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.
15/393,620	12/29/2016	Filippo Privitera	CUEBIQ 3.0F-001	2562

530 7590 08/26/2020

LERNER, DAVID, LITTENBERG,
KRUMHOLZ & MENTLIK
20 COMMERCE DRIVE
CRANFORD, NJ 07016

EXAMINER
MANOHARAN, MUTHUSWAMY GANAPATHY

ART UNIT	PAPER NUMBER
2647	

NOTIFICATION DATE	DELIVERY MODE
08/26/2020	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):

eOfficeAction@lernerdavid.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte FILIPPO PRIVITERA, CARMELO VECCHIO, GIULIO BIDER, ANTONIO TOMARCHIO, WALTER FERRARA, and WILLIAM NESPOLI

Appeal 2019-006018
Application 15/393,620
Technology Center 2600

Before JOHN A. EVANS, JAMES W. DEJMEK, and RUSSELL E. CASS, *Administrative Patent Judges*.

DEJMEK, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellant[1] appeals under 35 U.S.C. § 134(a) from a Final Rejection of claims 1–25. Oral arguments were heard on August 11, 2020. A transcript of the hearing will be placed in the record in due course. We have jurisdiction over the pending claims under 35 U.S.C. § 6(b).

We reverse.

---

[1] Throughout this Decision, we use the word "Appellant" to refer to "applicant" as defined in 37 C.F.R. § 1.42 (2018). Appellant identifies Cuebiq S.R.L. as the real party in interest. Appeal Br. 1.

STATEMENT OF THE CASE

Introduction

Appellant's disclosed and claimed invention generally relates to "increas[ing] the amount of data that the operating system of a mobile device allows to be collected by an app[lication] operating in the background of the mobile device." Spec. ¶ 4. In a disclosed embodiment, a software component separate from the operating system dynamically generates a set of one or more geofence borders about the mobile device, and when the mobile device crosses the generated geofence border, instructs the operating system to temporarily activate the application (i.e., software component) to collect and store location data of the mobile device. Spec. ¶¶ 4–5. In addition, after the mobile device crosses the generated geofence border a new geofence border is generated based on the current location of the mobile device. Spec. ¶ 5. According to the Specification, the claimed approach "restores at least some control over data collection to the background application," as opposed to the operating system. Spec. ¶ 24.

Claims 1 and 20 are exemplary of the subject matter on appeal and are reproduced below with the disputed limitations emphasized in *italics*:

> 1. A method performed on a mobile device, wherein the mobile device includes an operating system configured to track a current location of the mobile device based on location data collected from hardware of the mobile device, the operating system being further configured to control permission for an application program included in the mobile device to access, collect or store data based on a set of one or more custom virtual borders, and wherein the mobile device further includes *a software component external from the operating system* and in communication with the operating system for providing the set of one or more custom virtual borders, *wherein the method is performed by the software component* and comprises:

the software component receiving the location data indicating the current location of the mobile device;

the software component generating the set of one or more custom virtual borders around the current location based on the received location data;

the software component providing the generated set of custom virtual borders to the operating system, thereby causing the operating system to permit the application program to access, collect or store data only when the mobile device crosses one or more of the custom virtual borders;

upon the mobile device crossing the one or more custom virtual borders, the software component receiving new location data indicating a new current location of the mobile device;

the software component generating a new set of custom virtual borders around the new current location based on the new location data; and

the software component providing the new set of custom virtual borders to the operating system, thereby causing the operating system to permit the application program to access, collect or store data only when the mobile device crosses one or more of the new custom virtual borders.

20. A mobile device comprising:

a receiver for receiving location data indicating a current location of the mobile device;

a processor for controlling operations of the mobile device; and

a non-transitory computer-readable storage medium having encoded thereon:

an operating system configured to cause the processor to control operations of the mobile device;

a software component external from the operating system; and

an application program interface for enabling communication between the operating system and the software component,

> *wherein the software component comprises instructions configured to cause the processor to:*
>
> > *dynamically generate a set of one or more virtual borders around a current location of the mobile device; and*
> >
> > *transmit the generated set of custom virtual borders to the operating system*;
>
> wherein the operating system is programmed to, detect the mobile device crossing one or more of the dynamically generated virtual borders based on the received location data; and wherein the operating system is further programmed to temporarily enable an application programmed on the mobile device to collect and store data upon detecting the mobile device crossing one or more of the dynamically generated virtual borders,
>
> *and wherein the software component comprises instructions further configured to cause the processor to dynamically generate an updated set of virtual borders around the detected location of the mobile device* upon the operating system detecting the mobile device crossing one or more of the dynamically generated virtual borders.

