PATENT COOPERATION TREATY

From the INTERNATIONAL SEARCHING AUTHORITY

PCT

To:
Seegers, Mark D.
MEYERTONS, HOOD, KIVLIN, KOWERT & GOETZEL, P.C.
P.O. Box 398
Austin, TX 78767-0398
ETATS-UNIS D'AMERIQUE

INVITATION TO PAY ADDITIONAL FEES
AND, WHERE APPLICABLE, PROTEST FEE

(PCT Article 17(3)(a) and Rule 40.1 and 40.2(e))

Applicant's or agent's file reference 7000-13301	Date of mailing *(day/month/year)* 25 September 2018 (25-09-2018) **PAYMENT DUE** within **ONE MONTH** from the above date of mailing
International application No. PCT/US2018/044787	International filing date *(day/month/year)* 1 August 2018 (01-08-2018)
Applicant SALESFORCE.COM, INC.	

1. This International Searching Authority

(i) considers that there are 2 *(number of)* inventions claimed in the international application covered by the claims indicated on an extra sheet:

(ii) therefore considers that **the international application does not comply with the requirements of unity of invention** (Rules 13.1, 13.2 and 13.3) for the reasons indicated on an extra sheet:

(iii) [X] has carried out a partial international search (see Annex) [ ] will establish the international search report on those parts of the international application which relate to the invention first mentioned in claims Nos.:

see extra sheet

(iv) will establish the international search report on the other parts of the international application only if, and to the extent to which, additional fees are paid.

2. Consequently, the applicant is hereby **invited to pay,** within the time limit indicated above, the amount indicated below:

EUR 1.775,00	x	1	=	EUR 1.775,00
Fee per additional invention		number of additional inventions		currency/total amount of additional fees

3. The applicant is informed that, according to Rule 40.2(c), **the payment of any additional fee may be made under protest,** i.e., a reasoned statement to the effect that the international application complies with the requirement of unity of invention or that the amount of the required additional fee is excessive, where applicable, subject to the payment of a protest fee.

Where the applicant pays additional fees under protest, the applicant is hereby invited, within the time limit indicated above, to pay a protest fee (Rule 40.2(e)) in the amount of EUR 875,00 *(currency/amount)*

Where the applicant has not, within the time limit indicated above, paid the required protest fee, the protest will be considered not to have been made and the International Searching Authority will so declare.

4. [ ] Claim(s) Nos. _____ have been found to be unsearchable under Article 17(2)(b) because of defects under Article 17(2)(a) and therefore have not been included with any invention.

Name and mailing address of the International Searching Authority	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL-2280 HV Rijswijk Tel. (+31-70) 340-2040 Fax: (+31-70) 340-3016	BERTHON, Claude Tel: +31 (0)70 340-1001

Form PCT/ISA/206 (April 2005)

Registered Mail

This International Searching Authority found multiple (groups of) inventions in this international application, as follows:

1. claims: 1-9, 17

Alternative manner to inform standby nodes of the active node's log's state.

---

2. claims: 10-16, 18-25

Permitting a database system to revert to one of its previous states

---

1 The application does not meet the requirements of unity of invention in that there are two inventions. The reasons for which the inventions are not so linked as to form a single general inventive concept (Rule 13.1 PCT) are as follows.
2 The following inventions are identified:
2.1 invention 1: claims 1-9, 17;
2.2 invention 2: claims 10-16, 18-25.
3 With regard to the first invention, D1 (US 2014/324785 A1) discloses all features of the method of independent claim 1:
3.1 A method for a database system synchronizing current state of the database system among a plurality of nodes configured to handle requests for data of the database system stored in a distributed storage
4 Figure 3, ref. signs 300, 310, 320a-c, 322a-c: shows database system with multiple nodes and a distributed storage service.
[0036]: "the database tier may support the use of synchronous or asynchronous read replicas in the system, e.g., read-only copies of data on different nodes of the database tier to which read requests can be routed."
[0053]: "database engine head node 320a [...] may route read requests [...] to a read replica and/or various storage nodes within distributed database-optimized storage service 310, [...] receive requested data pages from distributed database-optimized storage service 310"
5 The data of a database system (current state of the database system) is synchronized between multiple nodes that handle read requests for data from (stored in) a distributed storage service. Importantly, the distributed storage service, the primary/head nodes, and the read replica nodes together (not only the distributed storage service alone) constitute the claimed distributed storage.
5.1 with one of the plurality of nodes being currently active and the other nodes of the plurality of nodes being currently standby nodes, the method comprising:
6 Figure 3, ref. signs 320a-c, 322a-c
[0020]: "the read replica may be configured to convert (e.g., fail over) into a primary node (e.g., after failure of a primary node) without loss of data."
7 The nodes of the database system are head/primary nodes (active nodes) and read replica nodes. The read replica nodes are on "stand by" from being a head/primary node.
7.1 receiving, at the active node, a request to perform a first

