

Internal distribution code:

- (A) [-] Publication in OJ
- (B) [-] To Chairmen and Members
- (C) [-] To Chairmen
- (D) [X] No distribution

**Datasheet for the decision
of 11 May 2022**

Case Number: T 0447/19 - 3.4.03

Application Number: 15180403.6

Publication Number: 2977941

IPC: G06Q10/06, G06Q50/04, G06F21/57

Language of the proceedings: EN

Title of invention:
SYSTEM AND METHOD FOR MANAGING ELECTRONIC ASSETS

Applicant:
BlackBerry Limited

Headword:

Relevant legal provisions:
EPC Art. 56, 123(2)

Keyword:
Amendments - added subject-matter (no)
Inventive step - (yes)

Decisions cited:

Catchword:



Beschwerdekammern
Boards of Appeal
Chambres de recours

Boards of Appeal of the
European Patent Office
Richard-Reitzner-Allee 8
85540 Haar
GERMANY
Tel. +49 (0)89 2399-0
Fax +49 (0)89 2399-4465

Case Number: T 0447/19 - 3.4.03

D E C I S I O N
of Technical Board of Appeal 3.4.03
of 11 May 2022

Appellant: BlackBerry Limited
(Applicant) 2200 University Avenue East
Waterloo, Ontario N2K 0A7 (CA)

Representative: Murgitroyd & Company
Murgitroyd House
165-169 Scotland Street
Glasgow G5 8PL (GB)

Decision under appeal: **Decision of the Examining Division of the
European Patent Office posted on 2 October 2018
refusing European patent application No.
15180403.6 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chair T. Häusser
Members: M. Papastefanou
E. Mille

Summary of Facts and Submissions

- I. The appeal is against the decision of the examining division refusing the European patent application No. 15 180 403 on the grounds that the subject-matter of claim 1 of the sole request before it extended beyond the application as filed (Article 123(2) EPC) and did not involve an inventive step (Article 56 EPC).
- II. Reference is made to the following documents:
D1: WO 2006/133545 A1,
D3: US 2008/0040550 A1.
- III. At the end of the oral proceedings before the board the request of the appellant-applicant was that the decision under appeal be set aside and that a patent be granted on the basis of the request underlying the impugned decision:
- Claims:
no. 1 to 5 filed with the letter dated 7 August 2018;
 - Description:
pages 1, 1A filed with the letter dated 27 July 2016,
pages 2, 14, 15 filed with the letter dated 27 February 2017,
pages 3 to 13 and 16 to 92 as originally filed;
 - Drawings:
sheets 1/73 to 5/73 and 7/73 to 73/73 as originally filed,
sheet 6/73 filed with the letter dated 27 February 2017.
- IV. Claims 1 and 5 of the sole request on file read as follows:

1. A method of controlling the distribution of electronic assets to a test application (116b) in a manufacturing process, said method comprising: providing a daemon application programming interface (API) (23) on an instance of said test application to provide assets upon detecting a request therefor, and to obtain log data from said instance of said test application during testing; initiating a daemon (25) in connection with said daemon API to obtain said log data from said daemon API (23) and to provide said assets to said daemon API, said daemon hosting an agent API (21) for communicating with an appliance (18) remote to said instance of said test application; utilizing said agent API to obtain a batch comprising a plurality of assets and to provide one or more log reports containing said log data separately from said instance of said test application; and said daemon: caching said assets to provide a quantity of said assets to said daemon API upon request therefrom to enable said daemon API to provide said assets to said instance of said test application thereby avoiding session establishment between said instance of said test application and said appliance for obtaining said electronic assets; and maintaining a record of assets not used when said instance of said test application terminates for a next instance of the test application.

5. A computer readable medium comprising computer executable instructions that when executed cause a computing device to perform the method of any one of claims 1 to 4.

- V. The appellant argued essentially that neither D1 nor D3 disclosed that the daemon used the log data to maintain a record of assets not used when the test application terminated. A combination of the teaching of these two documents would therefore not lead the skilled person to the claimed invention.

Reasons for the Decision

1. The claimed invention
- 1.1 The invention relates to a method for the distribution of electronic assets to a test application in a manufacturing process.