The Examiner's Rejections[2]

1. Claims 1, 3–5, 8, 10–13, 15–18, 22, 23, and 25 stand rejected under 35 U.S.C. § 103 as being unpatentable over Case et al. (US

[2] Appellant asserts the Examiner erred in designating the Office Action mailed on August 16, 2018 as "Final." Appeal Br. 22–23. That is a petitionable matter not properly before the Board. *See* 37 C.F.R. § 1.181(a)(1); *see also* Manual of Patent Examining Procedure (MPEP) § 706.01 (9th ed. Rev. 08.2017, Jan. 2018) ("[T]he Board will not hear or decide issues pertaining to objections and formal matters which are not properly before the Board."); *see also* MPEP § 1201 ("Some matters which have been determined to be petitionable and not appealable include: a requirement for restriction or election of species, finality, non-entry of amendments, and holdings of abandonment.").

2013/0267253 A1; Oct. 10, 2013) ("Case") and Broscoe et al. (WO 2012/000107 A1; Jan. 5, 2012) ("Broscoe"). Final Act. 2–7.

2. Claims 2, 14, and 19 stand rejected under 35 U.S.C. § 103 as being unpatentable over Case, Broscoe, and Sipher et al. (US 9,445,230 B1; Sept. 13, 2016) ("Sipher"). Final Act. 7–8.

3. Claim 9 stands rejected under 35 U.S.C. § 103 as being unpatentable over Case, Broscoe, and Huang et al. (US 2011/0256881 A1; Oct. 20, 2011) ("Huang"). Final Act. 8–9.

4. Claim 6 stands rejected under 35 U.S.C. § 103 as being unpatentable over Case, Broscoe, and Fraccaroli (US 2014/0074874 A1; Mar. 13, 2014). Final Act. 9.

5. Claim 7 stands rejected under 35 U.S.C. § 103 as being unpatentable over Case, Broscoe, and Feng et al. (US 10,038,972 B1; July 31, 2018) ("Feng"). Final Act. 10.

6. Claims 20 and 21 stand rejected under 35 U.S.C. § 103 as being unpatentable over Sipher and MacDonald et al. (US 2016/0286345 A1; Sept. 29, 2016) ("MacDonald"). Final Act. 10–13.

7. Claim 24 stands rejected under 35 U.S.C. § 103 as being unpatentable over Case, Broscoe, and MacDonald. Final Act. 13.

ANALYSIS[3]

Claims 20 and 21

In rejecting claims 20 and 21, the Examiner relies on the combined teachings of Sipher and MacDonald. Final Act. 10–13. In relevant part, the

[3] Throughout this Decision, we have considered the Appeal Brief, filed January 31, 2019 ("Appeal Br."); the Reply Brief, filed July 30, 2019

Examiner relies on MacDonald to teach, *inter alia*, a software component external from the operating system that causes a processor to dynamically generate a set of custom virtual boarders and allow an application to collect and store data upon detection of the mobile device crossing the custom virtual border. Final Act. 11–12 (citing MacDonald ¶¶ 38–39, 68, Fig. 4); *see also* Ans. 3–6 (citing MacDonald ¶¶ 36, 38–39, 73, Fig. 2).

Appellant asserts that rather than a software component external from the operating system dynamically generating custom borders around the present location of the mobile device, MacDonald describes a software component making calls to a web server to download regions of interest (i.e., what the Examiner maps to the claimed custom borders) to the mobile device. Appeal Br. 10–14 (citing MacDonald ¶¶ 41, 60, Fig. 8); Reply Br. 3–4 (citing MacDonald ¶¶ 36, 39, 73). In other words, Appellant argues that, contrary to the language of claim 20, MacDonald's *web server* is responsible for performing the various map-related calculations instead of the software component in the mobile device. Appeal Br. 11–12.

MacDonald generally relates to "a mobile device monitoring predefined regions and detecting when the mobile device enters and exits those predefined regions." MacDonald ¶ 3. MacDonald describes that region monitoring is accomplished using a web service application programming interface (API). MacDonald ¶ 29. Further, MacDonald teaches a Software Development Kit (SDK) may be provided to allow an application on a mobile device to register with the web server and monitor

("Reply Br."); the Examiner's Answer, mailed June 3, 2019 ("Ans."); and the Final Office Action, mailed August 16, 2018 ("Final Act."), from which this Appeal is taken.

regions downloaded from the web server. *See* MacDonald ¶¶ 29–31. MacDonald teaches the radius of the region being monitored is received from the web server. MacDonald ¶ 32–33. The radius of the region being monitored (as well as the number of regions being monitored) "may be changed at any time through the web server user interface." MacDonald ¶ 33. In addition, as relied on by the Examiner, MacDonald teaches that the SDK or client-side API calls within the application may retrieve a set of regions to monitor. MacDonald ¶¶ 36–37. In addition, MacDonald teaches "[w]hen the mobile app containing the SDK or client-side API calls is downloaded and installed, the SDK or client-side API calls within the app registers with the web server, retrieves the closest N regions, and establishes a personal region around the mobile device." MacDonald ¶ 38.

