UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/445,343	07/29/2014	Daniel Oberle	2058.966US1	6873

50400   7590   05/09/2018
SCHWEGMAN LUNDBERG & WOESSNER/SAP
P.O. BOX 2938
MINNEAPOLIS, MN 55402

EXAMINER
CASANOVA, JORGE A

ART UNIT	PAPER NUMBER
2165	

NOTIFICATION DATE	DELIVERY MODE
05/09/2018	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

uspto@slwip.com
SLW@blackhillsip.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte DANIEL OBERLE

Appeal 2017-011163
Application 14/445,343[1]
Technology Center 2100

Before TERRENCE W. McMILLIN, KARA L. SZPONDOWSKI, and
SCOTT B. HOWARD, *Administrative Patent Judges.*

McMILLIN, *Administrative Patent Judge.*

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) of the final
rejection of claims 1–20. We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

---

[1] According to Appellant, the real party in interest is SAP SE. App. Br. 2.

THE CLAIMED INVENTION

The present invention relates generally to "systems and methods for use with the storage of triples," and more particularly to "methods and systems for an in-memory cloud triple store." Spec. ¶ 1. Independent claim 1 is directed to a method; independent claim 8 is directed to a system; and independent claim 14 is directed to a non-transitory machine-readable medium. App. Br. 17–19.

Claim 1, reproduced below, is representative of the claimed subject matter:

> 1.   A method comprising:
>
>   receiving, from a first application, a triple to store in a triple store, the triple defining data having a subject, predicate, and object;
>
>   sending the triple to a triple store interface configured to persist the triple in an in-memory database management system, the in-memory database management system comprising an in-memory database where data is persisted;
>
>   receiving a request from a second application to query the triple store;
>
>   utilizing a cloud-based triple store query server to process the request and generate an in-memory database management system query to the in-memory database management system;
>
>   receiving results of the in-memory database management system query from the in-memory database management system; and
>
>   sending the result to the second application for display to a user.

REJECTIONS ON APPEAL

Claims 1–20 stand rejected under 35 U.S.C. § 101 because the claimed invention is directed to a judicial exception without significantly more. Ans. 2.

Claims 1, 8, and 14 stand rejected under 35 U.S.C. § 103 as being unpatentable over Stillerman et al. (US 2014/0108414 A1, published Apr. 17, 2014) ("Stillerman") and Santry (US 2015/0370544 A1, published Dec. 24, 2015), or alternatively over Stillerman and Saito et al. (US 2014/0101177 A1, published Apr. 10, 2014) ("Saito"). Final Act. 3.

Claim 2 stands rejected under 35 U.S.C. § 103 as being unpatentable over Stillerman, Santry, and Bowers et al. (US 2010/0114885 A1, published May 6, 2010) ("Bowers"), or alternatively over Stillerman, Saito, and Bowers. Final Act. 6.

Claims 3, 13, 15, and 16 stand rejected under 35 U.S.C. § 103 as being unpatentable over Stillerman, Santry, Aasman (US 2008/0243770 A1, published Oct. 2, 2008), or alternatively over Stillerman, Saito, and Aasman. Final Act. 7.

Claims 5, 7, 9, 10, 12, and 20 stand rejected under 35 U.S.C. § 103 as being unpatentable over Stillerman, Santry, and Rhoads et al. (US 2012/0134548 A1, published May 31, 2012) ("Rhoads"), or alternatively over Stillerman, Saito, and Rhoads. Final Act. 8.

Claim 18 stands rejected under 35 U.S.C. § 103 as being unpatentable over Stillerman, Santry, Aasman, and Rhoads, or alternatively over Stillerman, Saito, Aasman, and Rhoads. Final Act. 11.

Claim 4 stands rejected under 35 U.S.C. § 103 as being unpatentable over Stillerman, Santry, Aasman, and Boothroyd (US 2010/0318558 A1,

published Dec. 16, 2010), or alternatively over Stillerman, Saito, Aasman, and Boothroyd. Final Act. 12.

Claims 11 and 17 stand rejected under 35 U.S.C. § 103 as being unpatentable over Stillerman, Santry, and Boothroyd, or alternatively over Stillerman, Saito, and Boothroyd. Final Act. 13.

