EX1001, 25:14-23, 6:5-8; Bhattacharjee, ¶40. As was well-known, a snapshot is a "point in time image of the database." EX1001, 4:66-5:1; Simek (EX1014), [0002]; Ohno (EX1015), 1:54-61; Zha, 11:33-44; Bhattacharjee, ¶41. Snapshots are kept for a duration specified in the service level agreement ("SLA"). EX1001, 19:19-54, 20:65-21:2; Bhattacharjee, ¶42. Retaining snapshots for some user-defined duration was also well-known.

Bhasin (EX1016), [0009]; RDS, 5, 111-112; Tekade (EX1017), [0046], Fig. 6; Simek, [0034], Fig. 2; Bhattacharjee, ¶43.

As noted above, the '818 patent discusses a database "creating" interface and also an existing-database "registering" interface. EX1001, 22:30-32; Bhattacharjee, ¶44. The '818 patent's claims relate only to creating new databases; thus, the details of the patent's disclosure regarding provisioning existing databases are not relevant to this Petition.[2]

Finally, although the exemplary interfaces discussed above are graphical, the '818 patent notes that its "dashboard" "receive[s] user inputs" via either a "graphical user interface ('GUI')" or a "text-based [UI]." EX1001, 11:66-12:2, 12:33-50; Bhattacharjee, ¶45.

B. Challenged Claims

Claims 1-36 are reproduced *infra* §X, with labeled elements.

Independent claims 1 (computer-readable media), 8 (system), and 15 (method) each recite a processor receiving "inputs" of five parameters—database "name," "engine," "location," "level of a Service Level Agreement (SLA)," and a "protection schedule"—and then creating a database based on these inputs and

[2] Certain claims of related patents, e.g., U.S. Patent No. 11,010,336 claim 6, relate to existing databases.

based on an "amount of storage" determined for the database. The dependent claims recite additional limitations including (1) the database being located in either a "hyperconverged infrastructure system" or "cloud" (e.g., claims 2-3); (2) capturing snapshots based on the protection schedule (*e.g.*, claim 30); and (3) details regarding the SLA (*e.g.*, claim 29). Bhattacharjee, ¶51.

C. Prosecution History

The '818 patent's claims that were presented for examination (following voluntary preliminary amendments—EX1002, 203, 215) received a first-action allowance without any rejections. EX1002, 311-315. The Examiner's "[r]easons for allowance" discussed three references individually—EX1018, EX1019, and EX1020—and stated that none recited the specific combination of independent-claim limitations requiring receiving the protection schedule, presenting a UI to receive the various inputs, determining an amount of storage, and creating the database (*i.e.* [1E]-[1H], [8E]-[8H], [15E]-[15H]). EX1002, 313-314; *infra* §X. The Examiner then stated that "the prior art" did not teach that combination of limitations. Although Zha and a descendant of Sivasubramanian were of record, there is no evidence the Examiner considered the specific disclosures cited in the Grounds herein for the allegedly undisclosed limitations. Nor is there any evidence the examiner considered Sivasubramanian and Zha in combination. Bhattacharjee, ¶¶53-56.

D. Person of Ordinary Skill in the Art ("POSA")

The '818 patent relates to UIs for deploying database services. EX1001, Abstract, 1:34-2:19, 3:54-58; *supra* §IV.A; Bhattacharjee, ¶58. A POSA at the time of the claimed December 27, 2018 priority date would have had a bachelor's degree in computer science, computer engineering, or a related field, with two years of academic and/or industry experience with database technologies. A higher level of education may substitute for less experience. Bhattacharjee, ¶¶57-60.

V. CLAIM INTERPRETATION

Claim terms are construed herein using the standard used in civil actions under 35 U.S.C. §282(b), in accordance with the ordinary and customary meaning as understood by a POSA and the prosecution history. 37 C.F.R. §42.100(b).

VI. GROUND 1: CLAIMS 1-36 ARE UNPATENTABLE OVER SIVASUBRAMANIAN+ ZHA

A. Sivasubramanian (EX1005)

As discussed *supra* §IV.A, "provisioning" in the '818 patent encompasses either (1) creating and allocating resources to a new database; or (2) registering an existing database and providing resources. Bhattacharjee, ¶¶61-62; EX1001, 3:41-43, 15:62-65. As discussed below, POSAs understood that, although Sivasubramanian uses the terminology slightly differently and illustrates the functionality with fewer graphical figures than the '818 patent, Sivasubramanian

discloses UIs for the same two above-mentioned processes that fall within the scope of "provisioning" in the '818 patent. Bhattacharjee, ¶63.

1. Overview

Sivasubramanian is directed to "the ***creation***, ***provisioning***, and management of data stores and instances" using a "control environment." Sivasubramanian, Abstract; Bhattacharjee, ¶64. In Sivasubramanian, "the term 'data store' refers to… any device or combination of devices capable of storing, accessing, and retrieving data"; such a device or devices may include "databases." Sivasubramanian, 3:15-20.

Sivasubramanian's "control environment" (also called "control plane") can manage "workflows" that perform tasks in a "data plane" (also called a "data environment"). Sivasubramanian, Abstract, 2:21-24, Fig. 2. A "workflow" is a "sequence of tasks" for "specific job." Sivasubramanian, 6:25-26; Bhattacharjee, ¶¶65-66. "Workflows" may be initiated in response to "requests" submitted to the control plane using mechanisms including "application programming interfaces (APIs)." Sivasubramanian, 2:26-32, 4:59, 5:4-10, 5:35-44, 6:12-23; Bhattacharjee, ¶¶67-70. As explained *infra* §VI.A.1.a, requests can submitted using either graphical or textual interfaces. Bhattacharjee, ¶71.

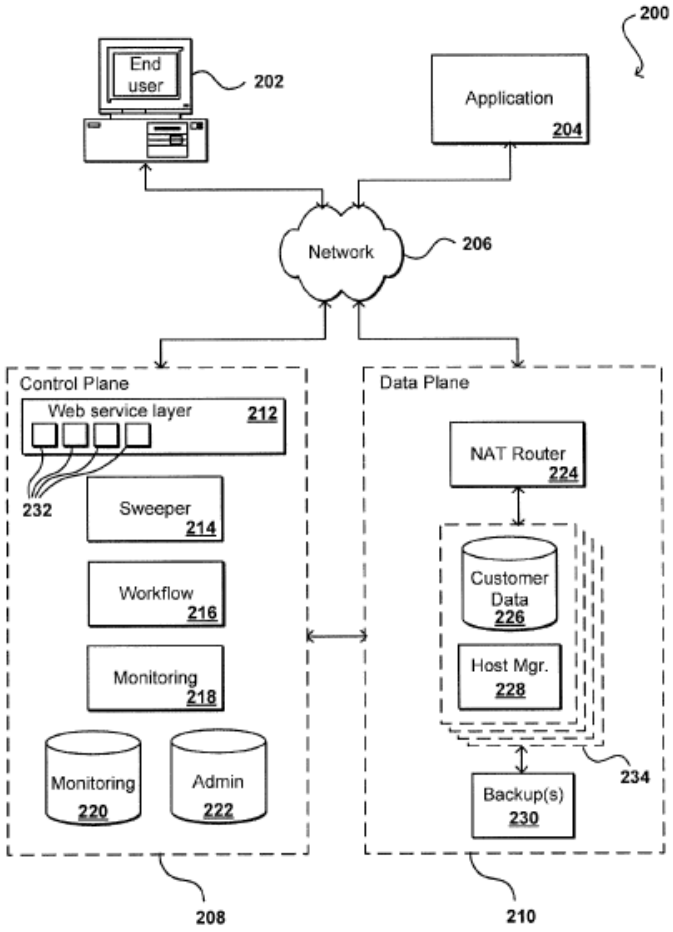


FIG. 2

Sivasubramanian, Fig. 2

One "workflow" the control plane can manage is "create database." Sivasubramanian, 6:59-65. As explained below, POSAs understood that Sivasubramanian's "create database" workflow includes what the '818 patent calls "provisioning" a database, for multiple reasons. Bhattacharjee, ¶¶72-73.

First, Sivasubramanian's "create database" workflow "includes tasks such as ***provisioning a data store instance***." 6:59-65.[3] POSAs understood that Sivasubramanian uses the term "data instance" or "data store instance" to refer to a database together with the resources—*e.g.*, processing power, memory, storage—that the database uses. *See*, *e.g.*, Sivasubramanian, 6:59-65 ("An example 'create ***database'*** workflow… includes tasks such as provisioning a ***data store instance***," "attaching" a "persistent storage volume to the ***data store instance***," and enabling the customer to "access or otherwise connect to the ***data instance***."), 11:48-51 ("customers can create a data store using a 'Create***Database***' or similar API. The user can call a Web service to specify any desired values for ***an instance type (which describes the CPU and memory capacity)***…"), 9:56-10:4; *see also* Sivasubramanian, 6:43-50 ("workflow template for creating a ***database***" is based on based on choice of "MySQL®… ***instance***… as opposed to an Oracle® RDMS or such other ***instance***"); Bhattacharjee, ¶74.

Second, Sivasubramanian references the "CreateDatabase" API "***provision[ing]***" a "***repository***." Sivasubramanian, 11:47-56 (describing "Create***Database***" API accepting "***repository*** name" input, and explaining that creation status can be checked using "Describe***Database***" API that "poll[s]… the status of the ***repository***" to determine "whether the ***repository*** state is

[3] Emphases added unless otherwise indicated.

provisioned"); Bhattacharjee, ¶75. POSAs understood that the "repository" is a *database* based on the referenced API names "Create***Database***" and "Describe***Database***," and because Sivasubramanian describes a "database" as a type of "repository" and frequently uses those terms interchangeably. Sivasubramanian, 4:49, 5:15-20, 15:37-38 ("***repository*** such as a relational ***database***"), 1:47-56, 4:58-62, 11:52-56, 12:24-33, 13:18-21, 12:64-13:5, 13:27-33; Bhattacharjee, ¶¶76-78. As noted above, in Sivasubramanian, workflows may be initiated in response to requests submitted via APIs; thus, POSAs understood from the name that the "Create***Database***" API causes initiation of the "create database" workflow. Sivasubramanian, 2:26-32, 5:35-44, 6:12-23; Bhattacharjee, ¶79.

Third, the '818 patent uses the term "provisioning" to encompass ***creating*** a database and giving it resources (as discussed *supra* §VI.A). As noted above, Sivasubramanian refers to the "create database" workflow "provisioning" a "repository," *i.e.* database, and also "provisioning" a "data instance" representing a database and its resources. Bhattacharjee, ¶80. POSAs understood from the name "***create*** database" that the workflow "***creates***" a database and then gives it resources, *e.g.* the resources in the instance; thus, Sivasubramanian's "create database" workflow does the same thing the '818 patent describes as "provisioning." Bhattacharjee, ¶80.

Fourth, the patent says "[d]atabase provisioning services involve creating and/or associating databases with the database system ***for management and use.***" EX1001, 3:41-43. Likewise, POSAs understood that following completion of Sivasubramanian's "create database" workflow that includes "provisioning," the database, and its "data instance" (as discussed above), can be ***managed*** by the control plane, and also can be ***used***. *See, e.g.*, Sivasubramanian, 6:59-67 ("'create database' workflow… might include[]" providing resources that "the customer can use to access… the data instance"), 9:4-8 (similar), 2:26-44 (similar), 9:45-65 (similar), 12:1-36 (describing API for modifying database), 13:18-15:31 (describing management actions available after creation), 19:32-64 (discussing workflow for backup); Bhattacharjee, ¶81.

Additionally, as noted *supra* §IV.A, the '818 patent teaches that "registering" an existing database and providing resources to it is also a process that qualifies as "provisioning." EX1001, 3:41-43. The patent does not expressly define "registering," but it describes an exemplary "registering" interface where a user identifies an existing database and then enters parameters for providing resources to that database. EX1001, 25:49-26:6, Fig. 5A-5E; Bhattacharjee, ¶82. As discussed *infra* §VI.A.1.a.i, Sivasubramanian likewise discloses a GUI that lets a user identify an existing database and specify parameters for configuring the database and providing resources to it.

a. UIs for "Create Database" Workflow

As noted above, Sivasubramanian teaches that users submit requests to the control plane to initiate workflows. As explained below, POSAs understood that Sivasubramanian describes two alternative interfaces to enable users to submit workflow-initiation requests to the control plane. Bhattacharjee, ¶83.

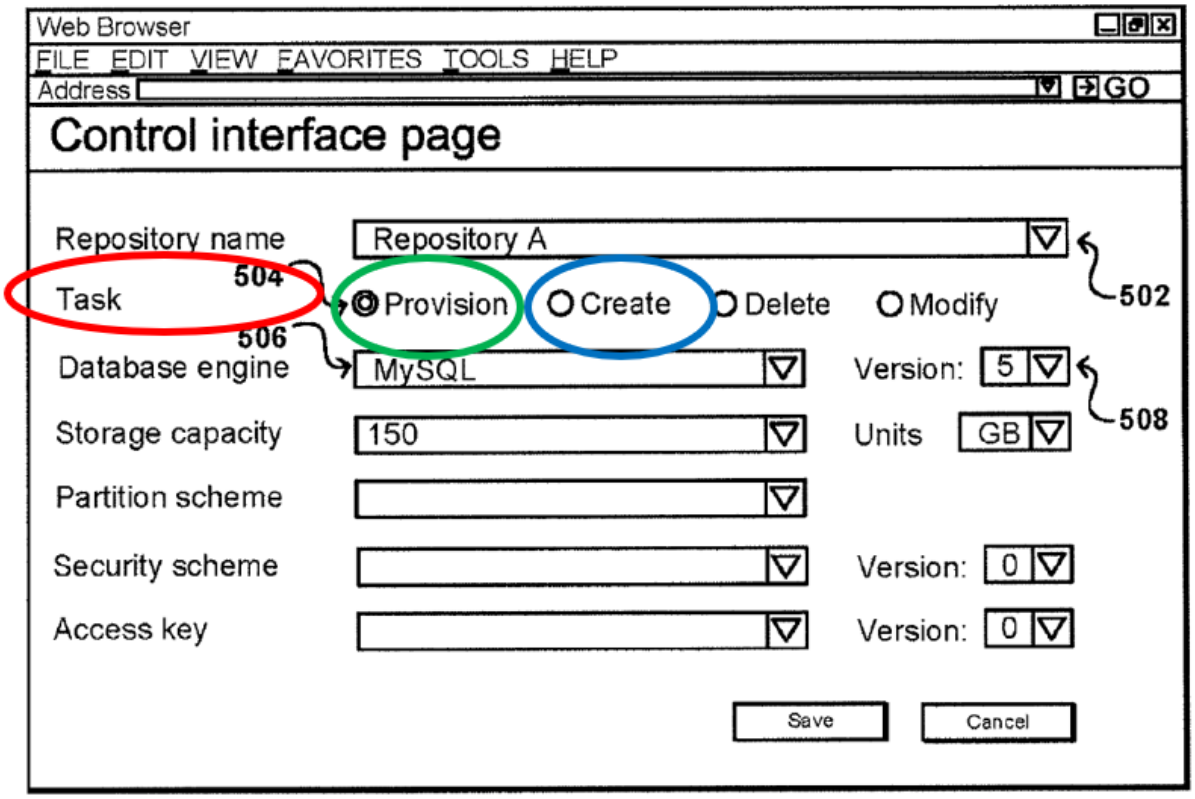
First, Sivasubramanian says "F[igure] 5 illustrates an example of a display… rendered in a browser application… that can allow a user to make calls into the control plane." Sivasubramanian, 12:53-55. This "display" is an example of an "interface page" "allow[ing] a user to submit requests to the control plane." Sivasubramanian, 12:49-52. Sivasubramanian describes the "display" in Figure 5 and the below-referenced command-line tool as two alternative ways to submit "calls to the control plane." Sivasubramanian, 12:53-13:5; thus, POSAs understood Sivasubramanian to disclose using a GUI like Figure 5's to submit requests to invoke the "create database" workflow. *Infra* §VI.A.1.a.i; Bhattacharjee, ¶84.