In response to Appellant's arguments, the Examiner finds the SDK calls establish a defined region about the mobile device and that the SDK calls within the application check the current location of the mobile device to determine if the device is within the region. Ans. 5 (citing MacDonald ¶ 36). Additionally, the Examiner finds MacDonald teaches the SDK calls retrieve a new set of regions as the device moves beyond the initial set of regions. Ans. 5–6 (citing MacDonald ¶¶ 38–39). Moreover, the Examiner finds that MacDonald's web server could be part of the mobile device itself, thus, a software component separate from the operating system of the mobile device. Ans. 6 (citing MacDonald ¶ 73).

We are persuaded by Appellant's arguments. As described above, MacDonald teaches a web server stores the various regions of interest and downloads certain regions to a mobile device in response to a call from the SDK of the application running on the mobile device. MacDonald teaches

that the regions are predefined at the web server. The Examiner has not provided persuasive evidence or technical reasoning that the SDK generates the custom borders associated with the region of interest to be monitored. Further, we disagree that an ordinarily skilled artisan would consider the webserver to be part of the mobile device. As relied on by the Examiner, MacDonald merely teaches the mobile device may communicate "with a remote web server." *See* MacDonald ¶ 73. This disclosure does not reasonably suggest the web server may be a part of the mobile device.

Accordingly, on the record before us, we do not sustain the Examiner's rejection of claims 20 and 21.

Claims 1–19 and 22–25

In rejecting independent claims 1, 12, and 18, the Examiner relies on the combined teachings and suggestions of Case and Broscoe. Final Act. 2–7. In relevant part, the Examiner finds Case teaches a software component external from the operating system that receives the current location of the mobile device and generates a custom virtual border around the location. Final Act. 3 (citing Case ¶¶ 61, 102, Figs. 16, 17). Moreover, the Examiner finds Case teaches the software component providing the generated set of custom virtual borders to the operating system "thereby causing the operating system to permit the application to access, collect or store data only when the mobile device crosses one or more of the custom virtual borders." Final Act. 3 (citing Case, Fig. 17). In addition, the Examiner finds, in relevant part, that Broscoe teaches that when the mobile device crosses one or more of the generated set of custom virtual borders, the software component receives an updated location of the mobile device and

generates a new set of custom virtual borders based on the updated location. Final Act. 4 (citing Broscoe 7–8, Fig. 7).

Appellant asserts (similar to MacDonald), that Case's trigger zones (i.e., the claimed set of custom virtual borders) are either preloaded on the mobile device or downloaded from a remote server. Appeal Br. 15–17 (citing Case ¶¶ 58–74); Reply Br. 2–3 (citing Case ¶¶ 61, 102). Thus, Appellant argues the trigger zones described in Case are not custom virtual borders generated by the software component based on a received location of the mobile device, but are fixed locations stored at a remote server and selectively provided to the mobile device. Appeal Br. 18. In addition, Appellant argues Broscoe does not cure the alleged deficiencies of Case and asserts Broscoe "is concerned with defining a 'complex geofence' around locations that a mobile device has been recently detected, based on instances of time at which the device user enters a password to unlock the device." Appeal Br. 17 (citing Broscoe 8).

Also, during Oral Arguments, counsel explained the claims require the generated set of custom virtual borders are provided to the operating system to cause the operating system "to permit the application program to access, collect or store data *only* when the mobile device crosses one or more of the custom virtual borders." *See* claim 1 (emphasis added); *see also* Tr. 6–8, 17–19. In other words, the virtual border data does not supplement any default geofence data used by the operating system, but instead replaces it. Counsel further suggested this argument was made in the briefs. *See also* Reply Br. 2 (asserting the Examiner has not shown how the cited art teaches this aspect of the claims) (citing Appeal Br. 12, 17, 19–20); *see also* Tr. 17–19.