In distributed manufacturing, devices are manufactured at one or more manufacturing sites, often located remotely from the central producer. In such a situation there is a need for the central producer to monitor and control the devices produced remotely in order to avoid fraud by local manufacturing (sub)contractors.

- 1.2 A common practice is that in order to be functional each device produced has to be provided with an electronic asset, such as a cryptographic key, a serial number or a specific feature. These electronic assets are distributed from the central producer to the remote manufacturers upon request.

Electronic assets are thus transmitted from a controller of the central producer to a local server at each remote manufacturer ("appliance" in the terminology of the application). These assets are then distributed by the appliance to test applications (testers) which test the produced devices and also insert the assets into them (see Figure 1 of the

published application). Each test application generates log data recording the assets inserted into the tested devices. An agent, as part of each test application, manages the assets for the test application (see e.g. Figure 6A of the published application).

1.3 The invention relates to the management of the asset distribution and usage at the manufacturer's site. The agent is executed as a separate daemon process, independent from the corresponding test application. In this way the agent takes over the communication with the appliance and the request/receipt of electronic assets for the corresponding test application. The assets are cached at the daemon and transmitted to the test application as needed. At the same time, the agent at the daemon receives the asset insertion log data from the test application and, comparing them with the assets recorded in the daemon's cache, can monitor their usage. In case the instance of the test application terminates, the unused assets remaining at the daemon's cache can be provided to a subsequent instance of the test application.

1.4 With this implementation, the appliance does not need to establish connection with the test application every time the application needs assets for insertion into the device, as this task is taken over by the daemon. The appliance does not need to monitor the usage of the assets for each test application, either, as this monitoring is also taken over by the daemon. Moreover, assets left at the daemon's cache at the termination of the test application are not lost but can be used by a subsequent instance of the test application (see Figure 6B of the published application).

2. Added subject-matter (Article 123(2) EPC)

2.1 The examining division held that there was no basis in the originally filed application for the last feature of claim 1, according to which the daemon maintained a record of assets not used when said instance of said test application terminated for a next instance of the test application.

According to the examining division, there was a basis in the originally filed application for the daemon maintaining a record of assets not used when the test application terminated, and for the fact that such assets were maintained for the next instance of the test application. What was missing was that it was the daemon maintaining those assets for a next instance of the test application (see points 1.1 and 1.2 of the impugned decision).

2.2 According to the application, the daemon 25 is a separate process that *retrieves assets from and provides assets to an appliance 18, reducing some of the work being done by the tester application 116* (see the description as originally filed, paragraphs [104] and [105]). The daemon 25 *can request batches of assets at a time using the agent API 21, and deliver assets as they are needed to the tester (ibid., last lines on page 14)*.

Moreover, the daemon 25 *maintains an asset cache 130 to store batches of assets for subsequent distribution to the tester 17 as needed (ibid., first lines on page 15)*. Finally, the daemon 25 and the asset cache 130 *would survive the test application 116b and thus no assets would be wasted without a chance to recover them (ibid., paragraph [106])*.

As this paragraph further explains, leftover assets, i.e. those assets that remain in the daemon's asset cache 130 without being delivered to the test application (tester), may be credited back to the customer, if it is permitted, or they can be maintained for the next instance of the test application 116b (*ibid.*, last lines of paragraph [106]).

2.2.1 It is true that the sentences of the description cited above do not mention explicitly that it is the daemon which credits the assets back to the customer or maintains them for the next instance of the test application. The board's understanding however is that, since the assets are stored in the daemon's asset cache and it is the daemon that takes over the management of the assets (see also the first lines of paragraph 107), it must also be the daemon that maintains the assets for the next instance of the test application. The application does not provide any indication that this is done by any other process/element/part than the daemon.

2.2.2 The board is thus persuaded that the skilled person would directly and unambiguously derive from the above cited passages that the daemon maintains a record of assets not used when the instance of the test application terminates for a next instance of the test application, as claim 1 defines.

Claim 1, therefore, fulfils the requirements of Article 123(2) EPC.

3. Inventive step (Article 56 EPC)

3.1 It is common ground that document D1 represents the closest prior art.

According to the appellant, claim 1 differs from D1 in that (numbering by the board in line with the impugned decision):

- (a) the agent software program is executed as a daemon process separately from the test application and comprises an agent API for communicating with the appliance, and
- (b) the daemon maintains a record of assets not used when the instance of the test application terminates for a next instance of the test application.