Claims 6 and 19 stand rejected under 35 U.S.C. § 103 as being unpatentable over Stillerman, Santry, and Betz et al. (US 2007/0185897 A1, published Aug. 9, 2007) ("Betz"), or alternatively over Stillerman, Saito, and Betz. Final Act. 14.

ANALYSIS
35 U.S.C. § 101 Rejections

Alice Corp. Pty. Ltd. v. CLS Bank Int'l, 134 S. Ct. 2347 (2014), identifies a two-step framework for determining whether claimed subject matter is judicially excepted from patent eligibility under 35 U.S.C. § 101. In the first step, "[w]e must first determine whether the claims at issue are directed to a patent-ineligible concept." *Alice*, 134 S. Ct. at 2355. A determination, under the first step of the *Alice* analysis, that the claims are *not* directed to a patent-ineligible concept requires no further inquiry as to patent eligibility.

The Examiner concludes the claims are directed to the abstract idea of "creating an index and using that index to search for and retrieve data," and specifically that "a triple store is being created (i.e., index) and being used by a second application to query and retrieve information (i.e., using the index to search for and retrieve data)." Ans. 2 (citing *Intellectual Ventures I LLC v. Erie Indemnity Company*, 850 F.3d 1315 (Fed. Cir. 2017)).

4

Appellant argues that the Examiner's interpretation of the claims as being directed to an abstract idea of creating an index and using that index to search for and retrieve data is an oversimplification of the claims. Reply Br. 8. As argued by Appellant, the claims do not "even mention the creation of an index or the use of such an index or searching and retrieving data" nor do the claims mention the word "index." *Id.*

We are persuaded by Appellant's arguments. We find that the Examiner's rejection, that the claims are directed to the abstract idea of creating an index and using that index to search for and retrieve data, oversimplifies the claims and is untethered from the actual language of the claims. *See Enfish, LLC v. Microsoft Corp.*, 822 F.3d 1327, 1337–1338 (Fed. Cir. 2016) ("describing the claims at such a high level of abstraction and untethered from the language of the claims all but ensures that the exceptions to § 101 swallow the rule.").

Accordingly, we do not sustain the Examiner's 35 U.S.C. § 101 rejection of claims 1–20.

35 U.S.C. § 103(a) Rejections

Claim 1 recites "receiving, from a first application, a triple to store in a triple store, the triple defining data having a subject, predicate, and object."

The Examiner finds that Stillerman's loading of index entries into the in-memory database teaches the receipt of the Resource Description Framework (RDF) data from another source. Ans. 5 (citing Stillerman ¶¶ 8, 21, 41); *see* Final Act. 3 (citing Stillerman ¶¶ 2, 37).

Appellant contends Stillerman teaches RDF database system storing RDF data, but does not teach that the RDF data is received, and thereby does

not teach "*receiving, from a first application*, a triple to store in a triple store." App. Br. 8–10 (emphasis added).

We are not persuaded by Appellant's arguments. As cited by the Examiner (Ans. 5), Stillerman teaches "script engine 18 receives a script 20 by an external interface and executes script 20 . . . Script engine 18 retrieves chunks 6A, 6B accessible by values of index entries 8B, 8K and *loads chunks 6A, 6B to in-memory RDF data 16*." Stillerman ¶ 41 (emphasis added). In other words, Stillerman teaches loading RDF data to an in-memory RDF database, and thereby teaches the in-memory RDF database receiving RDF data to store in a triple store.

Claim 1 further recites "sending the triple to a triple store interface configured to persist the triple in an in-memory database management system."

Appellant contends that Stillerman teaches retrieval of chunks from the chunk storage system 42 via the chunk storage API 40 for the script engine to retrieve and execute a script, but does not teach that the same triple is sent back over to the chunk storage API to be persisted. Reply Br. 2. More specifically, Appellant argues Stillerman teaches retrieving a piece of data from a persistent storage but not persisting a piece of data in a persistent storage. *Id.*

Appellant's argument against Stillerman separately from Santry does not persuasively rebut the combination made by the Examiner. One cannot show non-obviousness by attacking references individually, where the rejections are based on combinations of references. *In re Merck & Co.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986); *In re Keller*, 642 F.2d 413, 425 (CCPA 1981).