transaction that includes committing data to the distributed storage; and
8 [0033]: "the database tier [...] may include a primary node server, which may also be referred to herein as a database engine head node server, that receives read and/or write requests from various client programs, then parses them and develops an execution plan to carry out the associated database operation(s)."
[0088]: "At 610, a write request that specifies a modification to a data record stored by a database service may be received. For example, the write request [...] by a primary node. "
[0089]: "the primary node (client side driver) may generate the log record, which may be indicative of the change to the data record "
[0090]: "the log record may be sent (e.g., by the client side driver of the primary node) to a particular server node (or multiple server nodes) of a distributed storage service that stores a version of the data page that includes the given data record. The server node may then apply the modification from the log record to the actual data page stored by the server node. "
9 The primary/head node (active node) receives a write request (transaction with at least one write operation), creates a log record reflecting the data to be written, and sends it (commits it to) to a node of the distributed storage service which in turn stores the data to be written.
9.1 in response to receiving the request: committing, by the active node, the data to the distributed storage to update the current state of the database system; and
10 all the data, including the newly written data, in the distributed storage reflects/is the state of the database system. Thus, causing the new data to be written to the distributed storage constitutes "updating" the database system's state.
10.1 causing storing, by the active node, of first metadata providing an indication of the commitment in a transaction log stored in the distributed storage,
11 [0054]: "database engine head node 320a may also include transaction log 340"
[0089]: "the primary node (client side driver) may generate the log record, which may be indicative of the change to the data record "
12 When writing data, the primary node (active node) generates a log record (metadata providing indication of the commitment). Thus, this log record (metadata) must exist in the primary node's (active node's) memory, and is thus, at least temporarily, stored by the primary node (active node). As the primary node (active node) is part of the distributed storage, the log record (metadata) that the primary node stores is stored in the distributed storage.
12.1 wherein the transaction log identifies, to the standby nodes, information for the standby nodes to know the current state of the database system.
13 [0091]: " a cache invalidation indication may be sent [...] to a plurality of read replicas) indicating that a cached version of the given data record stored in the read replica's cache is stale. [...] the cache invalidation [...] may also include the actual log record that was sent to the storage service (e.g., for application by the read replica to its cached version of the data record). "
13.1.1 The primary node (active node) sends the log record (metadata in transaction log) to read replica nodes (standby nodes) in order for them to update their cache (inform them of the data/state of/in the

database system).
14 As D1 discloses all features of the method of claim 1, none of these features can define a contribution over the prior art D1 (Rule 13.1 PCT). This applies mutatis mutandis to the medium of claim 17.
15 Moreover, the additional features of the following dependent claims that belong to the first invention are also disclosed by D1 and therefore cannot define a contribution over the prior art:
15.1 Claims 3: D1 discloses that the primary node notifies a read replica node (standby node) of new log records (modifications of transaction log) by sending said new log records (metadata) to the read replica node (standby node). In response, the read replica node (standby node) updates its cache (update second metadata at the standby node) by reading the new log records (metadata) such that the cache is up to date for future requests for data (client requests).
15.1.1 [0091]: "a cache invalidation indication may be sent to a read replica (or to a plurality of read replicas) indicating that a cached version of the given data record stored in the read replica's cache is stale. [...] the cache invalidation indication may be a simple notification [...] and [...] may also include the actual log record that was sent to the storage service (e.g., for application by the read replica to its cached version of the data record). [...] for a subsequent request for data corresponding to the stale data, the read replica will know that the date is stale and retrieve it from the storage service instead of from its cache. [...] the read replica may apply the modification specified by the log record to its cache."
15.2 Claim 4: D1 discloses that the data that is stored in the distributed storage service (part of the distriuted storage) is cached on the read replica nodes (standby nodes). Thus, the cached data corresponds to the stored data.
15.2.1 Figure 3, ref. signs 630 and 640
15.3 Claim 5: D1 discloses that the primary/head node (active node) has a log (catalog) with log records (metadata). The primary node (active node) uses these log records (metadata) to update read replica nodes (second nodes). Thus, the log (catalog) with the log records (metadata) is, in some way, shared among the nodes (primary node and also multiple read replica nodes). When the primary node (active node) has a log record in its memory (stores metdata), the primary node thereby updates its log (catalog) and notifies the read replica nodes (standby nodes).
15.3.1 Figure 3, ref. signs 320a, 340, 322a 326a
[0091], 15.4 Claim 6: D1 discloses that read replica nodes (standby nodes) update their caches based on log records (metadata) received from a primary node (active node) [0091]. To update their cache, they must implicitly access entries in the cache using a memory address. This memory address is the key, and the data in memory is the value.
15.5 Claim 7: D1 discloses that data is stored in not only in the log records (metadata) but also in the distributed storage service (which is different from and therefore extern to the transaction log in the distributed storage).
15.5.1 Figure 3, ref. sign 310
Figure 6, ref. signs 630 and 640
15.6 Claim 8: D1 discloses that a read replica node (standby node) is converted into (becomes) a primary/head node in case of failure (thus being a high availability application). As the read replica node (standby node) takes over all responsibilities of the primary node (active node), when said new primary node (new active node) receives a write request, it

writes the data (commits data of second transaction) and performs the logging (indication of second transaction in transaction log).
15.6.1 [0058]: "one of read replicas 322a, 322b, or 322c may be converted into a new database engine head node (e.g., if the head node fails)."
Figure 9
15.7 Claim 9: as elaborated above, nodes (part of the distributed storage) have logs (catalogs) with log records (metadata), whereby said log records (metadata) are implicitly accessed using memory addresses (keys). Said references (keys) point to the node's memory location (physical location in memory that is part of the distributed storage) that stores the corresponding log records (metadata providing indications of updates).
16 D1 does not appear to disclose the additional features of the method of dependent claim 2. Thus, the first invention is defined by these features. In view of the originally filed description ([0015], [0016]), the additional features of said method solve the technical problem of providing an alternative manner to inform standby nodes of the active node's log's state.
17 With regard to the second invention, D1 discloses the following features of the database system of claim 10:
17.1 A database system, comprising:
18 Figure 3, ref. sign 300
18.1 a plurality of nodes configured to implement a database; and
19 Figure 3, ref. signs 320a-c, 322a-c
19.1 a distributed storage accessible to the plurality of nodes and configured to store data of the database;
20 Figure 3, ref. sign 310
20.1 wherein a first of the plurality of nodes is configured to: receive a request to perform a first transaction that includes committing data to the distributed storage; for the first transaction, store a first set of data in the distributed storage; and
21 21.1 store a first record of the first transaction in a transaction log maintained by the distributed storage,
22 [0054]: "database engine head node 320a may also include transaction log 340"
[0089]: "the primary node (client side driver) may generate the log record, which may be indicative of the change to the data record "
23 The primary node generates a log record which means that the log record must be at least temporarily stored in the primary node's memory. As the primary node is part of the distributed storage, the log record is maintained by the distributed storage.
23.1 wherein the transaction log defines an ordering in which transactions are performed with respect to the database.
24 The database system of claim 10 therefore differs from the database system of D1 in the following distinguishing feature:
24.1 the transaction log defines an ordering in which transactions are performed with respect to the database
25 This distinguishing feature therefore defines the second invention. In view of the originally filed description [0025], said feature solves the technical problem of permitting the claimed database system to revert to one of its previous states.
26 Still with regard to the second invention, D1 discloses the following features of the method of claim 18:
26.1 A method, comprising: maintaining, by a first of a plurality

