BESCHWERDEKAMMERN DES EUROPÄISCHEN PATENTAMTS

BOARDS OF APPEAL OF THE EUROPEAN PATENT OFFICE

CHAMBRES DE RECOURS DE L'OFFICE EUROPÉEN DES BREVETS

Internal distribution code:

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

Datasheet for the decision of 20 December 2023

Case Number:	T 0238/21 - 3.5.07
Application Number:	13776643.2
Publication Number:	2901321
IPC:	G06F17/30
Language of the proceedings:	EN

Title of invention:
Managing continuous queries with archived relations

Applicant:
Oracle International Corporation

Headword:
Managing continuous queries/ORACLE

Relevant legal provisions:
EPC Art. 56
RPBA 2020 Art. 11

Keyword:
Inventive step - main request (no)
Remittal - special reasons for remittal

Decisions cited:
T 0939/92, T 0415/21

This datasheet is not part of the Decisior
It can be changed at any time and without notic

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 0238/21 - 3.5.07

DECISION
# of Technical Board of Appeal 3.5.07
of 20 December 2023

Appellant:
(Applicant)

Oracle International Corporation
500 Oracle Parkway
Mailstop 5OP7
Redwood Shores, CA 94065 (US)

Representative:

Gill Jennings & Every LLP
The Broadgate Tower
20 Primrose Street
London EC2A 2ES (GB)

Decision under appeal:

Decision of the Examining Division of the European Patent Office posted on 17 November 2020 refusing European patent application No. 13776643.2 pursuant to Article 97(2) EPC

Composition of the Board:

Chairwoman	P. San-Bento Furtado
Members:	C. Barel-Faucheux
	E. Mille

Summary of Facts and Submissions

I. The appellant (applicant) appealed against the examining division's decision refusing European patent application No. 13776643.2 originally filed as international application PCT/US2013/062052 and published as international publication WO 2014/052679.

II. The decision under appeal cited the following prior-art document:

D1: US 7 672 964 B1, published on 2 March 2010

III. The examining division decided that the subject-matter of the claims of the main request and of the auxiliary request lacked an inventive step having regard to the disclosure of document D1 (Article 56 EPC). In claim 14 of both requests, a further step of determining "an optimal amount of the historical data with which to initialize the continuous query based at least in part on the operator of the continuous query" was specified: the examining division found that dependent claim 14 should be considered either to lack an inventive step, Article 56 EPC, or to infringe Article 83 EPC, depending on whether or not the criterion for optimality was considered non-technical.

IV. In the statement of grounds of appeal, the appellant requested that the contested decision be set aside and that a patent be granted on the basis of the main request or the auxiliary request, both subject of the contested decision and resubmitted with the statement of grounds of appeal.

V. In a communication pursuant to Article 15(1) RPBA, the board expressed the view that the subject-matter of claim 1 of the main request and of the auxiliary request was not inventive having regard to the disclosure of document D1 (Article 56 EPC).

VI. In a letter of 15 November 2023 in reply to the board's communication, the appellant provided further arguments in favour of inventive step for the main request and the auxiliary request.

VII. Oral proceedings were held as scheduled. At the end of the oral proceedings, the Chair announced the board's decision.

VIII. The appellant's final requests are that the decision under appeal be set aside and that a patent be granted on the basis of the set of claims of the main request or, alternatively, the auxiliary request, both requests subject of the contested decision.

IX. Claim 1 of the main request reads as follows (itemisation added by the board):

(a) A system, comprising:

(b) means for receiving a continuous query configured to process a stream of event data,

(c) the continuous query including an identifier of the stream and an identifier of historical data associated with the stream,

(d) and the stream comprising change notifications of changes in a data store;

(e) means for generating a query graph based at least in part on the continuous query, the query graph including at least a node representing an operator of the continuous query;

(f) means for executing a plurality of instructions to at least traverse the query graph in a topological order to identify a lowest stateful operator;

(g) means for determining historical data in a data store with which to initialize the continuous query based at least in part on the lowest stateful operator of the continuous query;

(h) means for establishing a change notification listener to listen for change notifications of the data store, and buffering the change notifications in a messaging service until the continuous query is initialized;

(i) means for initializing the continuous query with at least a portion of the historical data at the lowest stateful operator identified in the query graph;

(j) means for evaluating the continuous query with respect to the stream and based at least in part on the historical data; and

(k) means for processing the buffered change notifications of the data store after initializing the continuous query."

X. Claim 1 of the auxiliary request corresponds to claim 1 of the main request with feature (k) replaced with:

(k1) means for determining, from the data store, change notifications of the buffered change notifications that are included in the initialized continuous query and change notifications of the buffered changed notifications that are not included in the initialized continuous query to eliminate double counting of change notifications included in the initialized continuous query by comparing transaction ID values in the change notifications against a highest value transaction ID in the initialized continuous query and ignoring change notifications with transaction ID values less than or equal to the highest value transaction ID in the initialized continuous query; and

(k2) means for processing the buffered change notifications of the data store with transaction values greater than the highest value transaction ID in the initialized continuous query after initializing the continuous query.

Reasons for the Decision

The application

1. The application relates to a system for evaluating continuous queries based at least in part on a data stream and historical data (paragraphs [0005] and [0006] of the published application).

Inventive step in relation to D1 - claim 1 of the main request

2. It is common ground that document D1 is an appropriate starting point for assessing inventive step of the claimed subject-matter.

3. The appellant argued that features b, c, d, g, h and k were not disclosed in document D1.

4. Claim 1 of the main request defines a system which receives a "continuous query" configured to process a stream of event data (**see features a and b**).

Document D1 discloses a system for "dynamically joining views" of enterprise data in a streaming database (D1, column 1, lines 14 to 17). The views correspond to "continuous queries". The system dynamically initialises views for a streaming database system (column 2, lines 45 to 49; column 3, lines 37 to 40). The views (Figure 7: view 710) are initialised (Figure 7: initialisation 703) at creation time and incrementally and continuously updated in response to new events (column 12, lines 60 to 64; Figure 7).

Therefore, document D1 discloses a system as specified in features a and b.

4.1 The continuous query of claim 1 of the main request includes (**see feature c**):
- an identifier of the stream and
- an identifier of "historical data" associated with the stream.

Document D1 discloses identifiers of streams (see e.g. column 5, lines 34 to 43). It also discloses an "event context" as the combination of an event view (or event continuous query) with a context, where a context can comprise "historical data" (column 4, lines 47 to 52).

Document D1 does not explicitly disclose that the continuous query includes an identifier of historical data associated with the stream. The board is however of the opinion that document D1 implicitly discloses that feature because a context includes historical or other data associated with (or "meaningful to") the event (column 7, lines 14 and 15) and an identifier of a context, which can be modelled as a relation, can be used in a continuous query (column 7, lines 14 to 19 and lines 34 to 59).

In addition, in the system of D1, the initialisation or re-initialising of the view is the process of calculating the view state based on previous events, and those preceding events can be persisted to a foreign data store, such as a file system or a relational database management system (column 12, line 54 to column 13, line 14). Stored data typically has associated identifiers which can be used in a continuous query involving the preceding events.

The board further notes that the appellant did not contest that document D1 discloses the continuous query including identifiers of the stream and of the historical data.

In view of the above, document D1 discloses feature c.

4.2 In the system of claim 1, a "query graph" is generated based "at least in part" on the continuous query. The query graph includes at least "a node" representing "an operator" of the continuous query (**see feature e**).

Document D1 discloses "query operators" like "join", "aggregation", or "projection" to be applied to streams and relations (column 4, lines 53 to 55). An "intermediate operator" feeds its result into one or more "higher order operators" within the query block as well as a "final operator" of an outermost query block (column 5, lines 26 to 33).

This is also illustrated by Figure 1B of document D1 reproduced below: streams 41 and 42 are input to an intermediate "join" operator 51 to produce a new stream, which in turn, is an input to a "group by” operator 52. Any snapshots of this new stream (e.g.

produced by the join [or joint] operator 51) are referred to as stream snapshots. The stream produced by the "group by" operator 52 is the final result of the query block, and is referred to as a "view" (e.g. view 60) (column 5, lines 43 to 56).