Specifically, we agree with the Examiner that Santry teaches "Stillerman can be modified to include an in-memory database, for the predictable result of retaining (e.g., persist) data after an event such as a restart or a loss of power." Ans. 5. As cited by the Examiner (Final Act. 3–5), Stillerman teaches retrieving "RDF chunks into in-memory RDF data 16" (Stillerman ¶ 42) and Santry teaches "facilitating persistent storage of in-memory databases" (Santry ¶ 8). In other words, Stillerman and Santry, *in combination*, teach the claimed sending a triple to a triple store interface to an in-memory database (Stillerman's sending data chunks to chunk storage API to then go to an in-memory database), where the triple is persisted in the in-memory database management system (Santry's persistent storage of in-memory databases).

Appellant further argues it would be improper to combine Stillerman and Santry because Stillerman already has persistent storage where data is persisted and is safe from restart or loss of power. Reply Br. 3. We agree with the Examiner's findings that it would be obvious to modify Stillerman's RDF data storage and querying with Santry's persistent in-memory database storage "for [the] predictable result of retaining data after a restart or a loss of power eliminating the need to re-transfer the data to the in-memory database that facilitates faster retrieval." Final Act. 5; *see* Ans. 5–6. Appellant has not presented persuasive arguments to address the Examiner's findings.

Furthermore, we note the pending alternative rejection for claim 1 over Stillerman and Saito. Final Act. 3; *see* Final Act. 5 ("In the alternative, Saito also teaches this concept: see claim 7 – 'wherein the in-memory database persists data . . . .'"). Appellant has not addressed this alternative

rejection of claim 1 over Stillerman and Saito. As such, we summarily affirm the alternative rejection over Stillerman and Saito.

Claim 1 further recites "utilizing a cloud-based triple store query server to process the request and generate an in-memory database management system query to the in-memory database management system."

The Examiner finds Stillerman teaches "utilizing cloud based technologies, additionally, that it can also receive additional requests from other systems." Ans. 7–8 (citing Stillerman ¶¶ 35, 43, 66); *see* Final Act. 4 (citing Stillerman ¶¶ 26, 35, 65, 82, Fig. 2).

Appellant contends Stillerman's system using cloud based technology and receiving requests from other systems does not teach the claimed cloud-based triple store query server. Reply Br. 3–4; *see* App. Br. 14.

We are not persuaded by Appellant's arguments. As cited by the Examiner (Final Act. 4), Stillerman teaches "RDF database system 12 may provide for *cloud computing* and distributed computing. RDF database system 12 may provide for *remote invocation via query API 36*." Stillerman ¶ 65 (emphasis added). In other words, Stillerman teaches the RDF database system provides for cloud computing and remote invocation via query API (claimed query server), and thereby Stillerman's query API is cloud-based by association.

Claim 1 further recites "sending the result *to the second application*."

The Examiner finds Stillerman's parser/evaluator returning script results to the requestor teaches sending the result to the second application for display to a user. Final Act. 5 (citing Stillerman ¶ 82).

Appellant contends that Stillerman's returning query results to the script engine does not teach "returning its query results to the same application as the original query was received from." App. Br. 15.

We are not persuaded by Appellant's arguments. Specifically, we agree with the Examiner's finding that Stillerman teaches that the requestor that sent the script to the query API is the same application that gets the results. Ans. 8 (citing Stillerman ¶ 82). As cited by the Examiner (Final Act. 5; Ans. 8), Stillerman teaches "*query API* 36 receives MQL script 45 *from a requestor* (100)" and "MQL parser/evaluator 34 *returns* MQL script *result* 47 *to the requestor* (116)." Stillerman ¶ 82 (emphases added).

Accordingly, we sustain the 35 U.S.C. § 103 rejection of independent claim 1, as well as the rejections of independent claims 8 and 14 which have limitations commensurate in scope with claim 1, and dependent claims 2–7, 9–13, and 15–20, not separately argued.

DECISION

The Examiner's rejection of claims 1–20 under 35 U.S.C. § 101 is reversed.

The Examiner's rejections of claims 1–20 under 35 U.S.C. § 103(a) are affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED