UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
15/250,871	08/29/2016	Dong Wang	C940	1082

152593 7590 09/23/2020

FOUNTAINHEAD/VMWARE-NICIRA
900 LAFAYETTE STREET
SUITE 301
SANTA CLARA, CA 95050

EXAMINER	
UNG, LANNY N	

ART UNIT	PAPER NUMBER
2191	

NOTIFICATION DATE	DELIVERY MODE
09/23/2020	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):

docketing@fountainheadlaw.com
ipadmin@vmware.com
klhussain@fountainheadlaw.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte DONG WANG, YUNFEI SAN, PENG GUO,
and
FELIX YAN

Appeal 2019-003400
Application 15/250,871[1]
Technology Center 2100

Before JOSEPH L. DIXON, DAVID M. KOHUT, and
JON M. JURGOVAN, *Administrative Patent Judges*.

JURGOVAN, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellant seeks review under 35 U.S.C. § 134(a) from a Final Rejection of claims 1–20, which are all the claims pending in the application. We have jurisdiction under 35 U.S.C. § 6(b).

We REVERSE.[2]

---

[1] We use the word "Appellant" to refer to "applicant(s)" as defined in 37 C.F.R. § 1.42. The real party in interest is VMWARE, INC. (Appeal Br. 2.)

[2] Our Decision refers to the Specification ("Spec.") filed August 29, 2016, the Final Office Action ("Final Act.") mailed May 10, 2018, the Appeal Brief ("Appeal Br.") filed November 7, 2018, the Examiner's Answer ("Ans.") mailed January 28, 2019, and the Reply Brief ("Reply Br.") filed March 26, 2019.

CLAIMED INVENTION

The claims are directed to methods and systems for configuring a container hosted application by deploying a container engine and a configuration agent to execute on a machine, deploying a container to execute on the container engine, and using the container to request and receive from the configuration agent "configuration information . . . relating to the user's configuration of the application." (Spec. ¶ 4, Title; Abstract.)

Claims 1, 10, and 17 are independent. Claim 1, reproduced below, is illustrative of the claimed subject matter:

> 1. A method for configuring an application with user configuration information executing in a container, the method comprising:
>
> deploying a container engine to execute on a first machine;
>
> deploying a configuration agent to execute on the first machine;
>
> deploying one or more containers to execute on the container engine, wherein each container performs operations including:
>
> the container instantiating at least one corresponding application to execute in the container;
>
> the container sending to the configuration agent a configuration request for configuration information relating to the application, the configuration request including information that identifies the container, information that identifies a user associated with the container, and information that identifies the application;
>
> the container receiving from the configuration agent configuration information relating to the user's configuration of the application; and
>
> the container configuring the application using the received configuration information.

(Appeal Br. 17–21 (Claims App.).)

REJECTIONS & REFERENCES

(1) Claims 1–8 and 10–20 stand rejected under 35 U.S.C. § 103 based on Bjorkengren (US 2016/0378525 A1, published Dec. 29, 2016) ("Bjorkengren"), Chhaparia (US 2016/0026442 A1, published Jan. 28, 2016) ("Chhaparia"), and Alevoor et al. (US 9,069,607 B1, issued June 30, 2015) ("Alevoor"). (Final Act. 2–10.)

(2) Claim 9 stands rejected under 35 U.S.C. § 103 based on Bjorkengren, Chhaparia, Alevoor, and Shaik et al. (US 2015/0309828 A1, published Oct. 29, 2015) ("Shaik"). (Final Act. 10–11.)

ANALYSIS

With respect to claim 1, the Examiner finds that: Bjorkengren's Docker daemon (running scripts on destination computing device 114) teaches the claimed "container engine" deployed to execute on a first machine (the destination computing device 114); and Bjorkengren's container configuration module 522 (which uses an automated application deployment tool for configuring a container 120 on destination computing device 114) teaches the claimed "configuration agent" deployed to execute on the first machine. (Final Act. 2–3 (citing Bjorkengren ¶¶ 36, 50); Ans. 14.) The Examiner further finds:

> [Bjorkengren's] computing system[] performs operations including: . . .
>
> sending to the configuration agent a configuration request for configuration information relating to the application, (*in response to an indication (request) that the container has been created and is awaiting checkpoint data, Paragraph 49, lines 1–7*)

> receiving from the configuration agent configuration information relating to the configuration of the application; (*receiving application checkpoint data from an application checkpoint database, Paragraph 50, lines 9–14*),

as recited in claim 1. (Final Act. 3.) Thus, Examiner finds the claimed "configuration request" (for configuration information relating to the application) is taught by Bjorkengren's "*indication*" transmitted by 0destination computing device 114 to source computing device 102. (*See id*. (citing Bjorkengren ¶ 49).) In the Answer, the Examiner finds the claimed "configuration request" is additionally taught by: Bjorkengren's "*configuration initialization request* . . . [that] is for gathering of configuration information related to the application that is being migrated/restored"; and by Bjorkengren's "*continuous monitoring* (requesting) of when the checkpoint data has been received at the migration management module/container configuration module (configuration agent)." (Ans. 14 (citing Bjorkengren ¶¶ 47, 49, 53) (emphases added), 16.)

The Examiner acknowledges "Bjorkengren does not explicitly disclose: wherein each container performs the operations" recited in claim 1, but asserts Chhaparia teaches "that a container performs similar functions of the 'sending', 'receiving' and 'configuring.'" (Final Act. 3–4 (citing Chhaparia ¶¶ 25–27); Ans. 20.) The Examiner reasons "[o]ne skilled in the art would be motivated to combine Chhaparia into Bjorkengren in order to have a container that automatically synchronizes itself when needed which could require less user intervention." (Ans. 21.) We do not agree.