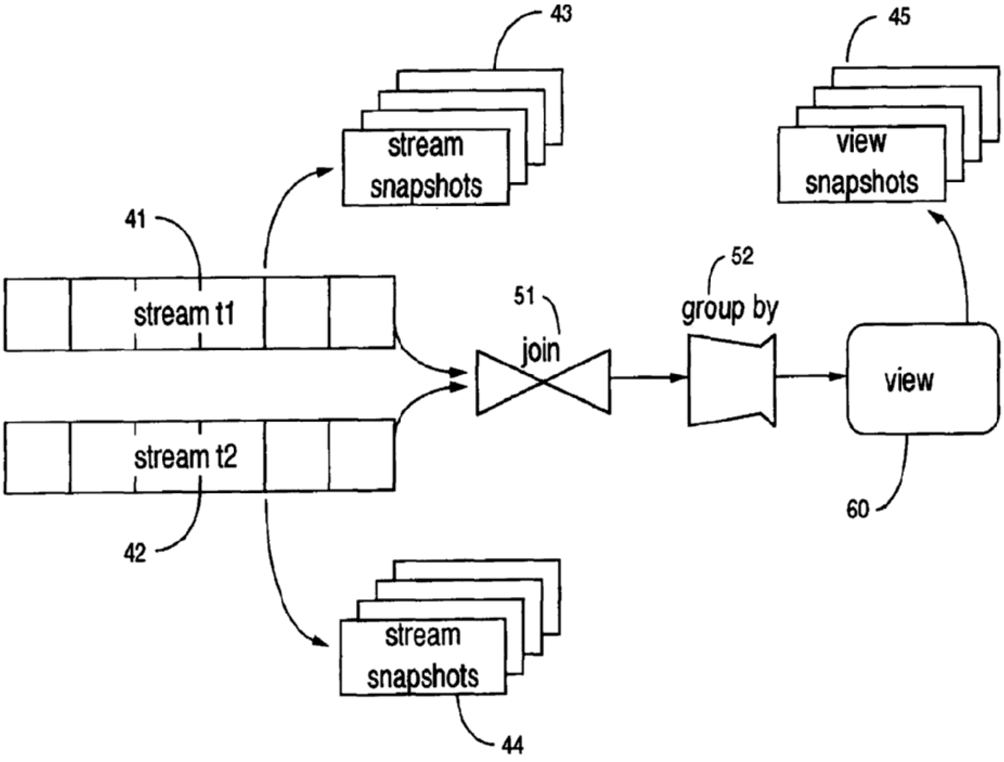

FIG. 1B

It is also clear that the main embodiments of D1 rely on traditional relational DBMS (RDBMS) technology for processing SQL queries (see e.g. column 5, lines 8 to 11, column 7, lines 34 to 45, Figure 3, column 8, lines 5 to 7), such technology using query graphs.

Therefore, document D1 discloses explicitly a hierarchy of operators and implicitly an abstract query graph including at least a node representing an operator of the continuous query, as specified in feature e.

4.3 The system of claim 1 executes a plurality of instructions to at least traverse the query graph in a "topological order" to identify a "lowest stateful operator" of the continuous query (**see feature f**).

According to the application, the "Project" operator may be stateless (page 43, lines 9 to 11), the "group by" operator is stateful (similarly to the "distinct", "group aggr" or "pattern" operators). Furthermore, "stateless" operators may just take input and send it to the parent, for example, "down-stream" operators (page 14, lines 18 to 21).

Document D1 discloses that, in the example provided by Figure 1B reproduced above, the "joint operator" 51 is "stateless" and the "group by" operator 52 is "stateful" (column 5, lines 54 to 56). Furthermore, as explained in the decision under appeal, D1 discloses traversing the query graph in a topological order ("bottom up fashion") to identify the lowest stateful operator and initialising the query at the "lowest stateful operator" (column 7, line 61 to column 8, line 31; Figures 4A and 4B and column 8, lines 20 to 23).

Therefore, the query graph of document D1 is also traversed in a "topological order" to identify a "lowest stateful operator" of the continuous query and the continuous query is initialised "at the lowest stateful operator".

Therefore, feature f is also known from document D1.

4.4 In the system of claim 1, "historical data" with which the continuous query is initialised is stored in a data store (**see first part of feature g**).

This feature is disclosed in document D1, according to which historical data can be persisted to a foreign data store (column 13, lines 5 to 8) and historic summary data of a context can be made available in a data warehouse or operational data store (column 1, lines 44 to 55).

4.5 In the system of claim 1, the continuous query is initialised with at least a portion of the historical data at the lowest stateful operator (**see last part of feature g**).

4.6 The stream of claim 1 comprises "change notifications" of changes in a data store (**see feature d**) and a "change notification listener" is established to listen for change notifications of the data store (**see first part of feature h**).