of database nodes of a database system, a cache for data stored in a distributed storage shared among the plurality of database nodes,
27 Figure 3, ref. signs 310, 320a-c, 322a-c, 326a, 335: shows that the distributed database service is accessible to primary/head nodes and read replica nodes, and thus shared among these nodes. The nodes, including read replica nodes (one of which is the first node), have a cache for the data in the distributed storage service.
27.1 wherein the cache includes an entry for a first key-value pair;
28 [0091]: "after updating its cache, [...] the read replica may remove the stale cache indication (e.g., from the data structure maintaining a list of the stale cache entries) for that particular data record."
[0100]: "At 820, the read replica's cache may be updated with versions of the data pages stored in the primary node's cache."
29 The primary/head and read replica nodes have a cache that includes stored data records. Data records must be, implicitly, readable into the memory of the node. The data record's memory address is said record's key, and the data record's content is said record's value.
reading, by the first database node, a transaction log29.1 ,
wherein the transaction log identifies an ordering in which transactions of the database system are committed to the distributed storage ;
30 [0088]: " a write request that specifies a modification to a data record stored by a database service may be received. [...] The write request may specify a modification to be made to a given data record stored in a database table. "
[0089]: "a log record representing the modification to be made to the data record may be generated ."
[0091]: "a cache invalidation indication may be sent [...] to a plurality of read replicas) indicating that a cached version of the given data record stored in the read replica's cache is stale. [...] the cache invalidation [...] may also include the actual log record that was sent to the storage service (e.g., for application by the read replica to its cached version of the data record). "
31 A log record (part of a transaction log) is sent to a read replica node (first node) for updating its cache. Thus, the read replica node (first node) must read the received log record (part of the transaction log).
31.1 based on the reading, the first database node determining that a second of the plurality of database nodes has committed, to the distributed storage, a first transaction that modifies a value of the first key-value pair; and
32 after writing (committing) data to the distributed storage, the primary node generates a log record and sends it to the read replica node (first node) which determines that its cached data entries (data records/key value pairs reflecting the data in the distributed storage) are stale (value is modified).
32.1 in response to the determining, the first database node updating the entry included in the cache based on the modified value of the first key-value pair.
33 [0091]: " the read replica will know that the date is stale and retrieve it from the storage service instead of from its cache. In an embodiment in which the cache invalidation indication includes the actual log record, the read replica may apply the modification specified by the log record to its cache. After doing so, that cache entry (and data

record) may no longer be indicated as stale. "
34 The read replica node (first node) updates the stale (modified) cache entry (data record/key value pair).
35 Thus, the method of claim 18 differs from the method of D1 in the following distinguishing feature (the same applies to independent claims 24 and 25):
35.1 the transaction log identifies an ordering in which transactions of the database system are committed to the distributed storage
36 This appears to be the same feature as the distinguishing feature of independent claim 10. Thus, claims 18, 24, 25 and 10 appear to share the same special technical feature and belong to the second invention, wherein this special technical feature defines the second invention.
37 The special technical features of the first and second invention are not the same. Additionally, these features solve different technical problems and therefore cannot be corresponding.
38 Thus, the two inventions do not share any of the same or corresponding special technical features (Rule 13.2 PCT). Consequently, these inventions are not linked by a single general inventive concept and the requirement of unity of invention is not met (Rule 13.1 PCT).

Annex to Form PCT/ISA/206
COMMUNICATION RELATING TO THE RESULTS OF THE PARTIAL INTERNATIONAL SEARCH

International Application No
PCT/US2018/044787

1. The present communication is an Annex to the invitation to pay additional fees (Form PCT/ISA/206). It shows the results of the international search established on the parts of the international application which relate to the invention first mentioned in claims Nos.:
see 'Invitation to pay additional fees'
2. This communication is not the international search report which will be established according to Article 18 and Rule 43.

3. If the applicant does not pay any additional search fees, the information appearing in this communication will be considered as the result of the international search and will be included as such in the international search report.

4. If the applicant pays additional fees, the international search report will contain both the information appearing in this communication and the results of the international search on other parts of the international application for which such fees will have been paid.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2014/324785 A1 (GUPTA ANURAG WINDLASS [US] ET AL) 30 October 2014 (2014-10-30) paragraph [0020] paragraph [0033] paragraph [0036] paragraph [0053] - paragraph [0054] paragraph [0088] - paragraph [0091] figures 3, 6	1-9,17
A	Anonymous: "Polling (computer science)", Wikipedia, 11 July 2017 (2017-07-11), pages 1-3, XP055507238, Retrieved from the Internet: URL:https://en.wikipedia.org/w/index.php?title=Polling_(computer_science)&oldid=790118174 [retrieved on 2018-09-14] page 1, paragraph 1	2

☐ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Form PCT/ISA/206 (Annex, first sheet) (July 1992; reprint January 2004)

Patent Family Annex

Information on patent family members

International Application No
PCT/US2018/044787

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2014324785 A1	30-10-2014	CA 2910270 A1	06-11-2014
		CN 105324770 A	10-02-2016
		EP 2992463 A1	09-03-2016
		JP 2016517124 A	09-06-2016
		US 2014324785 A1	30-10-2014
		WO 2014179504 A1	06-11-2014

Application no:
Demande nº:
Anmelde-Nr: PCT/US2018/044787

DISCLAIMER

The attached provisional opinion on the patentability of the first invention searched serves only as information.
A reply addressing the points raised in the opinion is **not** required and will **not** be taken into account when issuing the final search report and opinion on patentability.