These distinguishing features were also identified by the examining division, although the division interpreted feature (b) differently, in that it was not the daemon which maintained the assets for the next instance of the test application but simply "*assets not used ... are maintained ...*" (see the middle of page 6 of the impugned decision). Since the board agrees with the appellant's interpretation of feature (b), it also adopted the appellant's definition of the distinguishing features (see page 3 of the statement of the grounds of the appeal).

3.2 According to the appellant, these two distinguishing features combine to provide the technical effect of the daemon taking over the management of assets, reducing thus the load of the server. Hence, the objective technical problem the skilled person starting from D1 would be faced with was *how to minimise the overhead of the server 18 when dealing with equipment 20 (ibid., page 5)*.

3.2.1 As the appellant further explained during the oral proceedings, the daemon was monitoring the use of the cached assets by the test application. Receiving the log data from the test application regarding the assets inserted into the devices, the daemon was monitoring the use of the cached assets (see also Figure 6B of the published application). In D1 it was the controller of the producer which was receiving all the log data from the key agent and was monitoring which keys (assets) were inserted into the devices, whether there were any unused keys left, etc. (see e.g. Figures 1 and 4 of D1). Hence, by having this monitoring performed by the daemon, the overhead of the server was indeed minimised.

3.3 The board accepts the appellant's formulation of the objective technical problem.

In the decision under appeal, the examining division considered that the two identified distinguishing features did not combine to produce a synergistic technical effect and assessed them separately. However, since the board does not accept the division's interpretation of the second distinguishing feature, it does not follow its formulation of the technical problem, either. Since both distinguishing features (a) and (b) relate to the daemon, and the appellant's formulation of the objective technical problem is considered plausible, the board decided to follow the appellant in this respect.

3.4 The examining division referred also to D3. D3 describes a network in which a series of client applications 120 are connected to a directory server 110 (see Figure 1). Normally such client applications access the directory server by establishing a direct

connection through a binding operation, which initiates a protocol session between the application and the server, allows authentication of the client to the sever, etc. (see paragraph [0009]).

A problem in such a network architecture is that each application needs to establish a direct connection to the server before it can request any information from it. The server needs to establish separate connections (sessions) to each application for any exchange of information to take place. This increases the load on the server affecting its performance (see paragraph [0010]).

D3 solves this problem by installing a Light Directory Access Protocol (LDAP) caching daemon 210 between the applications and the server (see Figure 2). The daemon obtains data from the server and stores (caches) them in its data cache so that it can directly provide them to the requesting applications without any need for the applications to connect to the server. At the same time, the daemon connects to the directory server and retrieves any information requested by an application but not stored in its cache. In this way, the server has to manage only one individual connection (to the daemon) and can perform its main task of information retrieval more efficiently (see paragraphs [0023] to [0027]).

- 3.4.1 The appellant pointed out that in D3 there was no asset distribution. The daemon cached LDAP server information, which was accessed by the application(s). There were no log data regarding the use of this cached information by the applications nor any monitoring by the daemon of the use/access of the cached information

by the applications.

3.4.2 The board agrees with the appellant regarding this interpretation of D3. It notes, however, that the aspect of asset distribution and management is disclosed in D1. The examining division referred to D3 as an example of a "modular implementation" which would incite the skilled person to locate a part of the operations (processes) of the manufacturing line equipment (20) in D1, and specifically the key agent (21) at a separate daemon in the same way as in the claimed invention (see Figure 1 of D1).

3.5 Regarding the key agent (21) in D1, the board notes that it operates independently from the manufacturing equipment (20) (corresponding to the claimed test application). D1 states for example that the key agent requests and receives keys from the server (corresponding to the claimed appliance) based on predetermined threshold levels and thus independently from the insertion of the keys into the devices by the manufacturing equipment (see for example paragraphs [0043] and [0044]). Hence, there is no need for the manufacturing line equipment to connect constantly to the server and request keys. This is comparable to the function of the daemon of the application (see e.g. paragraphs [0104] and [0105] of the application as published).