The appellant argued that D1 did not specifically describe the stream as comprising notifications of changes (in the value of a critical data element in a transactional system) in a data store. They further argued that D1 did not disclose "means for establishing a change notification listener to listen for change notifications of the data store".

The examining division did not cite any passage of document D1 that would disclose this feature.

Document D1 does not <u>explicitly</u> disclose "change notifications" of changes in a data store. However, the system of D1 subscribes to event publishers, through asynchronous publishing mechanisms such as a Java messaging service (JMS) message bag (column 6, lines 28 to 31). Similarly, the application discloses that "the CQL engine may listen to a Java Messaging Service (JMS) or other messenger for change notifications" (page 16, lines 25 to 28). Thus document D1 implicitly discloses "change notifications" and a "change notification listener".

4.6.1 In the system of claim 1, the change notifications are buffered in a "messaging service" until the continuous query is initialised (**see last part of feature h**).

Document D1 does not disclose that the change notifications are buffered until the continuous query is initialised.

4.7 The appellant argued that D1 did not describe the historical data with which a continuous query was initialised as being determined based on a lowest stateful operator of the continuous query (**see feature i**) (statement of grounds, points 14 and 16).

As explained in points 4. and 4.1 above, a "continuous query" of claim 1 corresponds to a "view" of document D1. In the system of D1, the view or continuous query is initialised "with at least a portion of the historical data".

The board agrees with the appellant that the passage of column 14, lines 50 to 54 alone does not describe the query as being initialised with historical data at the lowest stateful operator. However, the board finds that the feature is disclosed in that passage in combination with other passages of document D1, especially those describing the statefulness algorithm. As explained in point 4.3 above, document D1 discloses initialising the continuous query based at least in part on the lowest stateful operator. This applies also to historical data at the lowest stateful operator during initialisation of the continuous query.

4.8 In the system of claim 1, the continuous query is evaluated "with respect to the stream" and "based at least in part on the historical data" (**see feature j**). For the reasons given above, this is also the case in document D1.

4.9 The distinguishing features of claim 1 having regard to document D1 are thus:

(DF1) the change notifications are buffered, in the "messaging service", until the continuous query is initialised;

(DF2) the buffered change notifications are processed after the initialisation of the continuous query (**see feature k**);

(DF3) the change notifications concern changes in a data store (**see feature d**).

4.10 The system of D1 can be used for processing different types of streams. Feature (DF3) is thus a minor obvious modification of the system of D1, which does not cause a synergistic effect in combination with the other two distinguishing features.

4.11 The application describes that "missing change notifications can be eliminated by establishing the change notification listener before the initial query is started but not processing them until [...] the state initialization is done. These change notifications may be buffered in the messaging service (JMS) until the CQL engine 156 is ready to process them" (description, page 32, lines 3 to 7; see also paragraph [00101] cited by the appellant).

4.12 The appellant argued that "[i]n this way", an optimal amount of historical data was fetched for the initialisation, and change notifications that arrived during initialisation were not missed, thereby improving the processing of the stream of event data.

4.12.1 It is however not apparent from claim 1 that "an optimal amount of historical data" is fetched for the initialisation. In the reply to the board's preliminary opinion, the appellant referred to paragraph [0139] of the description as originally filed and argued that use of the lowest stateful operator allowed an optimal amount of historical data to be determined. However, use of the lowest stateful operator is not a distinguishing feature. Furthermore, paragraph [0139] and Figure 8 merely state that, in some examples, the process 800 may include determining at step 808 an optimal amount of historical data for initialising "based at least in part on the operator of the query". It does not say exactly how this optimal amount is determined.

4.13 The appellant argued that the distinguishing features they had identified combined to provide for the continuous query to be efficiently initialised whilst inhibiting the missing of change notifications that occurred during initialisation, with the overall effect therefore being an improvement in the processing of a stream of event data. The objective technical problem was therefore how to improve the processing of a stream of event data.

4.14 However, the board is of the opinion that the effect of DF1 and DF2 is the prevention of missing change notifications during initialisation and the objective technical problem solved by DF1 and DF2 can be formulated as "how to prevent missing change notifications during initialisation?". Contrary to the appellant's argumentation on this point, this formulation does not contain any pointer to the solution and does not constitute an *ex post facto* analysis. It is implicit from document D1 that change notifications may be missed during initialisation with historical data in the system of D1. This is also not excluded by the mere fact that D1 teaches a view initialisation that includes as many preceding events as possible and a view re-initialisation when a materialised view fails or needs to be updated.