Second, Sivasubramanian states that users can "call a Web service" to invoke the "'CreateDatabase' or similar API." Sivasubramanian, 11:47-52; Bhattacharjee, ¶85. Sivasubramanian discloses a "command line tool" that is an "example" of a user "submit[ting] Web services calls to the control plane"; thus, POSAs understood that the command-line tool could be used to submit requests

via APIs to invoke the "create database" workflow. Sivasubramanian, 12:62-66; *infra* §VI.A.1.a.ii; Bhattacharjee, ¶86.

i. GUI

Sivasubramanian's Figure 5 "display" lets the user "select an action to be performed 504" and also select "options for the actions," such "options" including "database engine" (506) and "version information" (508). Sivasubramanian, 12:53-62; Bhattacharjee, ¶87.



Sivasubramanian, Fig. 5 (annotated)

In Figure 5 (annotated above), the "action to be performed 504" (Sivasubramanian, 12:59-60), labeled "Task," shows the "Provision" option

selected in this example. POSAs understood that the "Provision" "action"/"Task" performed using the Figure 5 interface provisions a *database*, and that the "Repository name" is a database name, because the interface includes an option for selecting a "***Database*** engine," and because of Sivasubramanian's repeated use of "repository" for "database" (noted *supra* §VI.A.1). Sivasubramanian, 12:56-62, Fig. 5; Bhattacharjee, ¶88. Furthermore, the figure shows that the "Repository [*i.e.*, database] name" field lets the user select from options via a drop-down menu (indicated by the down arrow); thus, POSAs understood that the "Provision" "action"/"Task" provisions an already-***existing*** database the user selects, rather than creating a new database. Bhattacharjee, ¶¶89-90; *see also* Sivasubramanian, 12:38-40 (discussing user choosing between "provision[ing] a new data instance" and "managing an existing MySQL database").

As shown above, Figure 5 also includes a "Create" option. POSAs understood that the "Create" option presents an interface for creating a *database*, at least because (1) Sivasubramanian describes the GUI and the "rds-create-***database***" command-line command as alternative methods for creating a database, and (2) Sivasubramanian describes "***database*** engine" as an "option" for the "action 504," labeled "Task 504" in Figure 5 (annotated above). Sivasubramanian, 12:49-13:5; Bhattacharjee, ¶91.

POSAs understood that selecting the "Create" option would cause the Figure 5 GUI to display fields for parameters needed for database creation, at least because Sivasubramanian states that the Figure 5 interface "include[s] options that allow the user to enter information needed to perform a control action on the data plane." Sivasubramanian, 12:55-57; Bhattacharjee, ¶92. Furthermore, POSAs would have understood Sivasubramanian to teach, or at a minimum render obvious, that the fields for database-creation parameters presented in the "Create" interface would include fields for the parameters shown in Figure 5 for the "Provision" interface, as explained below.

First, the create-database workflow includes provisioning the database by allocating it resources. *Supra* §VI.D.1.a. Thus, POSAs understood that the "Create" interface likewise included the fields on the "Provision" interface. Bhattacharjee, ¶¶93-94.

Second, as detailed *infra* VI.A.1.a.ii, Sivasubramanian discloses a command-line command, "rds-create-database," for database creation. Sivasubramanian, 13:1-5. POSAs understood that the parameters Sivasubramanian shows being accepted by "rds-create-database" include parameters shown in the "Provision" task screen in Figure 5, including name, engine, and size. Sivasubramanian, 13:1-5 (showing "rds-create-database" command accepts parameters including "dbname," "size," "engine"), Fig. 5 (showing fields for

"Repository name," Database engine," and "Storage capacity"); Bhattacharjee, ¶¶95 (explaining POSAs understood "db" in "dbname" as a common abbreviation for "database"), 203 (explaining POSAs understood "capacity" in Figure 5 as equivalent to "size" in "rds-create-database" command) (citing Sivasubramanian, 12:42-44).

POSAs understood that one difference between the "Provision" and "Create" interfaces is that in the "Create" interface, the user would enter a new "Repository name," *i.e.* database name (*see supra* § VI.D.1.a), for the new database to be created, rather than selecting the name of a previously-created database to provision (as shown in Figure 5 for the "Provision" task). Bhattacharjee, ¶96.

To the extent not taught by Sivasubramanian, POSAs understood that having the "Create" task GUI accept inputs of parameters shown on the "Provision" task GUI would have been a well-known UI implementation that beneficially increased user convenience by letting the "Create" UI accept parameters needed to provision a created database (functionality Sivasubramanian itself teaches, *see supra* §VI.A.1). Bhattacharjee, ¶97; RDS, 13-18 (describing "Creating a MySQL DB Instance" UI that creates a "provisioned" database). POSAs would have reasonably expected success because such a UI implementation was within a POSA's skill. *Infra* §VI.A.2.

As discussed *supra* §VI.A.1.a, the Figure 5 GUI is used to submit requests to the control plane to initiate workflow. Thus, POSAs understood that using the "Create" option in the Figure 5 GUI invokes the "create database" workflow, which *provisions* a new database (under the '818 patent's meaning) by creating it and allocating it resources as discussed *supra* §VI.A.1. Bhattacharjee, ¶98.

ii. Command-Line

The "command[-]line tool," instead of graphically presenting fields for the user to provide parameters, presents a text-based interface (command line) for the user to enter text in a specified command format to input parameters for tasks such as database creation; the command-line format accepts input parameters as a list of pairs of parameter names and values, as shown in the example "rds-create-database" command:

> rds-create-database --identifier customerprod --dbname customer --size 80 [-]-class small --engine mysq5.1 --master master_username --password master_password -port 4030

Sivasubramanian, 12:64-13:5; Bhattacharjee, ¶99.

POSAs understood that the "rds-***create-database***" command creates a database (1) because of the command's name; (2) because Sivasubramanian describes the command as an example command for "generat[ing] a request to ***create*** a new data ***repository***," a term which in Sivasubramanian encompasses a

"database" as discussed *supra* §VI.A.1; and (3) the command accepts "mySQL" as an "engine" value. Sivasubramanian, 12:64-13:5; Bhattacharjee, ¶100.

As discussed *supra* §VI.A.1.a, the command-line tool is used to submit requests to the control plane to initiate workflows. Thus, POSAs understood that the "rds-***create-database***" command invokes the "create database" workflow, which *provisions* a new database (under the '818 patent's meaning) by creating it and allocating it resources as discussed *supra* §VI.A.1. Bhattacharjee, ¶101.

b. Backup "Workflow"

Sivasubramanian discloses a "recovery and backup" "workflow[]." Sivasubramanian, 6:31-42; Bhattacharjee, ¶102. Sivasubramanian says the control plane can use "software frameworks" for "database backup and recovery," and also says a "framework" can handle "backing up both the data files and any associated log files." Sivasubramanian, 16:21-27, 19:25-28. Sivasubramanian describes one exemplary "approach for backing up data" whereby a "su[s]pendDatabaseForBackup" command is used to "suspend[] data operations" while "s[n]apshots are taken of the appropriate data volumes." Sivasubramanian, 19:32-41; Bhattacharjee, ¶¶103-104. Additionally, in Sivasubramanian's "approach for backing up data," "log files" are "copied" while the database is suspended. Sivasubramanian, 19:32-41; Bhattacharjee, ¶105. The log files are used for "replay[ing]… ***transactions*** in case the data files have to be ***restored***";

thus, POSAs understood that Sivasubramanian's "log files" are what other art and the '818 patent refer to as "transaction logs" or "transactional logs." Sivasubramanian, 19:65-20:1; Zha, 10:6-11; RDS, 112; EX1001, 36:56-61; EX1060, 1:14-22, 1:61-63; EX1061, 4:17-30; Bhattacharjee, ¶105.

2. Motivation to Customize Sivasubramanian's UI

Although Sivasubramanian's Figure 5 shows a particular example set of configuration parameters that can be input, POSAs would have understood that the UI shown could beneficially be implemented to accept additional inputs not expressly shown in the Figure's example. Sivasubramanian notes that Figure 5 is one "***example*** of a display" in an "interface" including "options that allow the user to enter information needed to perform a control action on the data plane," and notes that "other such values" aside from those shown can be specified. Sivasubramanian,12:53-64; Bhattacharjee, ¶¶106-107. Sivasubramanian also teaches that its UIs are intended to be flexible and to enable the user to enter whatever options are appropriate to accomplish a desired workflow. Sivasubramanian, 11:49-52 ("user can call a Web service to specify ***any*** desired values for an instance type…, storage size, repository name, port, ***and other such values***."), 2:23-26, 11:41-47, 6:7-12, 6:45-52; Bhattacharjee, ¶108. Furthermore, Sivasubramanian's discussion of the "command line" UI for creating a database shows different parameters than those shown in Figure 5, which indicates to a

POSA that other parameters could similarly be added to Figure 5's GUI. *See* Sivasubramanian, 13:1-5 ("rds-create-database" command including parameters for "identifier," "class," "password"); Bhattacharjee, ¶109.

Such modifications would have been well within a POSA's skill. Sivasubramanian describes Figure 5 as an example page rendered in a "browser application." Sivasubramanian, 12:52-55. Sivasubramanian also discloses implementing its functionality on "[w]eb server[s]" running "a variety of server or mid-tier applications, including HTTP servers," "CGI servers," and "Java servers" implemented in "any programming language." Sivasubramanian, 23:4-16. POSAs had the skill to use a "programming language" to implement a "web application" to display a modified version of Figure 5's exemplary UI that includes additional configuration options. Bhattacharjee, ¶110.

For the same reasons, POSAs would also have been motivated to modify Sivasubramanian's command-line tool to accept additional inputs not expressly shown in the example at 13:1-5 of Sivasubramanian. It would have been within a POSA's skill to use a "programming language" to implement such a modified command line. Sivasubramanian, 23:4-16; Bhattacharjee, ¶111.

B. Zha (EX1006)

Zha teaches techniques for creating a "virtual database"—*i.e.*, a copy of an existing database from a particular point in time—that is used for purposes

including "backup." Zha, Abstract; 6:49-58, 24:24-27; Bhattacharjee, ¶¶112-113. Zha's "[m]ethods of creating a virtual database (VDB) read different point-in-time copies of a source database," which Zha also refers to as a "production database." Zha, 1:66-2:6; Bhattacharjee, ¶114. Zha also teaches retrieving "transaction logs" from the source/production database; these logs can be combined with point-in-time copies to "reconstruct a copy of a database… corresponding to times in the past in between the times [at] which point-in-time copies are made." Zha, 9:63-10:11, 15:58-62, 18:1-3; Bhattacharjee, ¶115.

Zha's above-mentioned "database storage system" (where database copies are stored) can be configured by a "database administrator" using a "system configuration manager." Zha, 11:26-28, 6:11-58; Bhattacharjee, ¶¶116-117. This "manager" lets a user "setup… the ***schedule***" specifying the "frequency and times during the day" for "retriev[ing] ***PIT [point-in-time] copies of databases***" and "retriev[ing] updates to online ***transaction logs***." Zha, 11:33-43; Bhattacharjee, ¶117.

POSAs understood that "point-in-time copies" as discussed in Zha are "snapshots" of the source database at the time the copy was made, for multiple reasons. As noted *supra* §IV.A, POSAs understood that the term "snapshot" referred to a record of the database's state at a point in time. Bhattacharjee, ¶118; Simek, [0002]; Ohno, 1:54-61. Furthermore, Zha states that "transaction logs can

be applied to a VDB [virtual database] created based on a PIT copy so as to create a ***snapshot*** of the source database for a time point that occurs ***after*** the PIT copy was made." Zha, 17:63-66. Because applying transaction logs creates a "snapshot" of the database at a time ***after*** the PIT copy was made, POSAs understood that the PIT copy itself represents a "snapshot" ***at*** the time it was made, since Zha also explains that the transaction logs "can be used to reconstruct a ***copy*** of a database… corresponding to times in the past in ***between*** the times as which ***point-in-time copies*** are made." Zha, 10:6-11; Bhattacharjee, ¶119.

Zha also discloses "policies defined by a database administrator" for operations including "purging of information… including point-in-time copies of databases and transaction log information." Zha, 11:45-54. Such "policies" include a "schedule for purging the point-in-time copy after a retention period" (Zha, clam 11, 35:18-20) and "a schedule for purging the transaction logs after a retention period" (Zha, claim 13, 35:24-26). Bhattacharjee, ¶120.

C. Sivasubramanian+Zha Combination

POSAs seeking to implement Sivasubramanian's "backup" workflow discussed *supra* §VI.A.1.b would have considered references like Zha as to certain implementation details on which Sivasubramanian is silent/agnostic. As explained below, POSAs also understood that, although Sivasubramanian and Zha use slightly different terminology, both discuss database backup using snapshots and

transaction logs, with Zha's teachings supplying further details. POSAs would have found it obvious to implement Sivasubramanian using Zha's teachings in multiple independent ways, as discussed below. Bhattacharjee, ¶121.

Sivasubramanian's discussion of "snapshots" appears in a paragraph describing "[an] approach for backing up data" that immediately follows a paragraph introducing Sivasubramanian's discussion of "***repository*** and data backup" and that also notes that "***approaches*** can differ from various ***database engines***." Sivasubramanian, 19:20-35. Thus, POSAs understood from Sivasubramanian's language and structure that Sivasubramanian discloses ***database*** snapshots. Bhattacharjee, ¶122; *see also* Sivasubramanian, 15:1-15 (earlier discussion of "tak[ing] a snapshot of a production instance" so that a "test instance" created with a "create-***database***" command can be "fully populated and comparable with the production [data]").[4]

Additionally, Sivasubramanian's exemplary "approach" uses a "su[s]pend***Database***ForBackup" command to "suspend[] data operations" while "s[n]apshots are taken of the appropriate data volumes." Sivasubramanian, 19:32-41; Bhattacharjee, ¶123. Sivasubramanian also discloses that a database's data is

[4] Sivasubramanian 15:3 says, "production date," but POSAs understood this is a typo. Bhattacharjee, ¶122.

stored on "volume[s]" allocated to a database's storage "instance." Sivasubramanian, 6:59-63 ("'create ***database'*** workflow… might include… provisioning a data store ***instance***… [and] attaching [a] persistent storage ***volume*** to the data store instance"), 8:22-25, 15:56-60. Moreover, POSAs understood that taking snapshots of underlying data volumes was a typical and customary way to take a database snapshot. Bhattacharjee, ¶124; EX1022, 1; Ohno (EX1015), 1:28-29, 1:54-2:29; EX1055, 17:23-40, Abstract. Thus, POSAs understood Sivasubramanian to disclose taking ***database*** snapshots by taking snapshots of the underlying data volumes. Bhattacharjee, ¶125.