As an initial matter, we note that Appellant's reference to page 12 of the Appeal Brief relates to *MacDonald's* statement (i.e., not *Case*) that its disclosed approach attempts to obey the constraints of all mobile operating systems. *See* Appeal Br. 12 (citing MacDonald ¶ 60). On page 17 of the Appeal Brief (with regard to the Examiner's rejection of claim 1 over Case and Broscoe), Appellant merely states "the claims require the software component to repeatedly generate custom or dynamic virtual borders that are provided to the operating system to control when the operating system gives permission to an application program." Similarly on pages 19 and 20 of the Appeal Brief, Appellant summarizes a concern of the present application as "custom virtual borders (not fixed data) are generated at the mobile device and used to control *(e.g.,* limit) access and storage permissions of applications programed on the mobile device." These limited statements do not amount to an argument that Case (or Broscoe) fails to teach the operating system permitting the application program to access, collect, or store data only when the mobile device crosses one or more of the custom virtual borders.

Case generally relates to trigger zones and dwell time analytics. Case, Title. More specifically, Case describes triggering an event based on a current geographic position of a computing device relative to a time-dwell trigger zone. Case ¶¶ 4–5, 9. Case describes "receiving and processing trigger zone and geo-trigger content on a computing device." Case ¶ 41. In some embodiments, a server computer sends the trigger zone and trigger zone data to the computing device for processing. Case ¶ 42. Similarly, Case describes "trigger zone data can be pre-loaded on a computing device" or "a remote server [may be] configured to store trigger zone data and [be]

operable to wirelessly transfer the trigger zone data" to the computing device. Case ¶ 61. Further, Case describes the geo-fences may be "customizable." Case ¶ 44. In addition, Case also describes embodiments in which the trigger zone may be created locally and/or defined by the user. *See* Case ¶¶ 102 ("[i]n a further embodiment, a user can create geo-trigger content on computing device 510 and store it on computing device 510"), 136 (describing a method allowing a user to define a trigger zone).

Broscoe generally relates to the automatic creation and modification of dynamic geofences. Broscoe, Title. In particular, Broscoe teaches determining the location of the device to see if it is within a defined geofence. Broscoe 7. If the device is located outside the geofence (i.e., generated custom virtual borders), the user is prompted for a password "and the geofence is adjusted 28 to include the new location." Broscoe 7. Thus, as the location of the mobile device changes and exits a geofence, a new geofence based on the current location of the device is created if the user provides a correct password. Broscoe 7–8.

Based on our review of Case, we do not find Case's system to be limited only to a software component on a computing device receiving trigger zone data only from a remote server. Rather, Case describes geo-trigger information (i.e., trigger zone data) may be created by the user on the computing device. Thus, Case teaches, or at least reasonably suggests, a software component on a mobile device separate from the operating system generating a set of custom virtual borders.

However, it is not clear that Case teaches the generation of the custom virtual borders is based on received location data of the mobile device, as required by claim 1 (as well as independent claims 12 and 18). Although

Broscoe describes the generation of a geofence based on the location of the mobile device (as well as the generation of an updated geofence based on updated location information of the device), the Examiner has not made a finding or conclusion that one of ordinary skill in the art would have incorporated Broscoe's teaching of dynamically generating a geofence based on a received location of the device with Case's teaching of a software component generating a set of custom of virtual borders. *Cf.* Final Act. 4–5 (relying on Broscoe to teach the generation of a new set of custom virtual borders based on updated location information). Although the Board is authorized to reject claims under 37 C.F.R. § 41.50(b), no inference should be drawn when the Board elects not to do so. *See* Manual of Patent Examining Procedure (MPEP) § 1213.02 (9th ed. Rev. 10.2019, June 2020).

Accordingly, constrained by the record before us, we do not sustain the Examiner's rejection of claim 1. For similar reasons, we do not sustain the Examiner's rejection of independent claims 12 and 18, which recite commensurate limitations. In addition, we do not sustain the Examiner's rejections of claims 2–11, 13–17, 19, and 22–25, which depend directly or indirectly therefrom.

CONCLUSION

We reverse the Examiner's decision rejecting claims 1–25 under 35 U.S.C. § 103.

DECISION SUMMARY

Claims Rejected	35 U.S.C. §	Reference(s)/Basis	Affirmed	Reversed
1, 3–5, 8, 10–13, 15–18, 22, 23, 25	103	Case, Broscoe		1, 3–5, 8, 10–13, 15–18, 22, 23, 25
2, 14, 19	103	Case, Broscoe, Sipher		2, 14, 19
9	103	Case, Broscoe, Huang		9
6	103	Case, Broscoe, Fraccaroli		6
7	103	Case, Broscoe, Feng		7
20, 21	103	Sipher, MacDonald		20, 21
24	103	Case, Broscoe, MacDonald		24
Overall Outcome				1–25

REVERSED