4.15 The appellant argued that D1 simply focused on optimising the balance between increasing the speed of initialisation and the amount of historical data. Since D1 did not contemplate missing change notifications, or events, during initialisation, the skilled person would have had no reason to add the distinguishing features when considering improving the processing of a stream of event data (statement of grounds, point 21).

At most, when contemplating how to improve the processing of a stream of event data, the disclosure of document D1 would have prompted the skilled person to increase the speed of initialisation by allowing a user to specify the number of recent view snapshots (i.e. historical data) to be included in the initialisation. That is, the user could specify a lower number of recent view snapshots (i.e. less historical data) so that initialisation was faster, leading to a shorter initialisation time, thereby minimising the likelihood of an event being missed. Adjusting the number of recent view snapshots was itself described in D1 (reference was made to column 14, lines 32 to 38), rather than the claimed solution provided by the distinguishing features. As such, even if the skilled person had begun to consider missed events, the disclosure of document D1 would have led the skilled person to address the problem in a different manner to the claimed solution by making initialisation faster (statement of grounds, point 22).

However, reducing the historical data in order to speed up initialisation would come at the cost of sub-optimal initialisation for the continuous query. Therefore, the modifications that D1 might have prompted the skilled person to consider would still provide a less desirable system than that provided by the distinguishing features. The distinguishing features therefore provided a balanced solution to the objective technical problem in that an optimal amount of historical data could be used for optimisation whilst also accounting for any events that arrived during initialisation (statement of grounds, points 23 and 30).

4.16 The board is not convinced by this line of argument. Buffers are widely used in computer systems to temporarily store incoming data that, for example, cannot be processed at the same high pace as it is being received. At the oral proceedings, the appellant conceded that buffering was well known at the priority date but argued that it was in general used for a different purpose. Using buffering for initialisation in the context of the system of D1 would not have been obvious. The board sees no reason, however, why the skilled person would not use the JMS buffer of the system of D1 in initialisation. The board is further of the opinion that the skilled person faced with the problem of avoiding missing change notifications during initialisation in the system of D1 would consider buffering notifications during initialisation and processing them after initialisation, thereby arriving at features DF1 and DF2.

4.17 Therefore, claim 1 of the main request is not inventive (Article 56 EPC).

Auxiliary request

5. Claim 1 of the auxiliary request differs from claim 1 of the main request in that it additionally specifies means for

- determining, from the data store, buffered change notifications that are included, and those that are not included, in the initialised continuous query to eliminate double counting of change notifications included in the initialised continuous query; this is done by comparing transaction ID values in the change notifications against a highest value transaction ID in the initialised continuous query and ignoring change notifications with transaction ID values less than or equal to the highest value transaction ID in the initialised continuous query (**see feature k1**);

- processing the buffered change notifications of the data store with transaction values greater than the highest value transaction ID in the initialised continuous query after initialising the continuous query (**see feature k2**).

Inventive step in relation to document D1 - claim 1

6. The board is of the opinion that feature DF3 does not achieve a synergistic effect when combined with features k1 and k2 and is not inventive for the reasons given above for the main request.

7. Document D1 discloses unique keys with an assumed order, such as an event ID or a timestamp, used for synchronising processing of events. It discloses that a join of two tables with respect to event_id n is only processed when both streams have processed the event_id n (column 10, line 54 to column 11, line 13). In addition, document D1 also discloses transaction IDs (see also column 8, lines 59 to 64 and Figure 5). A transaction ID can be used as a key with an assumed order and as event ID.

However, document D1 does not disclose ignoring change notifications with transaction ID values less than or equal to the highest value transaction ID in the initialised continuous query.

7.1 The application explains that "eliminating double-counting of change notifications may be performed by supplying additional information to a Persistence Service to allow the CQL engine 156 to determine which change notifications are included in the initial query result and which are not" and "the transaction ID value in any change notifications may be compared against the highest value because a value less than or equal may be ignored (i.e., because it may already be counted) and a value greater than may be processed (i.e., because it may not yet be counted)" (page 32, lines 7 to 22).