POSAs understood that "tak[ing]" "snapshots" of a database in Sivasubramanian (Sivasubramanian, 15:1-15, 19:20-64) is analogous to what Zha refers to as "retriev[ing] PIT [point-in-time] copies" (Zha, 11:33-43, 17:63-66), and likewise understood that "backing up" the "log files" in Sivasubramanian (Sivasubramanian, 19:65-20:1) is analogous to what Zha refers to as "retriev[ing] updates to online transaction logs" (Zha, 11:33-43); both references' processes result in the snapshot/log being stored in some storage location used for storing snapshots/logs. *Compare* Sivasubramanian, 19:32-35 *with* Zha, 2:19-24, 9:67-10:3; Bhattacharjee, ¶126.

Sivasubramanian does not restrict implementation details such as how often to take the snapshots and back up the transaction logs each day, what time of day

these events occur, or how long to retain the snapshots and logs. Sivasubramanian, 19:20-20:11. POSAs understood that it was typical to collect snapshots and other database-backup information at user-configurable frequencies and times and store such information for user-configurable retention periods. Bhattacharjee, ¶127; Bhasin (EX1016), [0009]; RDS, 16, 111-112; Tekade (EX1017), [0046], Fig. 6; Simek, [0034], Fig. 2. Furthermore, POSAs understood that storing snapshots and transaction logs required storage space, and that taking snapshots and logs required the use of computing resources and potentially impacted the database's performance; thus, POSAs understood that allowing users of Sivasubramanian's control plane to specify snapshot and log frequencies, times, and retention periods would beneficially enable users to tune those parameters to suit their own needs. Bhattacharjee, ¶128; Madany, [0023] (explaining how available storage limits constrain snapshot storage and noting that schedules and retention periods can mitigate constraints); RDS, 5 ("Increasing your backup retention period or taking additional database snapshots increases the backup storage consumed by your database."), Sivasubramanian, 19:31-41 (exemplary backup scheme requires suspension of database); Ohno, 1:54-61 (prior art snapshotting commonly required pausing database).

Therefore, POSAs seeking to implement Sivasubramanian's exemplary "approach for backing up data" (Sivasubramanian, 19:32) would have been

motivated to use an administrator-defined "schedule" including a "frequency" and "times during the day" for taking snapshots and backing up transactional/transaction logs, and to store them for administrator-defined "retention periods," as disclosed in Zha. Zha, 11:33-54, 35:18-20, 35:24-26; Sivasubramanian, 19:20-20:11; Bhattacharjee, ¶129.

In the resulting "Sivasubramanian+Zha" combination, Sivasubramanian's "create database" workflow UIs (*supra* §VI.A.1.a) would incorporate UI elements to accept user-defined snapshot and transaction-log collection frequencies, times during the day, and/or retention periods, as detailed *infra* §§VI.C.1-VI.C.2. Additionally, in Sivasubramanian+Zha, the "create database" workflow would setup the created database to have snapshots and logs (1) collected at the respective user-defined frequencies and times and (2) stored for the respective user-defined periods. Sivasubramanian, 19:32-39; Bhattacharjee, ¶130.

Alternatively, POSAs would have found it obvious to employ Zha's technique of creating VDBs (alternatively or additionally to creating Sivasubramanian's snapshots) to beneficially achieve the "efficient[] use [of] storage and computing resources" that Zha's technique provides. Zha, 1:63-65; POSAs would have been motivated to customize Sivasubramanian's database creation UIs to let the user set the snapshot and log schedules and retention periods for the same reasons discussed above, and the resulting system would incorporate

the UI elements detailed *infra* §§VI.C.1 -VI.C.2 into Sivasubramanian's database-creation UIs. The resulting system would also implement Sivasubramanian's "create database" workflow to set up the database to have "point-in-time copies," *i.e.* snapshots, and "transaction logs" "retriev[ed]" from it, as taught in Zha, based on the user-entered parameters as discussed above. Zha, 11:33-43; *supra* §VI.B; Bhattacharjee, ¶131.

POSAs would reasonably have expected success with either alternative implementation of the Sivasubramanian+Zha combination discussed above. As detailed *infra* §§VI.C.1-VI.C.2, it was within a POSA's skill to customize Sivasubramanian's UIs to accept the inputs taught by Zha. Bhattacharjee, ¶¶132-133. It would also have been within a POSA's skill to have the "create database" workflow setup the database to have snapshots and logs taken/retrieved and retained based on user-entered parameters, at least because Sivasubramanian itself discloses: (1) the control plane executing tasks based on user-entered parameters; and (2) snapshot and log collection. *See, e.g.*, Sivasubramanian, 11:41-52, 12:64-13:5, 19:32-39; Bhattacharjee, ¶134.

Moreover, regarding the second variant, Sivasubramanian says the "approach" it describes is "***[a]n***" approach, and a POSA would have understood that Sivasubramanian's control plane could accommodate other approaches like Zha's. Sivasubramanian, 19:32; Bhattacharjee, ¶135. It would have been within a

POSA's skill to implement Sivasubramanian's control plane to use Zha's "virtual database" technique; Zha discloses that its teachings can be implemented in software, and Sivasubramanian likewise discloses implementing the control plane using software "framework" that can perform tasks including "backup and recovery." Bhattacharjee, ¶136; Zha, 32:49-52; Sivasubramanian, 16:21-28.

1. UI for retention period

Zha does not mandate a specific interface for specifying policies with snapshot or transaction log retention periods. However, one well-known UI technique for specifying retention periods was to let the user choose a "service level" specifying the retention period; such levels were often specified by choosing a predefined "service level agreement" ("SLA"). Bhattacharjee, ¶137; Tekade, [0046], Fig. 6 (depicting "[UI] for configuring a single SLA protecting a database" including a "snapshot policy" dictating that a "backup image is retained for [a specified] number of day[s]"); EX1024, [0120]; EX1025, [0004]; EX1026, 11:39-43; EX1027, 19:32-35. It was also well-known background knowledge to define multiple SLA "levels" with names like "gold," "silver," etc. Bhattacharjee, ¶138; Gomatam (EX1028), [0046], [0056]-[0058] (disclosing allowing a customer to specify "***a Service Level Agreement (SLA)***" from "different customer levels includ[ing] ***a basic level, a silver level, and a gold level***" for "services" such as "database services"); Higginson (EX1029), 9:4-7; EX1030, 3:62-4:6.

In Sivasubramanian+Zha, POSAs would have been motivated to have control plane users specify Zha's configurable snapshot and transaction log retention periods (*supra* §VI.B) by specifying an SLA level, for multiple reasons. First, POSAs understood that it was beneficial to give users familiar UIs, and that users were familiar with SLAs to specify retention periods, as discussed above (and also familiar generally with specifying time intervals via GUIs). Bhattacharjee, ¶¶139-140, Tekade, Fig. 6; RDS, 15-16; Simek, Fig. 2. Second, POSAs understood that letting users from predefined sets of parameter values was a well-known and increased user convenience. Bhattacharjee, ¶141; Higginson, 9:49-10:5; EX1031, 1:16-29. Thus, in either variant of Sivasubramanian+Zha discussed above, POSAs would been motivated to implement the database-creation UIs to let the user specify transaction log and snapshot retention periods via the well-known UI technique of selecting an SLA level from multiple levels, in order to beneficially increase user familiarity and convenience. Bhattacharjee, ¶139-142.

Furthermore, POSAs would have understood that accepting an input of the SLA as part of a database creation process would beneficially allowed the database to be protected as soon as it is created and provisioned, since parameters for snapshot and log capturing and retention would already have been specified. Bhattacharjee, ¶143. Moreover, it was well-known in database-creation and provisioning UIs both to (1) accept inputs related to retention of snapshots or other

backup data; and (2) accept an SLA-level selection. Bhattacharjee, ¶¶144-146; Gomatam, [0046], [0056]-[0058] (SLA specified as part of an "order" for services including "database services"); Higginson, 9:49-10:12, Fig. 4; RDS, 16-17 (database creation options include "backup retention period"); Oracle, 7 (database creation options include "Backup Destination"); Google, 2-3.

In Sivasubramanian's GUI, POSAs would have been motivated to implement letting the user select a level from a list of levels via a selectable interface like a drop-down menu, since selectable interfaces in GUIs were well-known, including in Sivasubramanian. Bhattacharjee, ¶147; Sivasubramanian, 15:59-60, Fig. 5 (*e.g.*, "Database engine" selection); Tekade, Fig. 6; RDS, 15. And POSAs would have been motivated to implement the "rds-create-database" command-line command to accept another parameter whose value represents a level, similar to the already-disclosed "engine" parameter whose value specifies an engine. Sivasubramanian, 13:1-5. POSAs would have reasonably expected success with such modifications, because SLAs were well-known (*see* above), and because customizing Sivasubramanian's UIs was within a POSA's skill (*supra* §VI.A.2). Bhattacharjee, ¶147.

Furthermore, POSAs would have found it obvious to have the SLA specify either one retention period for both snapshots and logs or separate periods for each. POSAs understood there were only two options—uniform or non-uniform—and

that either was an obvious design choice. POSAs would additionally have recognized the benefits of a uniform period for the reasons explained *infra* §VI.D.11 (claim 2). POSAs would reasonably have expected success with either approach as the UI modifications were within a POSA's skill. *Supra* §VI.A.2.

2. UI for schedule

As explained below, POSAs would have been motivated to customize Sivasubramanian's Figure 5 GUI to accept an input defining Zha's above-discussed "schedule" including a "frequency" and "times during the day" for taking snapshots and backing up transactional/transaction logs. Zha, 11:33-54; *supra* §VI.C. Including fields related to database protection in database-creation and/or provisioning UIs was well-known. Bhattacharjee, ¶¶148-149; RDS, 16-17, 187 (database creation options include a daily "backup window" for taking "snapshots"); Oracle, 7 (database creation options include "Backup Destination"); Google, 2-3. Additionally, POSAs understood that accepting an input of a protection schedule via a database-creation UI beneficially allowed the database to be protected as soon as it is created and provisioned. Bhattacharjee, ¶150. POSAs would have reasonably expected success with such modifications, because including fields related to database protection in database-creation UIs was well-known, as discussed above, and because customizing Sivasubramanian's UIs was within a POSA's skill (*supra* §VI.A.2). Bhattacharjee, ¶151.

POSAs would have found it obvious to add parameters for specifying the protection schedule to Sivasubramanian's "rds-create-database" command-line tool (Sivasubramanian, 13:1-5) for the same reasons. Bhattacharjee, ¶152.

D. Claim-by-Claim Analysis

1. Claim 1

a. [1Pre]: "A non-transitory computer-readable media comprising computer-readable instructions stored thereon for provisioning a database, wherein the computer-readable instructions when executed by a processor cause the processor to:"

Sivasubramanian's invention can be implemented with a "computing device[]" including a "***computer-readable storage media reader***" (23:39-47) that reads from "***computer readable media***" storing "information such as ***computer readable instructions***;" examples of such media include "non-volatile" media like "CD-ROM," and "magnetic disk," which POSAs understood are *non-transitory*. Sivasubramanian, 22:33-37, 23:39-47, 23:61-24:8; Bhattacharjee, ¶153. Sivasubramanian also says the "control plane" includes "computer-executable software" (*i.e.*, *instructions*), and Sivasubramanian includes claims reciting a "control environment" operating "under control of one or more computer systems configured with executable ***instructions***." Sivasubramanian, 5:35-39, 24:20-24, 25:4-8, 26:23-26; Bhattacharjee, ¶153. Thus, POSAs understood Sivasubramanian to disclose implementing its "control plane" via *a non-transitory computer-*

readable media comprising computer-readable instructions stored thereon. Bhattacharjee, ¶153.

Because the "control plane" controls "workflows," POSAs likewise understood that the control plane *instructions* include *instructions* for controlling workflows. Sivasubramanian, 6:7-38, Fig. 2; Bhattacharjee, ¶154.

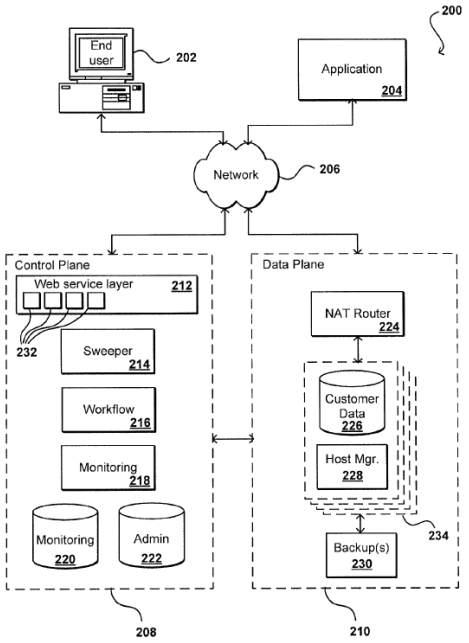


FIG. 2

Sivasubramanian, Fig. 2

Sivasubramanian's "create database" workflow includes what the '818 patent calls "***provisioning***" a database. *Supra* §VI.A.1. Thus, the *instructions* for implementing Sivasubramanian's "create database" workflow, including the instructions for presenting the user interfaces discussed *supra* §VI.A.1.a that

receive requests to initiate the workflow, are *instructions for provisioning a database*. Bhattacharjee, ¶155.

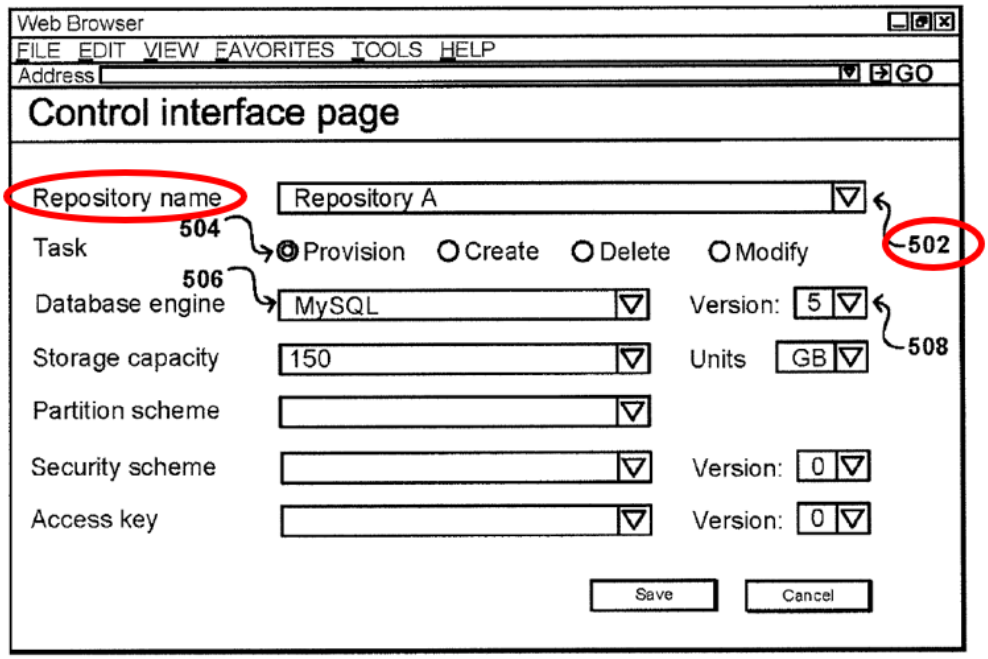
Furthermore, POSAs understood that Sivasubramanian's *instructions for provisioning a database* are *executed by a processor* and *cause the processor to perform* actions meeting the remaining limitations of claim 1. As discussed above, the *instructions for provisioning* are part of Sivasubramanian's above-discussed "computer-readable instructions" for the control plane; POSAs understood these instructions are intended to be executed by a processor to cause the processor to perform the functionality the instructions specify. Bhattacharjee, ¶156; Sivasubramanian, 27:18-35 (claim 20 reciting "system for controlling a data environment using a control environment" comprising "***processor***" and "instructions that, when executed by" the "processor" "***cause*** the system" to perform actions), 5:35-39 (control plane includes "computer-***executable*** software"). Furthermore, as discussed above, the *instructions for provisioning* are the instructions for the "create database" workflow, including instructions for presenting the workflow's UIs. As detailed *infra* §§VI.D.1.b-VI.D.1.i, Sivasubramanian+Zha's control plane performs the actions cited for Elements [1A]-[1H] as part of either the GUI or command-line UI for using the "create database" workflow to create and provision a *database—the database* of the

remaining elements. As shown below, both options independently meet claim 1. Bhattacharjee, ¶157.

b. [1A]: "receive a first input of a database name for the database;"

i. GUI

Sivasubramanian's Fig. 5 (annotated below) shows a field for a "Repository ***name***" (502).