AVERTISSEMENT

L'avis provisoire ci-joint sur la brevetabilité de la première invention recherchée ne sert qu'à titre d'information.
Une réponse abordant les points soulevés dans l'avis n'est **pas** nécessaire et ne sera **pas** prise en compte lors de l'établissement du rapport final de la recherche et de l'avis sur la brevetabilité.

DISCLAIMER

Die beigefügte vorläufige Stellungnahme zur Patentierbarkeit der ersten geprüften Erfindung dient lediglich zur Information.
Eine Antwort auf die erhobenen Punkte in der Stellungnahme ist **nicht** erforderlich und bleibt bei der Erstellung des endgültigen Recherchenberichts und der Stellungnahme zur Patentierbarkeit **unberücksichtigt**.

Re Item IV
Lack of unity of invention

1 The application does not meet the requirements of unity of invention in that there are two inventions. The reasons for which the inventions are not so linked as to form a single general inventive concept (Rule 13.1 PCT) are as follows.

2 The following inventions are identified:

2.1 invention 1: claims 1-9, 17;

2.2 invention 2: claims 10-16, 18-25.

3 With regard to the first invention, D1 (US 2014/324785 A1) discloses all features of the method of independent claim 1:

3.1 A method for a database system synchronizing current state of the database system among a plurality of nodes configured to handle requests for data of the database system stored in a distributed storage

Figure 3, ref. signs 300, 310, 320a-c, 322a-c: shows database system with multiple nodes and a distributed storage service.
[0036]: "the database tier may support the use of synchronous or asynchronous read replicas in the system, e.g., read-only copies of data on different nodes of the database tier to which read requests can be routed."
[0053]: "database engine head node 320a [...] may route read requests [...] to a read replica and/or various storage nodes within distributed database-optimized storage service 310, [...] receive requested data pages from distributed database-optimized storage service 310"

The data of a database system (current state of the database system) is synchronized between multiple nodes that handle read requests for data from (stored in) a distributed storage service. Importantly, the distributed storage service, the primary/head nodes, and the read replica nodes together (not only the distributed storage service alone) constitute the claimed distributed storage.

3.2 with one of the plurality of nodes being currently active and the other nodes of the plurality of nodes being currently standby nodes, the method comprising:

Figure 3, ref. signs 320a-c, 322a-c
[0020]: "the read replica may be configured to convert (e.g., fail over) into a primary node (e.g., after failure of a primary node) without loss of data."

The nodes of the database system are head/primary nodes (active nodes) and read replica nodes. The read replica nodes are on "stand by" from being a head/primary node.

3.3 receiving, at the active node, a request to perform a first transaction that includes committing data to the distributed storage; and

[0033]: "the database tier [...] may include a primary node server, which may also be referred to herein as a database engine head node server, that receives read and/or write requests from various client programs, then parses them and develops an execution plan to carry out the associated database operation(s)."
[0088]: "At 610, a write request that specifies a modification to a data record stored by a database service may be received. For example, the write request [...] by a primary node."
[0089]: "the primary node (client side driver) may generate the log record, which may be indicative of the change to the data record"
[0090]: "the log record may be sent (e.g., by the client side driver of the primary node) to a particular server node (or multiple server nodes) of a distributed storage service that stores a version of the data page that includes the given data record. The server node may then apply the modification from the log record to the actual data page stored by the server node."

The primary/head node (active node) receives a write request (transaction with at least one write operation), creates a log record reflecting the data to be written, and sends it (commits it to) to a node of the distributed storage service which in turn stores the data to be written.

3.4 in response to receiving the request: committing, by the active node, the data to the distributed storage to update the current state of the database system; and

see §3.3: all the data, including the newly written data, in the distributed storage reflects/is the state of the database system. Thus, causing the new data to be written to the distributed storage constitutes "updating" the database system's state.