The appellant stated that "[i]n some scenarios", there could be an offset/overlap between the change notification listener and the initialisation. This could result in events being double-counted in that they were in both the historical data and the buffered change notifications in the messaging service.

7.2 In the decision under appeal and in the board's preliminary opinion, the objective technical problem for the auxiliary request was considered to be that of preventing missing change notifications and double counting of change notifications for an initialised continuous query.

7.3 In the decision under appeal, the examining division was of the opinion that the additional characterisation by features k1 and k2 appeared "to reflect a particular choice, not necessarily of technical nature, of the applicant". That choice could be of a technical or non-technical nature. The application did not explain why double counting could happen. The decision under appeal then reads "Be that as it may, the division considers the requirement of avoiding duplicate data to be sufficiently generic and commonplace so as to be an implicit requirement in any such data processing system, noting that in D1 the events, as acknowledged by the applicant, relate to a 'transactional system', which might require the enforcement of a (non-technical) policy preventing 'double spending'."

7.4 The appellant argued that there was no indication in document D1 that double counting could occur. There was simply no prompting in D1 that would have made the skilled person consider what to do to avoid double-counting the buffered events, and consider using a transaction ID to ignore events with transaction ID values less than or equal to the highest transaction ID in the initialised continuous query. Document D1 provided no hint toward any of these points. Only with the benefit of hindsight could one establish that such modifications could be made to D1. Without the benefit of such hindsight, inventive thought would have been necessary to identify and address all of these points, when starting from D1 and faced with the objective technical problem of improving the processing of a stream of event data, to arrive at the distinguishing features.

7.5 In the board's opinion, the problem of avoiding double counting during initialisation in the context of a system such as that of document D1, irrespective of its origin, is a technical problem. The board agrees with the appellant that document D1 does not mention that this problem occurs in the system of document D1. In view of this, at the oral proceedings, the board formulated the objective technical problem as "how to process change notifications during initialisation".

7.6 At the oral proceedings, the board argued that it was well known to the skilled person that the problem of double counting could occur when implementing continuous query processing in a system with query initialisation using a buffer. When trying to solve the problem of processing change notifications in the system of D1, the skilled person would thus consider the problem of double counting. The solution of features k1 and k2 was not inventive. Given that D1 discloses using IDs for establishing the order of events and for the synchronised processing of events, the skilled person seeking to avoid double counting of change notifications would have considered processing the events in order. The board was of the preliminary opinion that, using their ordinary programming skills, the skilled person would envisage processing the notifications in order by keeping track of the "highest value transaction ID" (identifying the latest arrived transaction) in the initialised continuous query and comparing transaction ID values in the change notifications against this value in order to process only buffered change notifications with a transaction ID greater than the highest transaction ID.

7.7 However, the appellant contested that the skilled person would, in view of their common general knowledge, recognise that the problem of double counting could occur when implementing continuous query processing in a system with query initialisation using a buffer. There was no evidence on file of the alleged common general knowledge.

7.8 With some exceptions, if the common general knowledge relevant to the outcome of a case is disputed by a party, it normally has to be proved like any other fact under contention, for instance by documentary or oral evidence (T 939/92, OJ EPO 1996, 309, Reasons 2.3; Case Law of the Boards of Appeal, 10th edition, 2022, I.C. 2.8.5, see also T 415/21, Reasons 2.11.3).

7.9 The board came to the conclusion that in the present case evidence was necessary that double counting of change notifications was a known problem at the priority date. This evidence was not provided in the decision under appeal, presumably because the problem of double counting was seen as "sufficiently generic and commonplace so as to be an implicit requirement". The board is not aware either that such evidence has been provided in the proceedings thus far. The final decision on inventive step depends on such evidence.

7.10 The board is of the opinion that these circumstances constitute a special reason under Article 11 RPBA 2020 for remitting the case for further prosecution in order to give the examining division the opportunity to provide evidence of the alleged common general knowledge and the appellant the opportunity to comment on any new evidence that may be provided.

Conclusion

8. The main request does not fulfil the requirements of Article 56 EPC. A decision on inventive step with regard to the auxiliary request cannot be made and the case is to be remitted to the examining division for further prosecution on the basis of the auxiliary request.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.

2. The case is remitted to the examining division for further prosecution.

The Registrar: The Chairwoman:

S. Lichtenvort P. San-Bento Furtado

Decision electronically authenticated