POSAs understood that the "Repository name" in Figure 5 is a *database* name, as explained *supra* §VI.A.1.a.i. Thus, POSAs understood that the "repository name" in Sivasubramanian's Fig. 5 GUI is a *database name for the database*. Bhattacharjee, ¶¶158-159.

POSAs understood that the Figure 5 GUI would display a field for entering a new "repository name," *i.e. database* name, when the "Create" task is selected. *Supra* §VI.A.1.a.i. POSAs understood that the purpose of presenting a UI with a field for a value is to *receive an input* of that value; thus, POSAs understood that Sivasubramanian's GUI *receives a first input* of the *database name for the database* via the GUI. Sivasubramanian, 5:59-62 (referencing "[w]eb service frontend features such as … validating user ***input*** …"), 12:55-57 (Fig. 5 UI lets user "enter information"); 11:49 (discussing control plane user "specify[ing]" a "name" (11:49-55). Bhattacharjee, ¶160.

ii. Command Line

Sivasubramanian's "rds-create-database" command accepts the parameter "dbname," which POSAs understood is a *database name for the database*. Sivasubramanian, 12:64-13:5; Bhattacharjee, ¶161 (explaining POSAs understood "db" in "dbname" to be a common abbreviation for "database"). POSAs also understood that the purpose of providing a command-line command with an option for specifying a parameter value is to *receive an input* of that value; thus, Sivasubramanian's "rds-create-database" command *receives* the claimed *first input*. Bhattacharjee, ¶¶162-163; Sivasubramanian, 5:59-62 (referencing "[w]eb service frontend features such as… validating user ***input***…"), 12:62-67 (explaining

that "command line tool" lets user "create and submit Web service calls to the control plane").

iii. Order of Inputs Irrelevant

The recitation in Elements [1A]-[1E] of "*first*" through "*fifth*" inputs does not impose any chronological ordering (or any other ordering type) on reception of the inputs. Interpreting the '818 patent's claims to require a particular order of inputs would be inconsistent with the specification. For example, [1A] recites that the "*first*" input is "*database name*," while [1B] recites that the "*second*" input is a "*database engine*;" however, the specification discusses receiving those two inputs in the opposite order. In the "example user interfaces" shown in the '818 patent's Figures 4A-4G (EX1001, 22:33-34), database engine selection occurs in Figure 4C (EX1001, 24:11-14), while database name input occurs later in the sequence, in Figure 4E. Bhattacharjee, ¶¶164-165; EX1001, 24:26-29 (user clicks "Next" to go from 4C to 4D), 24:52-56 (using clicks "Next" to go from 4D to 4E). Furthermore, Figure 4F shows a single screen for selecting both an SLA level ([1D]'s "*fourth*" input) and a protection schedule ([1E]'s "*fifth*" input), confirming that no particular order for receiving the inputs is required. EX1001, 25:6-28.

Additionally, [1F] recites "*present[ing]* ***at least one*** *user interface to receive*" the five inputs, which allows all five inputs to be received via "*one*" interface. Bhattacharjee, ¶¶166-167; EX1058-EX1059.

c. [1B]: "receive a second input of a database engine for the database;"

i. GUI

Sivasubramanian's Fig. 5 (annotated below) shows selection of a "***Database engine***" (element 506).

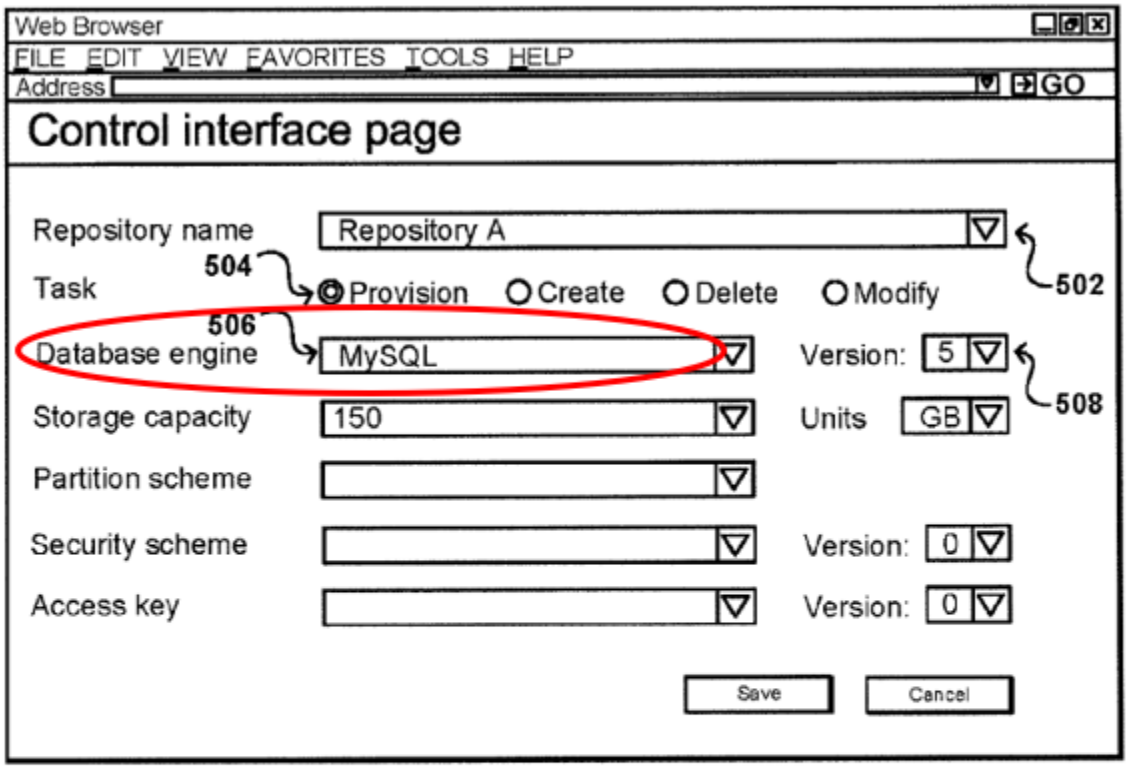


FIG. 5

POSAs understood that the Figure 5 GUI would display a field for selecting a "database engine" when the "Create" task is selected (*supra* §VI.A.1.a.i).

POSAs also understood that the purpose of presenting a UI with a field for a value is to *receive an input* of that value (*supra* §VI.D.1.b.i), and Sivasubramanian discusses the control plane performing workflows or actions for a specified "engine"; thus, POSAs understood that Sivasubramanian's GUI *receives a second input of a database engine for the database*. Bhattacharjee, ¶¶168-170; Sivasubramanian, 6:39-41, 10:37-40, 12:59-61, 19:25-31.

ii. Command Line

Sivasubramanian's "rds-create-database" command includes the parameter "engine," which POSAs understood is a *database engine for the database*. Sivasubramanian, 12:64-13:5. POSAs also understood that the purpose of providing a command-line tool with an option for specifying a parameter value is to *receive an input* of that value. *Supra* §VI.D.1.b.ii. Thus, Sivasubramanian's "rds-create-database" includes *receives* the claimed *second input*. Bhattacharjee, ¶171.

d. [1C]: "receive a third input indicative of a location where the database is to be provisioned"

i. GUI

The only reference to database "location" in the '818 patent's specification related to database provisioning is a reference to "the ***network location*** of the database being created." EX1001, 17:38-40; Bhattacharjee, ¶172.

Sivasubramanian's "example 'create database' workflow" includes "allocating and attaching a DNS address or other address, port, interface, or identifier," and "a user is provided with the DNS address and a port address… to access" the "[data] instance," *i.e.* the instance running the database (*see supra* §VI.A.1). Sivasubramanian, 6:59-67. POSAs understood that control plane users can specify the "DNS address" and "port" because Sivasubramanian says control plane users can "specify" a "port and other such values," including via the GUI. Sivasubramanian, 11:40-52; Bhattacharjee, ¶174-175. Furthermore, Sivasubramanian's "rds-create-database" invocation example shows a "port" parameter, and it would have been obvious to accept the same parameter via a GUI. Sivasubramanian, 13:1-5; *supra* §VI.A.2. Thus, POSAs understood that obvious implementations of Sivasubramanian's control plane can *receive* "DNS address" and "port" *inputs* from the "Create" GUI. Bhattacharjee, ¶176. Implementing such functionality would beneficially enable Sivasubramanian's above-discussed disclosure of creating a database accessible via DNS address and port, and collecting the parameters via GUI was within a POSA's skill (*supra* §VI.A.2); such a GUI would also beneficially match the command line, discussed *infra* §VI.D.1.d.ii. Bhattacharjee, ¶176.

POSAs understood that a "DNS address" is *indicative of a* [network] *location*. Bhattacharjee, ¶177; Sivasubramanian, 2:40-44 (referencing "the ***DNS***

(domain name system) address or other location information" for a database's "data instance"); EX1033, [0003]. POSAs also understood that the "port" is *indicative of a* [network] *location*, because an external entity seeking to "access" the database "use[s]" a "port address." Sivasubramanian, 6:65-67; Bhattacharjee, ¶177. The '818 patent similarly describes "a listener port 460G" as an input in an example "user interface 460" used to specify "various parameters for creating a database," but does not describe anything else indicative of location as being input in the "example user interfaces for creating a database" in Figures 4A-4G. EX1001, 22:30-42, 24:57-67, FIG. 4E; Bhattacharjee, ¶177.

Finally, the port and DNS address *location* is *where the database is to be provisioned* because the database is created at that location (as discussed *infra* §VI.D.1.i ([1H])), and *provisioning* is part of database creation (*supra* §VI.A.1). Bhattacharjee, ¶177. Therefore, each of the above-mentioned "DNS address" and "port" *inputs* that is *received* is independently a *third input indicative of a location where the database is to be provisioned*.

ii. Command Line

Sivasubramanian's "rds-create-database" command includes a port parameter (13:1-5). POSAs understood that the purpose of providing a command-line tool with an option for specifying a parameter value is to *receive an input* of that value (*supra* §VI.D.1.b.ii). Thus, Sivasubramanian's command-line interface

includes instructions to *receive* the claimed *third input*. Bhattacharjee, ¶179. Furthermore, POSAs would have found it obvious to add a "DNS address" parameter for the same reasons discussed *supra* §VI.D.1.d.i for the GUI; this implementation of Sivasubramanian's command-line interface includes instructions to *receive* two inputs that are each independently the claimed *third input*. Bhattacharjee, ¶180.

e. [1D]: "receive a fourth input comprising a selection of a level of a Service Level Agreement (SLA) from a plurality of levels of the SLA to protect the database after provisioning"

i. GUI

The '818 patent says "[a]n ***SLA is an agreement***… that outlines, among other things, the protection scope of the database. The protection scope defines for how long data from the database being created or registered is retained." EX1001, 19:19-53. The same storage duration may be used for snapshots and transactional logs. EX1001, 19:29-53. The patent provides examples of different "levels"—like "GOLD, SILVER, BRONZE"—of SLAs with different values for "parameters" such as how long "snapshots" are stored. Bhattacharjee, ¶181; EX1001, 20:65-21:18.

Sivasubramanian+Zha's database-creation GUI includes a UI element enabling selection an SLA level from multiple SLA levels, each SLA level specifying either a uniform snapshot/log retention period or separate periods.

Supra §VI.C.1. In either case, in Sivasubramanian+Zha, Sivasubramanian's GUI *receives* a *fourth input comprising a selection of a level of a Service Level Agreement (SLA) from a plurality of levels of the SLA*. Bhattacharjee, ¶182.

Additionally, POSAs would have understood that the purpose of receiving this selection is *to protect the database*. The '818 patent says ***"[p]rotecting a database means replicating a state of the database*** for creating a fully functional copy of the database"; this "***[r]eplicating***… may involve ***creating*** fully functional clones (e.g., ***back-ups***)," which may in turn "be created by periodically capturing snapshots of the source database." EX1001, 4:55-67; Bhattacharjee, ¶183. POSAs understood that in both Sivasubramanian+Zha variants discussed *supra* §VI.C, the snapshot and log data that is retained for the specified retention period is retained for the purposes of "replicating a state of the database for creating a fully functional copy of the database" (EX1001, 4:55-67), as explained below.

First, Sivasubramanian's own "approach for backing up data" (19:32), used in the first Sivasubramanian+Zha variant, performs "backup" on a "database" by capturing database "snapshots." Sivasubramanian, 19:20-64; *supra* §VI.C. Moreover, Sivasubramanian discusses "tak[ing] a snapshot of a production instance" so that a user can "clone" a database and ensure it is "fully populated" with "production [data]." Sivasubramanian, 15:1-15; *supra* §VI.C; Bhattacharjee, ¶184. Sivasubramanian's "approach" also backs up "log files," *i.e.* transaction logs

(*supra* §VI.A.1.b), that can "replay[] various transactions in case the data files have to be restored." Sivasubramanian, 19:65-20:1. POSAs understood the "data files" that are "restored" are "data files" for a database, since Sivasubramanian earlier says "backing up both the data files and any associated log files" is done to backup a "repository," *i.e.*, a "database." Sivasubramanian, 19:27-31; Bhattacharjee, ¶185. Thus, POSAs would have understood that in the first Sivasubramanian+Zha variant, both the snapshots and logs would be retained *to protect the database* by replicating its state, per Sivasubramanian's exemplary "approach for backup up data" (19:32). Bhattacharjee, ¶186.

Second, Zha's "virtual database" technique, used in the second Sivasubramanian+Zha variant (*supra* §VI.C), "create[s] a virtual database [VDB]," by "creat[ing] files that represent the ***information corresponding to [a] production database system… at a given point in time***." Bhattacharjee, ¶187; Zha, 6:49-58. The VDB can be used for Zha's "backup" process, which "can be used to reconstruct information of the original storage system in case the original data is lost." Zha, 25:24-34. Zha reconstructs the database by applying transaction logs to VDBs created using snapshots, as discussed *supra* §VI.B (citing Zha, 17:63-66). POSAs therefore understood that Zha's technique "replicat[es] a state of the database for creating a fully functional copy of the database" and thus "*protects*" the database, as the '818 patent describes, using the snapshots and transaction logs

retained for the user-defined retention period. EX1001, 4:55-67; Bhattacharjee, ¶188.