We agree with Appellant that Bjorkengren and Chhaparia, alone or in combination, fail to teach or suggest a "container sending to the configuration agent a configuration request for configuration information

relating to the application," with "the container receiving from the configuration agent configuration information relating to the user's configuration of the application," as recited in claim 1. (Appeal Br. 7–13; Reply Br. 2–7.) As Appellant explains, none of Bjorkengren's *indication*, *configuration initialization request*, and *monitoring* teach the claimed "configuration request for configuration information relating to the application" sent from an entity (e.g., container) executing on a first machine, to a configuration agent on the same (first) machine, as required by claim 1. (*See* Appeal Br. 8–11; Reply Br. 3–6.) For example, Bjorkengren's *configuration initialization request* is not a "configuration *request for configuration information relating to the application* [that is to execute in the container]" as claimed, but is merely a request to configure a container on destination computing device 114 (the first machine). (Reply Br. 3, 6 (citing Bjorkengren ¶ 47).) Bjorkengren's *configuration initialization request* is also not exchanged between entities *on a first machine* (as is the case for claim 1's configuration request), rather, Bjorkengren's *initialization request* is transmitted from one machine (source computing device 102) to another machine (destination computing device 114/first machine). (*See* Bjorkengren ¶¶ 41 ("the source computing device **102** transmits a configuration initialization request to the destination computing device **114**"), 47.)

Bjorkengren's *indication* (*see* ¶ 49) and Bjorkengren's *monitoring* (*see id*.) do not teach the claimed "configuration request," either. (Appeal Br. 9–11; Reply Br. 4.) Bjorkengren's *indication* is merely a notification transmitted by destination computing device 114 (first machine) to another machine (source computing device 102) to indicate that a container has been

configured. (*See* Appeal Br. 10–11; Bjorkengren ¶¶ 49, 53.) Bjorkengren's *monitoring* is a recurring check performed by destination computing device 114 (first device) to "determine[] whether application checkpoint data (e.g., the application checkpoint data **108**) was received from the source computing device **102**." (*See* Bjorkengren ¶¶ 45, 49.) Bjorkengren does not describe the destination computing device 114 as *sending a request* for application checkpoint data; instead, destination computing device 114 waits in a loop (step 812 in Figure 8) for a transmission (of application checkpoint data) from source computing device 102. (Appeal Br. 9–10; Reply Br. 4.)

Thus, Bjorkengren does not teach or suggest an entity (executing on a first machine) sending to a configuration agent (on the first machine) a configuration request for configuration information relating to an application, and responsive to the configuration request, receiving from the configuration agent configuration information relating to the user's configuration of the application, as recited in claim 1. (Appeal Br. 7–12.)

Chhaparia does not make up for the above-noted deficiencies of Bjorkengren. (Appeal Br. 12–13; Reply Br. 5–6.) Although Chhaparia describes a software container that fetches and updates *dynamic application files from a cloud server* and periodically synchronizes dynamic applications *across a plurality of user devices*, Chhaparia does not teach or suggest that its container sends to a configuration agent (on the same machine as the container) a configuration request, and in response, receives from the configuration agent the configuration information as claimed. (*See id*.; Chhaparia ¶¶ 18, 25–27.) In addition, the Examiner has not provided an adequate reason based on rational underpinnings to explain why a skilled artisan would use Chhaparia's container in Bjorkengren *for performing the*

claimed "*container sending*" and "*container receiving*" steps. (Appeal Br. 12–13; Reply Br. 5–7.) Chhaparia does not teach that its container performs the claimed "sending" and "receiving"; Bjorkengren's container does not perform the claimed "sending" and "receiving," either. (*See id*.) Instead, Bjorkengren's *container is* (i) *passively configured* (based on a configuration initialization request sent by another machine/source computing device 102 to the first machine/destination computing device 114) and (ii) *is passively* (*without requesting*) *provided* with application checkpoint data (transmitted by another machine/source computing device 102). (*See* Bjorkengren ¶¶ 41, 45, 47, 49; Appeal Br. 8–9, 11–12; Reply Br. 6.) As Appellant further explains, Chhaparia's teaching of a container fetching files and periodically synchronizing data "cannot be incorporated into Bjorkengren to obtain the claim limitation, because Bjorkengren's [configuration initialization] request is a request to configure the very container that the Examiner asserts (via Chhaparia) is sending the request." (Reply Br. 7.) *See In re Chaganti*, 554 Fed. Appx. 917, 922 (Fed. Cir. 2014) ("It is not enough to say that . . . to do so would 'have been obvious to one of ordinary skill.' Such circular reasoning is not sufficient—more is needed to sustain an obviousness rejection.")

The Examiner also has not shown that the additional teachings of Alevoor and Shaik make up for the above-noted deficiencies of Bjorkengren and Chhaparia. Thus, for the reasons set forth above, we do not sustain the Examiner's obviousness rejection of independent claim 1 and claims 2–9 dependent therefrom. We also do not sustain the Examiner's obviousness rejection of independent claims 10 and 17, argued for substantially the same

reasons as claim 1, and claims 11–16 and 18–20 dependent therefrom. (Appeal Br. 14.)

CONCLUSION

The Examiner's rejection of claims 1–20 under 35 U.S.C. § 103 is REVERSED.

In summary:

Claims Rejected	35 U.S.C. §	Reference(s)/Basis	Affirmed	Reversed
1–8, 10–20	103	Bjorkengren, Chhaparia, Alevoor		1–8, 10–20
9	103	Bjorkengren, Chhaparia, Alevoor, Shaik		9
Overall Outcome				1–20

REVERSED