In this respect, the implementation of the key agent (21) within a separate daemon process would not have changed anything since the key agent (21) already manages the communication with the server, the request/receipt of the keys (assets) and their distribution to the manufacturing equipment.

- 3.5.1 In the light of these considerations the board takes the view that the skilled person wishing to reduce the overhead of the server in D1 would not have been incited to locate the key agent in a separate daemon process as this would have not reduced the load to the server.
- 3.6 Regarding the log data, in D1 there is a first log ("key_to_server" log) generated at the producer when keys are transmitted to the server (18) of the manufacturer. A second log ("key_to_agent" log) is generated when keys are transmitted from the server (18) to the key agent (21) and a third log ("key_injection" log) is generated by the manufacturing equipment (20) as keys are inserted into the devices (see Figure 4). The "key_injection" log is transmitted to the server through the key agent, which concatenates it with the "key_to_agent" log to generate a "Log Report R". This report is transmitted to the controller of the producer which, comparing it with the "key_to_server" log, evaluates the use of the distributed keys (see also paragraphs [0056] to [0063]). There is no indication in D1 that the key agent (21) is in any way involved in the monitoring of the key distribution and usage, other than transmitting the "key_injection" log generated by the manufacturing equipment to the server.
- 3.7 The examining division considered that executing the key agent (21) as a separate background daemon process would have been an obvious choice for the skilled person which would be aware of the advantages of such a modular implementation based only on their common general knowledge. Moreover, such an a implementation would also have been obvious in view of the teaching of D3 (see first two paragraphs on page 7 of the impugned

decision).

- 3.7.1 As explained previously, the board does not agree with the examining division's separate assessment of the distinguishing features or its formulation of the objective technical problem.

Even if this argument were followed and it were to be accepted that moving the key agent (21) of the manufacturing equipment (see Figure 1 of D1) to a separate daemon process would have been obvious to the skilled person, the board notes that there would still be feature (b) missing from such an implementation to arrive at the claimed invention.

In the claimed invention it is the daemon which, using the log data received from the test application, maintains a record of the unused assets remaining in its cache. The key agent (21) in D1 does not do anything similar. As explained in point 3.6 above, the monitoring of the keys (assets) and their usage in D1 is performed by the controller of the producer and not by the key agent. Implementing the key agent in a separate daemon process would not change this. The log data would still be sent by the manufacturing equipment (20) to the server (18) through the key agent (21). The evaluation and monitoring of the key usage would still be done by the controller of the producer based on the received log report "R".

Since according to the method of D1 the evaluation and monitoring of key usage based on the log data are effected at the producer and not at the server of the manufacturer, the skilled person wishing to reduce the load to the server would not have any reason to move this functionality from the *producer* to the key agent.

They would rather seek functionalities that could be moved from the *server of the manufacturer* to the key agent.

Hence, there is no incentive in D1 for the skilled person to introduce any key usage monitoring functionality at the key agent without hindsight. D3, which does not mention any keys/assets or any monitoring of the usage/access of the data cached at the daemon by the applications would not provide any such incentive, either.

- 3.8 The board's conclusion is, therefore, that the subject-matter of claim 1 involves an inventive step within the meaning of Article 56 EPC.

Independent claim 5 relates to the corresponding computer readable medium. Claims 2 to 4 are dependent on claim 1.

Accordingly, the subject-matter of claims 1 to 5 of the sole request involves an inventive step (Article 52(1) and 56 EPC).

4. The description has been adapted to the claims and D1 is mentioned therein (see page 1A).
5. The board is thus convinced that the application and the invention to which it relates meet the requirements of the EPC and a European patent is to be granted according to Article 97(1) EPC.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.
2. The case is remitted to the department of first instance with an order to grant a patent based on the following documents:
 - Claims:
no. 1 to 5 filed on with the letter dated 7 August 2018;
 - Description:
pages 1, 1A filed with the letter dated 27 July 2016,
pages 2, 14, 15 filed with the letter dated 27 February 2017,
pages 3 to 13 and 16 to 92 as originally filed;
 - Drawings:
sheets 1/73 to 5/73 and 7/73 to 73/73 as originally filed
sheet 6/73 filed with the letter dated 27 February 2017.

The Registrar:

The Chair:



S. Sánchez Chiquero

T. Häusser

Decision electronically authenticated