Additionally, POSAs would have understood that in Sivasubramanian+Zha, the *protecting* of the database would occur *after provisioning* because the SLA level selection occurs as part of gathering information for creating the database (*see infra* §VI.D.1.i (Limitation [1H])), which in Sivasubramanian includes *provisioning* as discussed *supra* §VI.A.1. Bhattacharjee, ¶189.

ii. Command Line

As discussed *supra* §VI.C.1, in Sivasubramanian+Zha, Sivasubramanian's "rds-create-database" command would include a parameter for specifying a *selection* of an SLA level from a list of SLA levels. Thus, in Sivasubramanian+Zha, Sivasubramanian's "rds-create-database" command *receives* the claimed *fourth input*; the input is *to protect the database after provisioning* for the same reasons explained *supra* §VI.D.1.e.i for the GUI. Bhattacharjee, ¶190.

f. [1E]: "receive a fifth input of a protection schedule comprising a first frequency of capturing a snapshot of the database after provisioning and at least one time period for capturing the snapshot;"

i. GUI

The patent uses "*time[-]period*" in the context of capturing snapshots to refer to a time of day at which a snapshot is captured. EX1001, 6:5-8 ("The

protection schedule may allow the user to define … the ***time-period*** for capturing daily, weekly, monthly, and/or quarterly snapshots"), 20:25-31 (referring to "another ***time period***" as an alternative to "the protection schedule defin[ing] that the snapshot captured at 11:00 AM is the daily snapshot"); Bhattacharjee, ¶191.

As discussed *supra* §VI.C.2, in Sivasubramanian+Zha, Sivasubramanian's database-creation GUIs would have accepted an input of a user-specified "***schedule***" as taught by Zha that includes (1) a "***frequency***" for "retriev[ing] PIT copies," *i.e.*, taking snapshots; and (2) "times during the day" for "retriev[ing]" snapshots. Zha, 11:33-43; Bhattacharjee, ¶192. The snapshots are for what the '818 patent calls "***protecting***" the database. *Supra* §VI.D.1.e.i; EX1001, 4:55-67. POSAs understood that taking/retrieving snapshots involved *capturing* snapshots. Bhattacharjee, ¶193; EX1001, 5:13-17 ("because ***capturing*** a snapshot requires quiescing… the source database… it is desirable to ***take*** only a minimum number of snapshots"), Heritage (EX1009), 277. Therefore, the "schedule" (Zha, 11:33-43) accepted by the GUI (*supra* §VI.C.2) is a *fifth input of a protection schedule comprising a first frequency of capturing a snapshot of the database and at least one time period for capturing the snapshot*. Bhattacharjee, ¶194.

Furthermore, because the schedule would be specified during Sivasubramanian's database creation process, which incudes *provisioning* (*supra* §VI.D.1.a), POSAs would have understood that capturing of snapshots at the

specified *frequency* starts once the provisioning process is complete; thus, the *frequency* is *of capturing a snapshot of the database after provisioning*. Bhattacharjee, ¶195.

ii. Command Line

In Sivasubramanian+Zha, Sivasubramanian's "rds-create-database" command *receives an input* specifying Zha's "schedule," as discussed *supra* §VI.C.2. This input is the claimed *fifth input* for the same reasons discussed *supra* §VI.D.1.f.i for the GUI. Bhattacharjee, ¶196.

g. [1F]: "present at least one user interface to receive the first input, the second input, the third input, the fourth input, and the fifth input;"

i. GUI

Sivasubramanian describes Figure 5 as an "example of a display for generating a request for action in the data plane to be submitted to a ***user interface*** of the control plane." Sivasubramanian, 2:7-13; Bhattacharjee, ¶¶197-198.[5] Sivasubramanian itself discloses or at least renders obvious that the Figure 5 UI receives the *first* through *third* inputs. *Supra* §§VI.D.1.b-VI.D.1.d ([1A]-[1C]). And in Sivasubramanian+Zha, the Figure 5 UI would be further modified to

[5] Sivasubramanian refers to "FIG. 6" as the "user interface" at 2:10-13. However, POSAs reading Sivasubramanian and viewing its figures understood that the descriptions of Figures 5 and 6 at 2:7-13 are reversed. Bhattacharjee, ¶198.

receive the claimed *fourth* and *fifth* inputs. *Supra* §§VI.D.1.e-VI.D.1.f ([1D]-[1E]). Thus, the Sivasubramanian+Zha Figure 5 GUI *receives* the required *first* through *fifth* inputs. Bhattacharjee, ¶199.

ii. Command Line

POSAs understood that Sivasubramanian's "command line tool" (12:66) is a *user interface*, because "user[s]"/"customer[s]" ***use*** it to issue requests to the control plane. Sivasubramanian, 12:64-67, 14:1-3; Bhattacharjee, ¶200. The command line tool in Sivasubramanian+Zha *receives* the *first* through *fifth* inputs for the reasons explained *supra* §§VI.D.1.b-VI.D.1.f for Elements [1A]-[1E] with respect to the command line.

h. [1G]: "determine an amount of storage for the database; and"

The '818 patent refers to the user specifying a "database size to determine how much storage is needed for storing" the database. EX1001, 18:53-59, 4:27-31; Bhattacharjee, ¶201. The exemplary UI also includes a database "size" field. EX1001, 24:57-25:2, Fig. 4E.

Sivasubramanian discloses the user of the control plane specifying a "***storage*** capacity" or "***storage*** size" for a database being created via both the Figure 5 GUI and the command line. Sivasubramanian, 11:49-52, 12:42-44, 12:62-13:5, Figure 5; Bhattacharjee, ¶¶202-203.

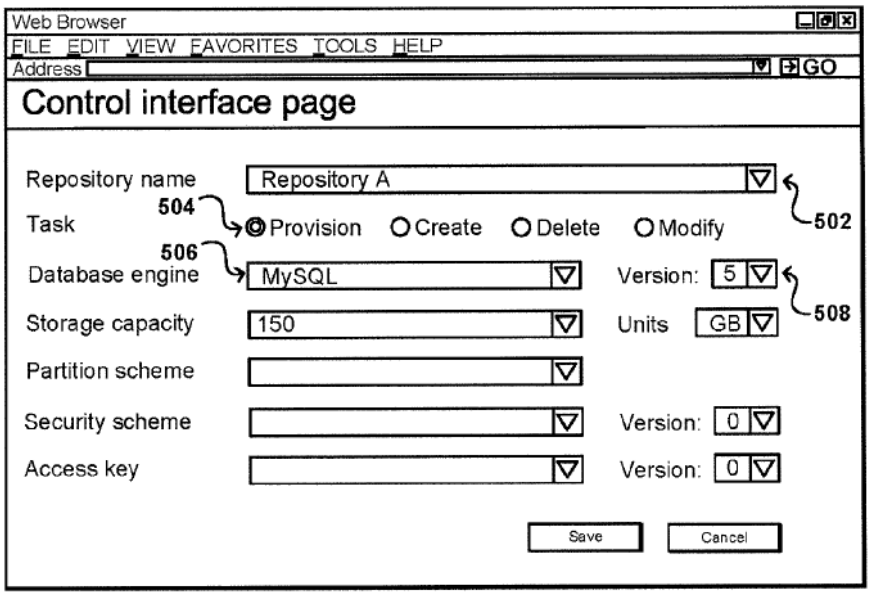


FIG. 5

Thus, POSAs would have understood that the "create database" workflow *determines an amount of storage for the database*, at least by accepting the user's input storage capacity/size. Bhattacharjee, ¶204.

If the claim requires calculating an amount of storage base on the user-specified value, this would have been obvious. Sivasubramanian's control plane performs the "necessary tasks" to create a database. Sivasubramanian, 9:59-61. POSAs understood that for a database to store a certain amount of data, the database needed some additional data storage due to the overhead of the database itself. Bhattacharjee, ¶205; Oracle (EX1012), 5; Quernemoen, Abstract, 5:22-44;

Wasserman (EX1034), [0076]; Lober (EX1035), [0004]. Thus, a POSA would have been motivated to implement the control plane to calculate *an amount of storage* necessary to accommodate the user's desired data storage capacity. Bhattacharjee, ¶206. POSAs would have reasonably expected success implementing such a calculation, since methods for such calculation were well-known. Bhattacharjee, ¶207; Oracle, 8; Quernemoen, Abstract, 5:22-44; Wasserman, [0076]; Lober, [0004].

i. [1H]: "create the database at the location based on the first input, the second input, the third input, the fourth input, the fifth input, and the amount of storage"

In Sivasubramanian+Zha, the claimed *first* through *fifth* inputs are received via user interfaces (the GUI or command line) for database creation. *Supra* §§VI.D.1.a-VI.D.1.f ([1PRE]-[1E]). POSAs would have understood that the purpose of accepting these inputs is to use them to "*create*" a database *based on* these inputs and *based on* the *amount of storage* determined from the user input (*supra* §VI.D.1.h ([1H])). Sivasubramanian, 11:47-52 ("customers can ***create*** a data store using a 'Create***Database***' or similar API. The user can call a Web service to specify ***any desired values*** for an instance type …, ***storage size***, ***repository name***, ***port***, ***and other such values***."); *id.*, 12:38-13:9 (describing creating database using Fig. 5 UI and "rds-create-database" command-line tool);

supra §VI.C (explaining "create database" workflow's use of parameters specified in SLA and protection schedule); Bhattacharjee, ¶208.

Furthermore, a POSA would have understood that the database is created *at the location* the user enters via either the GUI or command line. Sivasubramanian discloses that a database user "access[es] the data store instance" for the *database* "directly using the DNS address and port." Sivasubramanian, 7:10-12; *supra* §VI.D.1.d ([1C]). POSAs understood that this access is possible because the database is created at the network *location* indicated by the DNS address and port (not some other location). Bhattacharjee, ¶209.

2. Claim 2

Although claim 2 recites a "*hyperconverged infrastructure system*," "Hyperconverged" appears only in the '818 Patent's title, and is nowhere in the specification except when reciting titles of priority applications. EX1001, 1:9-16; Bhattacharjee, ¶210.

Sivasubramanian discloses that the *location* where databases are "create[d]" and "*provision[ed]*" is the "data plane." Sivasubramanian, 7:3-9 ("workflow component can manage execution" of "tasks" such as "'create database'" in the "data plane"), 2:40-44 ("Once provisioned, a user can… access… the data instance(s) in the "data plane"), 5:15-20, 12:55-66, 13:28-29; Bhattacharjee, ¶211.

Sivasubramanian states that "***in one embodiment***," the "data plane" is a "cloud" system. Sivasubramanian, 9:9-13. POSAs understood, however, that Sivasubramanian does not require that the data plane be a cloud. *See, e.g.*, Sivasubramanian, 24:19-28:56 (Sivasubramanian's claims not requiring cloud); Bhattacharjee, ¶212. Rather, POSAs understood that Sivasubramanian's disclosure is directed towards automating workflows such as creating/provisioning databases (*supra* §VI.A.1) but is flexible as to the environment where workflows are performed. Sivasubramanian, 4:48-58 ("aspects of the various embodiments can be used with any appropriate services available or offered over a network in an electronic environment"), 3:10-25, 9:35-44, 15:45-50, 23:17-24; Bhattacharjee, ¶213.

One well-known alternative to a cloud-based system was a "hyperconverged infrastructure" (HCI) system. EX1036, [0003] ("Hyper Converged Infrastructure (HCI) combines computing applications and storage applications into a single infrastructure."); EX1037, 1:13-33 (discussing prior art "hyper-converged infrastructure"); Bhattacharjee, ¶214. Some prior art described a "hyperconverged" infrastructure an infrastructure type used by private entities where a single apparatus includes servers that can each be configured to perform different roles as needed, such as computation, storage, or networking. Srinivasan (EX1038), 1:15-27; Bhattacharjee, ¶215.

In Sivasubramanian+Zha, POSAs would have found it obvious to implement Sivasubramanian's control plane to provision databases located in a "data plane" implemented as a *hyperconverged infrastructure system*. Bhattacharjee, ¶216. As noted above, HCI systems were known. Furthermore, it was known to deploy databases in HCI systems. EX1039, [0242] (referring to "hyperconverged database type management system"); EX1040, [0089]; Srinivasan 5:49-6:55, Fig. 2; Bhattacharjee, ¶217. POSAs would have been motivated to implement the data plane as an HCI system instead of as a cloud to achieve the known benefits of HCI, such as resource sharing, flexibility, ease of management, and control of hardware. Bhattacharjee, ¶218; EX1041, [0002]; EX1042, 1:5-10; EX1043, 1:6-20. POSAs would reasonably have expected success because Sivasubramanian's "control plane" is not limited to one type of data plane and because HCI systems and their use for databases were well-known, as discussed above. Bhattacharjee, ¶219.

In the resulting implementation of Sivasubramanian+Zha, the *location where the database is to be provisioned*, *i.e.*, the data plane, *is a hyperconverged infrastructure system*, thus meeting claim 2. Bhattacharjee, ¶220.

3. Claim 3

Sivasubramanian discloses that *location* where databases are "create[d]" and "provision[ed]" is the "data plane," as explained *supra* §VI.D.2. Sivasubramanian

discloses that the "data plane" may be a "***cloud***." Sivasubramanian, 9:4-17, 4:48-53; Bhattacharjee, ¶221.

4. Claim 4

Sivasubramanian *determines an amount of storage for the database* based on a "storage capacity" or "storage ***size***" input by the user. Sivasubramanian, 13:1-5, Figure 5; *supra* §VI.D.1.h ([1G]). Thus, in both the GUI and command line, this input is *a sixth input indicative of the size of the database to be provisioned*. Bhattacharjee, ¶222.

5. Claim 5

Sivasubramanian+Zha's database-creation GUIs receive an input of the *SLA* level (*supra* §VI.D.1.e ([1D])) and a schedule including the *first frequency of capturing the snapshot* and the *at least one time period for capturing the snapshot* (*supra* §VI.D.1.f ([1E])). In Sivasubramanian+Zha, the database is setup to have snapshots captured at the frequencies and times, and retained for the durations, specified in the database-creation GUIs. *Supra* §VI.C. Thus, POSAs would have understood this capture process *captures one or more snapshots* (1) *from the database* ***that is created***, and (2) ***based on*** the *first frequency* for capturing snapshots*, the at least one* [time] *period* during the day for capturing snapshots*, and the SLA*, which defines the snapshot retention period (*supra* §VI.C.1). Bhattacharjee, ¶¶223-224.

Sivasubramanian explains that the "workflow component" in the "control plane" controls the "repository creation" and "recovery and backup" "workflows." Sivasubramanian, 6:31-42; Bhattacharjee, ¶225. And Sivasubramanian's claim 20 recites a system with "at least one processor" that "generate[s] a workflow" based on "request." Sivasubramanian, 27:18-44. Thus, POSAs understood Sivasubramanian to disclose a single "processor" executing all "control plane" instructions for controlling all "workflows," including the "create database" workflow and the "recovery and backup" workflow. Bhattacharjee, ¶225. At minimum, this would have been obvious because one processor performing multiple functions was a known and simple, and thus expectedly successful, way to beneficially accomplish Sivasubramanian's teachings. Bhattacharjee, ¶225. In such an implementation, the processor executing the "create database" instructions, which is *the processor* (*supra* §VI.D.1.a [1Pre])), also *executes computer readable instructions* to perform the capture.

6. Claim 6

The body of claim 6 lacks a verb where similar claim 13 has the verb "capture." *See* claim listing *infra* §X.

Sivasubramanian+Zha's database-creation GUIs receive input of the *SLA* level (*supra* §VI.D.1.e ([1D]) that specifies a retention period for transaction logs (*see supra* §VI.C.1). Bhattacharjee, ¶¶226-227. Additionally, the database-

creation GUIs receive an input of a schedule including "the ***frequency*** and the times at which the transaction log manager 320 retrieves updates to online transaction logs" (Zha, 11:33-43). *Supra* §VI.C.2. POSAs understood that "retriev[ing] updates" to a log efficiently *captures* the log as it exists at retrieval time; thus, the "frequency" for "retriev[ing] updates" is *a second frequency for capturing one or more* transaction logs, *i.e.*, *transactional logs*. EX1001, 36:58-65 ('818 patent using "transaction log" and "*transactional log*" interchangeably); Bhattacharjee, ¶228; Zha, 15:21-24 (describing log update process); Heritage, 277.

In Sivasubramanian+Zha, the database is setup to have transaction logs captured at the frequencies, and retained for the durations, specified in the database creation GUIs. *Supra* §VI.C. Thus, POSAs would have understood that this capture process *captures one or more transactional logs* (1) *from the database* ***that is created***, and (2) ***based on*** *the SLA and* the *second frequency*. Bhattacharjee, ¶229. POSAs would have found it obvious to implement *the processor* of claim 1 to *execute* the required *instructions* to perform this capture for the reasons discussed above for claim 5. Bhattacharjee, ¶230.

7. Claim 7

POSAs would have been motivated to implement Sivasubramanian's database-creation UIs to *receive a sixth input of a compute profile indicative of a number of vCPUs and memory for provisioning the database* because both of these

parameters were well-known database parameters; this modification would be expectedly successful because it was within a POSA's skill (*supra* §VI.A.2). Bhattacharjee, ¶231; RDS, 14 (user can select "DB Instance Class" with certain amount of "memory"), 104-105 (different "instance classes" have different amounts of "Memory (GB)" and numbers of "vCPU[s]"); EX1044, [0032].

8. Claim 22

The patent describes "direct-attached storage" as including "storage components that are provided internally," "[i]n contrast to" network-attached storage "accessible via the network." EX1001, 6:42-56. Fig. 1 (annotated below); Bhattacharjee, ¶232.

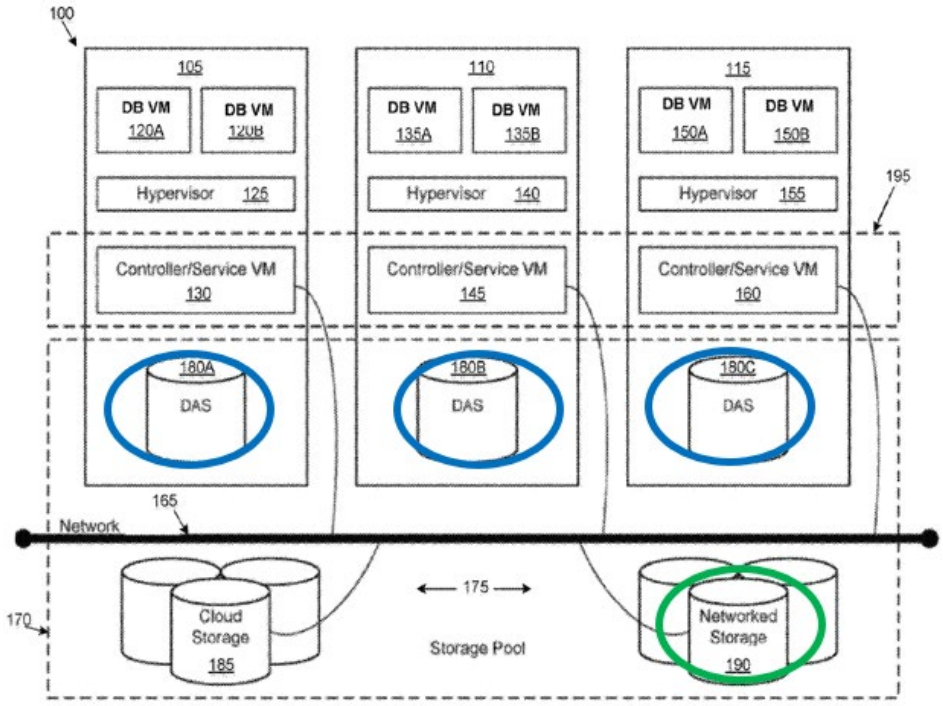


FIG. 1

In Sivasubramanian+Zha, the control plane *creates* the database by executing Sivasubramanian's database-creation "workflow" (*supra* §VI.D.1.i ([1H])), which performs "tasks" including "allocating a volume of" "persistent storage, [and] attaching the persistent storage volume to the data store instance." Sivasubramanian, 6:43-65; Bhattacharjee, ¶233. Thus, POSAs would have understood that the workflow "attach[es]" the "data store instance" on which the *database* runs (*supra* §VI.A.1), to "persistent storage." Bhattacharjee, ¶234.

Sivasubramanian also discloses that its invention's "operating environment" includes "storage media" that is "***local*** to (and/or resident in)" computers "***or remote***" from computers "across the network." Sivasubramanian, 23:17-24;

Bhattacharjee, ¶235. Thus, POSAs understood Sivasubramanian to disclose creating a database that may be attached to, *i.e.*, *associated with*, "local," *i.e.*, *direct attached*, "***storage***," "or" storage that is "remote" across the "*network*," *i.e.*, *network attached*. Bhattacharjee, ¶235; EX1001, 38:50-57.

POSAs understood Sivasubramanian to also disclose the option of associating one database with ***both*** local and remote storage. Sivasubramanian, 3:11, Fig. 1; Bhattacharjee, ¶236. At minimum this would have been obvious, for multiple reasons. Mixing direct-attached and network-attached storage was well-known. Bhattacharjee, ¶237; EX1045, [0034]; EX1046, [0030]. Furthermore, POSAs would have been motivated to implement a mix of the two by Sivasubramanian; Sivasubramanian discloses using the control plane "scaling," (*e.g.*, 5:32-35), and POSAs would have understood that using both storage types would allow a database to scale up in storage size by utilizing the network-attached storage when the direct-attached storage was full. Bhattacharjee, ¶238. POSAs would reasonably have expected success in such an implementation since Sivasubramanian discloses both types of storage and using both types together was well-known, as noted above. Bhattacharjee, ¶238.

9. Claim 23

Sivasubramanian discloses associating a database with local (i.e. *direct attached*) "or" remote (i.e. *network* attached) storage, and thus discloses the

database *associated with either* the former *or* the latter. Sivasubramanian, 23:17-24; *supra* §VI.D.8 (claim 22); Bhattacharjee, ¶239.

10. Claim 28

Sivasubramanian's "rds-create-database" tool accepts the parameter "identifier" that is separate from the parameter "dbName." Sivasubramanian, 13:1-5. POSAs understood that the "identifier" is an example of a *database description*. The '818 patent does not define "***database description***," but describes element 460B in Figure 4E, which is simply a text input field distinct from the "name" field 460A (EX1001, 24:61), as a "description" "for the database being created." EX1001, 24:60-65; Bhattacharjee, ¶240.

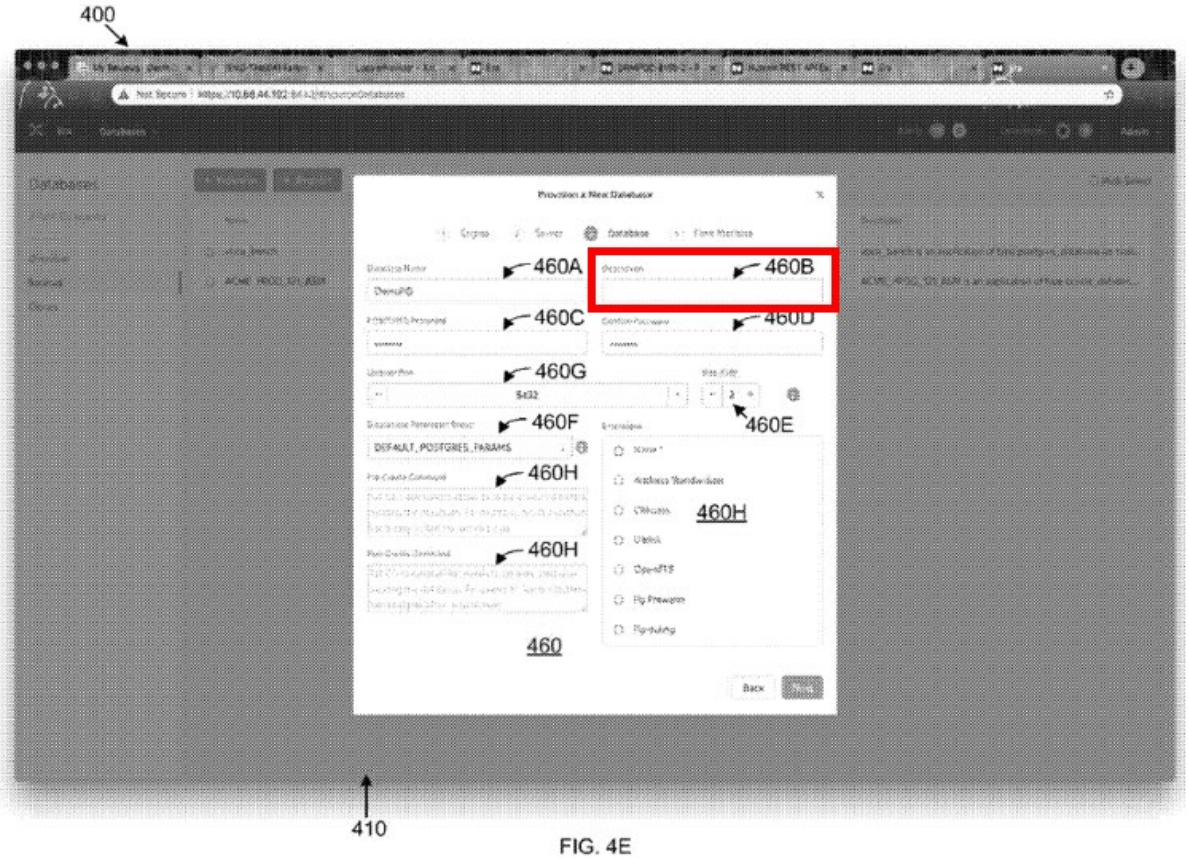


FIG. 4E

Thus, *the processor further executes computer-readable instructions to receive a sixth input of a database description for the database* via the command-line. POSAs would have likewise found it obvious to include instructions for receiving a description in Sivasubramanian's Figure 5 GUI; this would beneficially provide consistency with the command-line interface, and was within a POSA's skill. Bhattacharjee, ¶241; *supra* §VI.A.2.

11. Claim 29

The '818 patent says, "the ***continuous protection*** parameter within the SLA defines the ***duration*** in days for which ***all*** captured snapshots and transactional logs are retained." EX1001, 19:52-54.

In Sivasubramanian+Zha, the user specifies a retention period for snapshots and for transaction logs, as Zha teaches. *Supra* §VI.C. Zha is silent as to whether the same retention period applies to all snapshots, and is also silent as to whether the same retention period applies to both snapshots and transaction logs. However, a POSA would at minimum have found specifying one retention period of some number of days applicable to both snapshots and logs to be obvious. A uniform retention period measured in days was well-known in the art and would beneficially simplify the management of snapshots and logs, and POSAs would have reasonably expected success because it was simple to apply a one duration to all snapshots. Bhattacharjee, ¶¶242-246; Tekade, [0046]; Bhasin, [0009]; RDS, 5, 111-12. In this implementation, the single retention period applicable to all snapshots and transaction logs is a *first duration for a continuous protection of the database*. Bhattacharjee, ¶247.

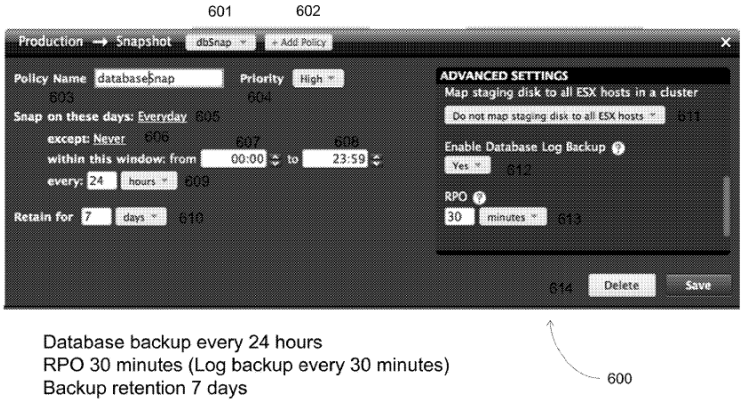
In Sivasubramanian+Zha, the retention period is specified by the SLA level. *Supra* §VI.D.1.e ([1D]). Thus, *each of the plurality of levels of the SLA defines… a first duration for a continuous protection of the database*. This satisfies claim 29

if, due to the claim language "***one or*** *more of*," only the *first duration* is required. *See also* EX1001, 39:45-59; Bhattacharjee, ¶248.

If the claim requires *all* the recited items, this would likewise have been obvious. The SLA may "define a ***daily protection*** parameter" defining a "***duration***" for storing "a daily snapshot," (19:62-64), and "may" similarly define "weekly"/"monthly"/"quarterly" "protection parameters" for storing "weekly"/"monthly"/"quarterly" snapshots. EX1001, 20:8-55; Bhattacharjee, ¶¶249-250. The patent describes the "daily" snapshot as the snapshot taken at a "designated" daily time; this snapshot is stored for the "daily protection parameter" duration instead of the shorter "continuous protection parameter" duration (*e.g.,* 90 days v. 30). EX1001, 19:67-20:7, 21:11-17 ("GOLD" SLA example); Bhattacharjee, ¶251. "Weekly," "monthly," and "quarterly" snapshots can be similarly designated. EX1001, 20:8-55; Bhattacharjee, ¶252.

Designating specific snapshots of data as daily/weekly/monthly/quarterly snapshots was well-known. EX1047, [0165]; Madany (EX1023), [0023]; EX1048, [0060]; Taylor (EX1049), [0164]; Deshmukh (EX1050), [0053]; EX1051, [0025]; Bhattacharjee, ¶253. Having different retention periods for snapshots collected at different intervals was also known in the art. Tekade, [0046], Fig. 6 (showing UI for creating separate snapshot policies with configurable collection days/times and

storage durations); Gaurav (EX1052), 11:14-20, Fig. 2; EX1053, 4:67-5:6; Deshmukh, [0053]; Taylor, [0164]; Bhattacharjee, ¶254.



Tekade, Fig. 6

POSAs would have understood that having a ***minimum*** duration for all snapshots (*i.e.*, the *continuous protection parameter*) ***and*** different durations for daily/weekly/monthly/quarterly snapshots beneficially helped lengthen the historical period for recovery while conserving snapshot storage space. Madany, [0023]; Taylor, [0164]; EX1054, [0232]-[0234]; Bhattacharjee, ¶255. POSAs would have been motivated to have a single SLA level specify values for each duration type to beneficially increase user convenience, and POSAs would have reasonably expected success because such a UI was within a POSA's skill (*supra* §VI.A.2). Bhattacharjee, ¶256.

12. Claim 30

In Sivasubramanian+Zha, the *at least one time period* is met by Zha's "times during the day" for *capturing* the "point in time" copies, i.e., *snapshots*. Zha, 11:33-43; *supra* §VI.D.1.f ([1E]). As explained *supra* §VI.D.11 for claim 29, in Sivasubramanian+Zha, it would also have been obvious to designate a time of day for collecting a *daily* snapshot that is retained longer than the standard duration. POSAs would have found it obvious to implement Sivasubramanian+Zha's schedule UI (*supra* §VI.C.2) to additionally accept input of the daily snapshot time, as this was a known UI field that beneficially let a user choose the best daily snapshot time based on their own circumstances. RDS, 16-17, 187 (database-creation options include a daily "backup window" during which "snapshot" is taken); Gaurav, 11:14-20, Fig. 2; Bhattacharjee, ¶257. POSAs would have expected success because the UI modification was within a POSA's skill (*supra* §VI.A.2). In such an implementation, *the at least one time period comprises* the "times during the day" for capturing snapshots (Zha, 11:33-43), and also *comprises* the *time of day* for *capturing* the specific snapshot designated each day as the *daily snapshot*. Bhattacharjee, ¶258. This satisfies claim 30 if, due to the claim language "*one or more of…or,*" only the *daily* time is required.

If the claim so requires, the remaining items would have also been obvious. Designating and collecting weekly/monthly/quarterly snapshots would have been

obvious (*supra* §VI.D.11), and POSAs would have found it obvious for the schedule UI to accept user input of the corresponding time designations—*day of week*, *day of month*, and start of quarter (*group of months*, *see* EX1001, 20:47-55)—for the same user flexibility reasons discussed above for the *time of day* designation. Bhattacharjee, ¶¶259-264.

13. Claims 8-14, 24-25, 31-33

Claim 8's preamble recites "*[a] system comprising a memory having computer-readable instructions stored thereon to provision a database; and a processor that executes the computer-readable instructions to*." Claim 8's remaining elements are identical to [1A]-[1H]. *Infra* §X. Sivasubramanian's control plane is implemented by a "***system***." *See*, *e.g.*, Sivasubramanian, claim 20 ("A system for controlling a data environment…"). This system has *a memory having computer-readable instructions stored thereon to provision a database and a processor that executes the computer-readable instructions* for the reasons explained *supra* §VI.D.1.a ([1Pre]). Sivasubramanian+Zha meets [8A]-[8H] for the reasons explained *supra* §§VI.D.1.b-VI.D.1.i for identical Elements [1A]-[1H]. Bhattacharjee, ¶265.

Claims 9-14, 24-25, and 31-33 depend directly or indirectly from claim 8 but recite identical limitations to claims 2-7, 22-23, and 28-30, respectively. *Infra* §X. Sivasubramanian+Zha therefore meets claims 9-14, 24-25, and 31-33 for the same

reasons explained *supra* §§VI.D.2-VI.D.7, VI.D.8-VI.D.9, and VI.D.10-VI.D.12, respectively. Bhattacharjee, ¶266.

14. Claims 15-21, 26-27, 34-36

Claim 15 recites a *method comprising* certain steps in Limitations [15A]-[15H]. Limitation [15A] recites "*receiving, by a processor executing computer-readable instructions to provision a database*" the same *first input* recited in Limitation [1A]. Limitations [15B]-[15H] recite "*the processor*" performing steps identical to the actions recited in Limitations [1B]-[1H].

As discussed *supra* §VI.D.1.a for Limitation [1Pre], Sivasubramanian discloses *a processor executing computer-readable instructions to provision a database*. This meets the "*processor*" recited in [15A]-[15H]. Bhattacharjee, ¶¶267-268. POSAs understood that when the processor executes those instructions, it performs a *method comprising* the steps corresponding to those instructions. *See*, *e.g.*, Sivasubramanian, 2:17-18 ("***methods*** in accordance with various embodiments of the present disclosure"), claim 4 (25:4-6) ("A computer-implemented ***method***…"); Bhattacharjee, ¶268. This meets [15PRE] as well as the "*by the processor*" aspect of each of limitations [15A]-[15H]. Bhattacharjee, ¶268.

Claims 16-21, 26-27, and 34-36 depend directly or indirectly from claim 15 but recite additional limitations that are identical to the additional limitations in claims 2-7, 22-23, and 28-30, respectively. *Infra* §X. Sivasubramanian+Zha

therefore meets claims 16-21, 26-27, and 34-36 for the same reasons explained *supra* §§VI.D.2-VI.D.7, VI.D.8-VI.D.9, and VI.D.10-VI.D.12, respectively. Bhattacharjee, ¶269.

VII. GROUND 2: CLAIMS 1-36 ARE UNPATENTABLE OVER SIVASUBRAMANIAN+SHEKAR+ZHA

A. Shekar (EX1007)

Shekar discloses a "[h]yperconverged infrastructure" ("HCI") providing "storage" and "compute" capabilities. Shekar, Title, [0014], [0032], Fig. 1. Shekar's goal is to provide a "storage device" with "high performance." Shekar, [0005].

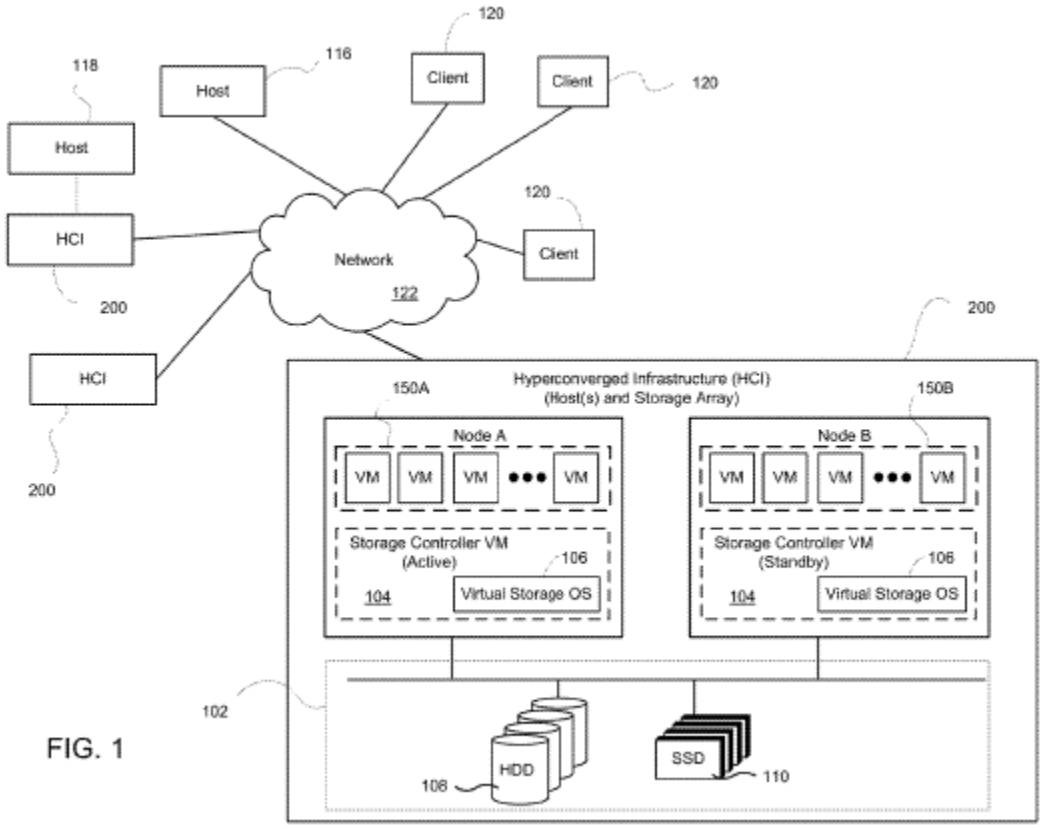


FIG. 1

Shekar, Fig. 1

Shekar's HCI includes two "nodes" that each include a "virtual storage connect module" that connects the node to a "storage array." Shekar, [0037], [0043], Fig. 2A; Bhattacharjee, ¶¶270-271. The storage array "provide[s] storage services to one or more host applications executing on the corresponding HCI 200." Shekhar, [0035]; Bhattacharjee, ¶272. The applications run on "guest virtual machines [VMs]" executing on the hardware of each node. Shekar, [0008], [0038], Fig. 2A; Bhattacharjee, ¶272. Each node also includes a "storage VM" (or "storage controller VM"). Shekar, [0033], Fig. 2A. The storage VMs on the two nodes "cooperatively manage access to the storage array" by using an "active/standby" scheme where the standby controller can take over access to the array if the active controller fails. Shekar, [0033], [0040]-[0042]; Bhattacharjee, ¶¶273-274.

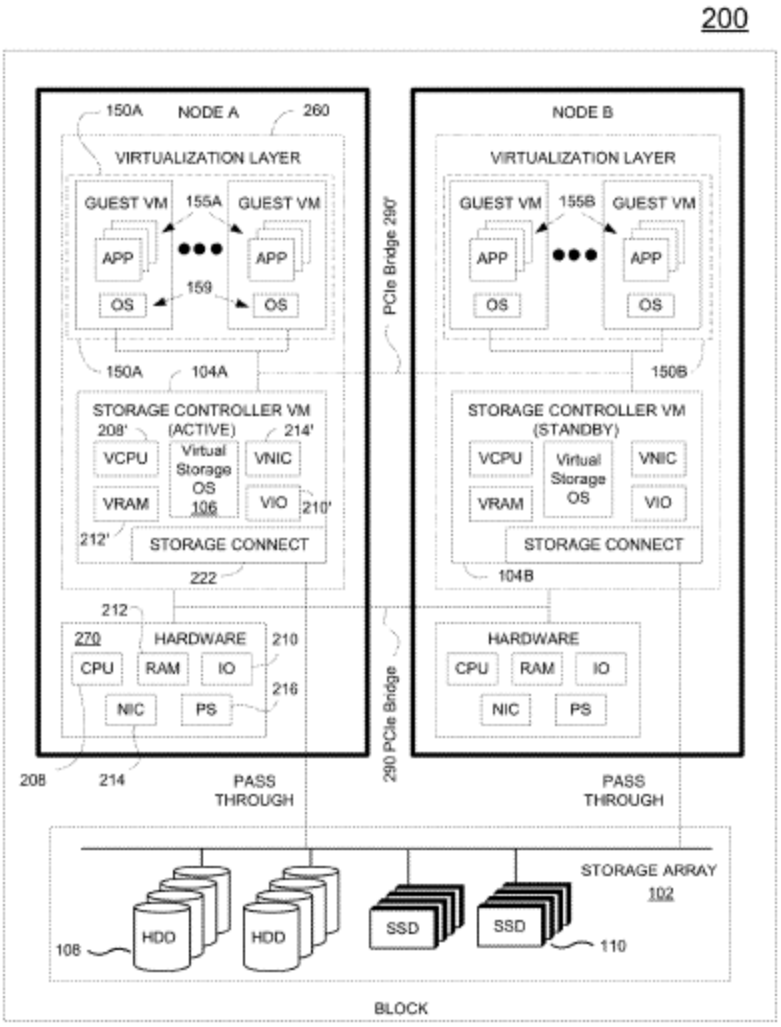


FIG. 2A

Shekar, Fig. 2A

B. Sivasubramanian+Shekar+Zha Combination

As discussed *supra* §VI.D.2, Sivasubramanian's control plane is flexible regarding data plane type. In view of Shekar, POSAs would have been motivated to provide the option of a "data plane" implemented using one or more of Shekar's "HCIs," for multiple reasons. Bhattacharjee, ¶275. ***First***, as explained *supra* §VI.D.2 (claim 2, Ground 1), POSAs understood that hyperconverged

infrastructure was a known option for implementing a "data plane," and further understood that this option had known benefits. Bhattacharjee, ¶276. ***Second***, POSAs would have been specifically motivated to use Shekar's "HCI[s]" (discussed *supra* §VII.A) to beneficially achieve the "high performance" of Shekar's HCIs (Shekar, [0005]) in a "data plane." Bhattacharjee, ¶277. ***Third***, as noted *supra* §VII.A, Shekar discloses that each HCI includes a "storage array" that provides "storage services" to "host applications" running in the HCI. Shekar, [0035]. POSAs understood that "database[s]" created using Sivasubramanian's "create database" "workflow" (Sivasubramanian, 6:59-56; *supra* §VI.A.1) are the types of applications that would make use of such storage services, and thus would benefit from the incorporation of Shekar's HCI. Bhattacharjee, ¶278; Sivasubramanian, 6:43-58.

POSAs would reasonably have expected success in implementing a data plane of Sivasubramanian using one or more of Shekar's "HCIs," for multiple reasons. Using hyperconverged systems for databases was known in the art. *Supra* §VI.D.2. Moreover, POSAs understood that Shekar's "HCI" includes components functionally equivalent to components that Sivasubramanian describes as being used to run databases in the data plane. Bhattacharjee, ¶279. For example, Shekar's HCI includes a "storage array" providing "storage services" to "host applications" running in the HCI. Shekar, [0035] (discussed *supra* §VII.A).

Similarly, Sivasubramanian discloses that a "data instance" for a database (*see supra* §VI.A.1), is "allocate[ed]" and "attach[ed]… to" "off-instance persistent storage." Sivasubramanian, 6:59-65. As another example, each of Shekar's "virtual host systems" (also called "VMs") on which "applications" run includes a "VCPU" (virtual CPU) and "virtual memory (RAM)." Shekar, [0046]. Similarly, Sivasubramanian discloses that "instances" for databases may have an "instance type" that "describes the CPU and memory capacity." Sivasubramanian, 11:49-52; Bhattacharjee, ¶279. Furthermore, Shekar explains that the nodes in an HCI include "one or more network interface cards (NICs)… for exchanging data packages through a network." Shekar, [0039]. Thus, POSAs understood that Sivasubramanian's control plane would be able to communicate with the HCIs in order to execute workflows. Bhattacharjee, ¶279.

The resulting Sivasubramanian+Shekar system includes the "control plane" of Sivasubramanian (*see supra* §VI.A) able to execute workflows in a data plane implemented using one or more of Shekar's HCIs (*see supra* §VII.A). Furthermore, POSAs would have found it obvious to incorporate Zha's teachings regarding user-defined snapshot and transaction log collection frequencies, times during the day, and/or retention periods, and also incorporate UI elements to gather such information, into the Sivasubramanian+Shekar system for the same reasons discussed *supra* §VI.C for both variants of the Sivasubramanian+Zha combination.

Bhattacharjee, ¶280. The resulting Sivasubramanian+Shekar+Zha combination is identical to the Sivasubramanian+Zha combination discussed *supra* §VI in Ground 1 but additionally includes the ability to execute workflows in a data plane implemented using one or more of Shekar's HCIs. Bhattacharjee, ¶281.

C. Claim-by-Claim Analysis

1. Claims 1, 3-8, 10-15, 17-36

The Sivasubramanian+Shekar+Zha system is identical to the Sivasubramanian+Zha system except for the option of a data plane implemented with Shekar's HCI(s). *Supra* §VII.B Thus, Sivasubramanian+Shekar+Zha meets claims 1, 4-8, 11-15, and 18-36 same reasons explained *supra* §§VI.D.1 and VI.D.4-VI.D.14 for those claims, since the analysis of those claims is unaffected by the data plane implementation. Sivasubramaian+Shekar+Zha meets claims 3, 10, and 17 for the same reasons given in Ground 1 when the cloud is chosen. Bhattacharjee, ¶282.

2. Claims 2, 9, 16

As discussed *supra* §VII.C.1, Sivasubramanian+Shekar+Zha meets independent claims 1, 8, and 15, from which claims 2, 9, and 16, respectively, depend. Sivasubramanian discloses that the *location* where databases are "*provision[ed]*" is a "data plane." *Supra* §VI.D.2; Sivasubramanian, 7:3-9, 2:40-44, 5:15-20, 12:55-66, 13:28-29; Bhattacharjee, ¶283. In Sivasubramanian+Shekar+Zha, a data plane may be implemented using Shekar's

"*Hyperconverged Infrastructure (HCI)*," which Shekar describes as an "integrated ***system***." Shekar, [0037]; *supra* §VII.B. Thus, when the HCI option is chosen, the *location where the database is to be provisioned comprises a hyperconverged infrastructure system*, meeting claims 2, 9, and 16. Bhattacharjee, ¶283.

VIII. NO BASIS EXISTS FOR DISCRETIONARY DENIAL

A. 315(a)

This Petition presents "compelling evidence of unpatentability," which should end the *Fintiv* analysis. Director's Interim Procedure for Discretionary Denials, 5 (June 21, 2022) ("Interim Procedure") (discussing *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper 11 (Mar. 20, 2020) (precedential)).

If considered, the *Fintiv* factors weigh against denial.

1. Stay Potential

This factor is neutral because no stay in favor of IPR has yet been requested, so there is no "specific evidence" how the court might rule. *Sand Revolution II v. Continental Intermodal Group-Trucking,* IPR2019-01393, Paper 24, 7 (June 16, 2020) (informative).

2. Trial Timing

This weighs against denial. *Fintiv*, 3 & n.3. No trial has yet been set. Given the Northern District of California's median time to trial (47.9 months)[6], institution

[6] *See* https://www.uscourts.gov/file/79295/download, at 9.

would mean FWD before trial, further favoring institution. Interim Procedure, 8-9 & 4 n.12. Petitioner diligently filed this Petition roughly eight months after complaint service, before receiving infringement contentions, and before answering (EXs. 1056-1057). *Apple v. Seven Networks*, IPR2020-00156, Paper 10, 9 & n.8 (June 15, 2020) ("*Seven*").

3. Litigation Investment

This weighs against denial. Petitioner filed this petition before any substantive orders in the litigation or significant investment. *Seven*, 11-12 (quoting *Fintiv*, 11).

4. Issue overlap

This weighs against denial. Petitioner challenges all claims; PO has not served contentions. The Board should institute to enhance patent system integrity and meet Petitioner's interest in resolving invalidity challenges on all claims. *Seven*, 18.

5. Litigation Defendants

This is neutral; Petitioner is a litigation defendant, which is not unusual. *Sand Revolution II*, 12-13.

6. Other Circumstances

The Petition's "strong showing on the merits" weighs against denial. *Seven*, 20.

B. 325(d)

Under two-part framework in *Advanced Bionics, LLC v. Med-El Elektromedizinische Gerate GMBH*, IPR2019-01469, Paper 6 (Feb. 13, 2020) (precedential), the Board should not exercise its §325(d) discretion.

1. Part One

As the Director recently noted, "application of section 325(d) should be limited to situations in which the prior art or arguments were ***meaningfully*** addressed by the Office"—meaning the Office "***articulated*** its consideration of the art or arguments ***in the record***." 89 Fed. Reg. 28693, 28700 (Apr. 19, 2024). The record here lacks such articulation. *See* EX1002. Although Zha and a descendant of Sivasubramanian are listed on an IDS signed by the examiner (EX1002, 281-284), the examiner did not articulate consideration of them (*supra* §IV.C). That weighs against discretionary denial. PTAB Rules for 325(d) Considerations, 89 Fed. Reg. 28693, 28705 (2024) (to be codified at 37 C.F.R. §42.108(f)) (proposed Apr. 19, 2024) (discretionary denial under §325(d) applies only if prior art was "meaningfully addressed"). Further, under *Becton, Dickinson* Factor (d), the Examiner never assessed patentability using the Grounds' ***combinations*** in the Grounds (or any combination).

2. Part Two

If the Board reaches *Advanced Bionics* step two because Zha and a descendant of Sivasubramanian were of record, "specific teachings of [both

references] impact[ing] patentability of the Challenged Claims" were "misapprehend[ed] or overlook[ed]." *Cellco P'ship v. Huawei Device Co.*, IPR2020-01117, Paper 10 at 12 (Feb. 3, 2021). That neither reference formed a "basis of rejection weighs strongly against exercising [the Board's] discretion to deny… under 35 U.S.C. § 325(d)." *Id.* at 13. Furthermore, the three "*Becton, Dickson*" ("*BD*") factors applicable to Part Two (*Advanced Bionics*, 10) weigh against denial.

Factor (c) weighs against denial because the examiner did not use any reference from the Grounds in a rejection. *Group III v. Targus*, IPR2021-00371, Paper 21, 33 (July 9, 2021); 89 Fed. Reg. 28693, 28705 (325(d) discretionary denial only applies if prior art was "meaningfully addressed").

The examiner did not identify or explain any clam limitation that either Sivasubramanian or Zha, or the combination of the two, failed to disclose. Allowing the claims was error material to patentability because—as Ground 1 explains—Sivasubramanian+Zha renders all claims obvious (*supra* §VI) (as does Sivasubramanian+Shekar+Zha in Ground 2 *supra* §VII)). Factor (e) thus weighs against denial.

The prosecution did not include the expert testimony and evidence of a POSA's background knowledge presented in this Petition regarding why the claims are obvious. Thus, Factor (F) weighs against denial.

IX. CONCLUSION

The Board should institute review and cancel claims 1-36

Dated: December 13, 2024

Respectfully submitted,
Tessell, Inc.

/Anant Saraswat/_____
Anant K. Saraswat, Reg. No. 76,050
WOLF, GREENFIELD & SACKS, P.C.

X. CLAIM LISTING

The following claim listing assigns element labels (e.g., [1A], [1B], etc.) to certain claims for clarity.

Claim 1
[1PRE] A non-transitory computer-readable media comprising computer-readable instructions stored thereon for provisioning a database, wherein the computer-readable instructions when executed by a processor cause the processor to:
[1A] receive a first input of a database name for the database;
[1B] receive a second input of a database engine for the database;
[1C] receive a third input indicative of a location where the database is to be provisioned;
[1D] receive a fourth input comprising a selection of a level of a Service Level Agreement (SLA) from a plurality of levels of the SLA to protect the database after provisioning;
[1E] receive a fifth input of a protection schedule comprising a first frequency of capturing a snapshot of the database after provisioning and at least one time period for capturing the snapshot;
[1F] present at least one user interface to receive the first input, the second input, the third input, the fourth input, and the fifth input;
[1G] determine an amount of storage for the database; and
[1H] create the database at the location based on the first input, the second input, the third input, the fourth input, the fifth input, and the amount of storage.
Claim 2
The non-transitory computer-readable media of claim 1, wherein the location where the database is to be provisioned comprises a hyperconverged infrastructure system.
Claim 3
The non-transitory computer-readable media of claim 1, wherein the location where the database is to be provisioned comprises a cloud.
Claim 4
The non-transitory computer-readable media of claim 1, wherein the amount of storage for the database is determined based on a sixth input indicative of a size of the database to be provisioned.

Claim 5
The non-transitory computer-readable media of claim 1, wherein the processor further executes computer-readable instructions to capture one or more snapshots from the database that is created based on the first frequency, the at least one period, and the SLA.
Claim 6
[6PRE] The non-transitory computer-readable media of claim 1, wherein the processor further executes computer-readable instructions to:
[6A] one or more transactional logs from the database that is created based on the SLA and a second frequency for capturing a transactional log.
Claim 7
The non-transitory computer-readable media of claim 1, wherein the processor further executes computer-readable instructions to receive a sixth input of a compute profile indicative of a number of vCPUs and memory for provisioning the database.
Claim 8
[8PRE] A system comprising: a memory having computer-readable instructions stored thereon to provision a database; and a processor that executes the computer-readable instructions to:
[8A] receive a first input of a database name for the database;
[8B] receive a second input of a database engine for the database;
[8C] receive a third input indicative of a location where the database is to be provisioned;
[8D] receive a fourth input comprising a selection of a level of a Service Level Agreement (SLA) from a plurality of levels of the SLA to protect the database after provisioning;
[8E] receive a fifth input of a protection schedule comprising a first frequency of capturing a snapshot of the database after provisioning and at least one time period for capturing the snapshot;
[8F] present at least one user interface to receive the first input, the second input, the third input, the fourth input, and the fifth input;
[8G] determine an amount of storage for the database; and
[8H] create the database at the location based on the first input, the second input, the third input, the fourth input, the fifth input, and the amount of storage.
Claim 9
The system of claim 8, wherein the location where the database is to be provisioned comprises a hyperconverged infrastructure system.

Claim 10
The system of claim 8, wherein the location where the database is to be provisioned comprises a cloud.
Claim 11
The system of claim 8, wherein the amount of storage for the database is determined based on a sixth input indicative of a size of the database to be provisioned.
Claim 12
The system of claim 8, wherein the processor further executes computer-readable instructions to capture one or more snapshots from the database that is created based on the first frequency, the at least one period, and the SLA.
Claim 13
[13PRE] The system of claim 8, wherein the processor further executes computer-readable instructions to:
[13A] capture one or more transactional logs from the database that is created based on the SLA and a second frequency for capturing a transactional logs.
Claim 14
The system of claim 8, wherein the processor further executes computer-readable instructions to receive a sixth input of a compute profile indicative of a number of vCPUs and memory for provisioning the database.
Claim 15
[15PRE] A method comprising:
[15A] receiving, by a processor executing computer-readable instructions to provision a database, a first input of a database name for the database;
[15B] receiving, by the processor, a second input of a database engine for the database;
[15C] receiving, by the processor, a third input indicative of a location where the database is to be provisioned;
[15D] receiving, by the processor, a fourth input comprising a selection of a level of a Service Level Agreement (SLA) from a plurality of levels of the SLA to protect the database after provisioning;
[15E] receiving, by the processor, a fifth input of a protection schedule comprising a first frequency of capturing a snapshot of the database after provisioning and at least one time period for capturing the snapshot;
[15F] presenting, by the processor, at least one user interface to receive the first input, the second input, the third input, the fourth input, and the fifth input;
[15G] determining, by the processor, an amount of storage for the database; and

[15H] creating, by the processor, the database at the location based on the first input, the second input, the third input, the fourth input, the fifth input, and the amount of storage.
Claim 16
The method of claim 15, further comprising provisioning the database on a hyperconverged infrastructure system.
Claim 17
The method of claim 15, further comprising provisioning the database on a cloud.
Claim 18
The method of claim 15, further comprising receiving, by the processor, a sixth input indicative of a size of the database to be provisioned, wherein the amount of storage for the database is determined based on the size.
Claim 19
The method of claim 15, further comprising capturing, by the processor, one or more snapshots from the database that is created based on the first frequency, the at least one period, and the SLA.
Claim 20
[20PRE] The method of claim 15, further comprising:
[20A] capturing, by the processor, one or more transactional logs from the database that is created based on the SLA and a second frequency for capturing a transactional log.
Claim 21
The method of claim 15, further comprising receiving, by the processor, a sixth input of a compute profile indicative of a number of vCPUs and memory for provisioning the database.
Claim 22
The non-transitory computer-readable media of claim 1, wherein the database is associated with a direct attached storage and a network attached storage.
Claim 23
The non-transitory computer-readable media of claim 1, wherein the database is associated with either a direct attached storage or a network attached storage.
Claim 24
The system of claim 8, wherein the database is associated with a direct attached storage and a network attached storage.
Claim 25
The system of claim 8, wherein the database is associated with either a direct attached storage or a network attached storage.

Claim 26
The method of claim 15, wherein the database is associated with a direct attached storage and a network attached storage.
Claim 27
The method of claim 15, wherein the database is associated with either a direct attached storage or a network attached storage.
Claim 28
The non-transitory computer-readable media of claim 1, wherein the processor further executes computer-readable instructions to receive a sixth input of a database description for the database.
Claim 29
The non-transitory computer-readable media of claim 1, wherein each of the plurality of levels of the SLA defines one or more of a first duration for a continuous protection of the database, a second duration of protecting the database using a daily snapshot, a third duration of protecting the database using a weekly snapshot, a fourth duration of protecting the database using a monthly snapshot, and a fifth duration of protecting the database using a quarterly snapshot.
Claim 30
The non-transitory computer-readable media of claim 29, wherein the at least one time period comprises one or more of a time of day for capturing the daily snapshot, a day of a week for capturing the weekly snapshot, a day of a month for capturing a monthly snapshot, or a group of months for capturing a quarterly snapshot.
Claim 31
The system of claim 8, wherein the processor further executes computer-readable instructions to receive a sixth input of a database description for the database.
Claim 32
The system of claim 8, wherein each of the plurality of levels of the SLA defines one or more of a first duration for a continuous protection of the database, a second duration of protecting the database using a daily snapshot, a third duration of protecting the database using a weekly snapshot, a fourth duration of protecting the database using a monthly snapshot, and a fifth duration of protecting the database using a quarterly snapshot.
Claim 33
The system of claim 32, wherein the at least one time period comprises one or more of a time of day for capturing the daily snapshot, a day of a week for capturing the weekly snapshot, a day of a month for capturing a monthly snapshot, or a group of months for capturing a quarterly snapshot.

Claim 34
The method of claim 15, further comprising receiving, by the processor, a sixth input of a database description for the database.
Claim 35
The method of claim 15, wherein each of the plurality of levels of the SLA defines one or more of a first duration for a continuous protection of the database, a second duration of protecting the database using a daily snapshot, a third duration of protecting the database using a weekly snapshot, a fourth duration of protecting the database using a monthly snapshot, and a fifth duration of protecting the database using a quarterly snapshot.
Claim 36
The method of claim 35, wherein the at least one time period comprises one or more of a time of day for capturing the daily snapshot, a day of a week for capturing the weekly snapshot, a day of a month for capturing a monthly snapshot, or a group of months for capturing a quarterly snapshot.

CERTIFICATE OF SERVICE UNDER 37 C.F.R. § 42.6 (E)(4)

I certify that on December 13, 2024, I will cause a copy of the foregoing document, including any exhibits or appendices filed therewith, to be served via Priority Mail Express at the following correspondence address of record for the patent:

FOLEY & LARDNER LLP
3000 K STREET N.W.
SUITE 600
WASHINGTON, DC 20007-5109

Date: December 13, 2024

/MacAulay Rush/
MacAulay Rush
Paralegal
WOLF, GREENFIELD & SACKS, P.C.

CERTIFICATE OF WORD COUNT

Pursuant to 37 C.F.R. § 42.24, the undersigned certifies that the foregoing Petition for *Inter Partes* Review contains 13,999 words excluding a table of contents, a table of authorities, Mandatory Notices under § 42.8, a certificate of service or word count, or appendix of exhibits or claim listing. Petitioner has relied on the word count feature of the word processing system used to create this paper in making this certification.

Date: December 13, 2024

/MacAulay Rush/
MacAulay Rush
Paralegal
WOLF, GREENFIELD & SACKS, P.C.