3.5 causing storing, by the active node, of first metadata providing an indication of the commitment in a transaction log stored in the distributed storage,

[0054]: "database engine head node 320a may also include transaction log 340"
[0089]: "the primary node (client side driver) may generate the log record, which may be indicative of the change to the data record"

When writing data, the primary node (active node) generates a log record (metadata providing indication of the commitment). Thus, this log record (metadata) must exist in the primary node's (active node's) memory, and is thus, at least temporarily, stored by the primary node (active node). As the primary node (active node) is part of the distributed storage (see §3.1, last sentence), the log record (metadata) that the primary node stores is stored in the distributed storage.

3.6 wherein the transaction log identifies, to the standby nodes, information for the standby nodes to know the current state of the database system.

[0091]: "a cache invalidation indication may be sent [...] to a plurality of read replicas) indicating that a cached version of the given data record stored in the read replica's cache is stale. [...] the cache invalidation [...] may also include the actual log record that was sent to the storage service (e.g., for application by the read replica to its cached version of the data record)."

The primary node (active node) sends the log record (metadata in transaction log) to read replica nodes (standby nodes) in order for them to update their cache (inform them of the data/state of/in the database system).

4 As D1 discloses all features of the method of claim 1, none of these features can define a contribution over the prior art D1 (Rule 13.1 PCT). This applies mutatis mutandis to the medium of claim 17.

5 Moreover, the additional features of the following dependent claims that belong to the first invention are also disclosed by D1 and therefore cannot define a contribution over the prior art:

5.1 Claims 3: D1 discloses that the primary node notifies a read replica node (standby node) of new log records (modifications of transaction log) by sending said new log records (metadata) to the read replica node (standby node). In response, the read replica node (standby node) updates its cache (update second metadata at the standby node) by reading the new log records (metadata) such that the cache is up to date for future requests for data (client requests).

[0091]: "a cache invalidation indication may be sent to a read replica (or to a plurality of read replicas) indicating that a cached version of the given data record stored in the read replica's cache is stale. [...] the cache invalidation indication may be a simple notification [...] and [...] may also include the actual log record that was sent to the storage service (e.g., for application by the read replica to its cached version of the data record). [...] for a subsequent request for data corresponding to the stale data, the read replica will know that the date is stale and retrieve it from the storage service instead of from its cache. [...] the read replica may apply the modification specified by the log record to its cache."

5.2 Claim 4: D1 discloses that the data that is stored in the distributed storage service (part of the distriuted storage) is cached on the read replica nodes (standby nodes). Thus, the cached data corresponds to the stored data.

Figure 3, ref. signs 630 and 640

5.3 Claim 5: D1 discloses that the primary/head node (active node) has a log (catalog) with log records (metadata). The primary node (active node) uses these log records (metadata) to update read replica nodes (second nodes). Thus, the log (catalog) with the log records (metadata) is, in some way, shared among the nodes (primary node and also multiple read replica nodes). When the primary node (active node) has a log record in its memory (stores metdata), the primary node thereby updates its log (catalog) and notifies the read replica nodes (standby nodes).

Figure 3, ref. signs 320a, 340, 322a 326a
[0091], see §5.1

5.4 Claim 6: D1 discloses that read replica nodes (standby nodes) update their caches based on log records (metadata) received from a primary node (active node) [0091]. To update their cache, they must implicitly access entries in the cache using a memory address. This memory address is the key, and the data in memory is the value.

5.5 Claim 7: D1 discloses that data is stored in not only in the log records (metadata) but also in the distributed storage service (which is different from and therefore extern to the transaction log in the distributed storage).

Figure 3, ref. sign 310
Figure 6, ref. signs 630 and 640

5.6 Claim 8: D1 discloses that a read replica node (standby node) is converted into (becomes) a primary/head node in case of failure (thus being a high availability application). As the read replica node (standby node) takes over all responsibilities of the primary node (active node), when said new primary node (new active node) receives a write request, it writes the data (commits data of second transaction) and performs the logging (indication of second transaction in transaction log).

[0058]: "one of read replicas 322a, 322b, or 322c may be converted into a new database engine head node (e.g., if the head node fails)."
Figure 9

5.7 Claim 9: as elaborated above (§5.4), nodes (part of the distributed storage) have logs (catalogs) with log records (metadata), whereby said log records (metadata) are implicitly accessed using memory addresses (keys). Said references (keys) point to the node's memory location (physical location in memory that is part of the distributed storage) that stores the corresponding log records (metadata providing indications of updates).

6 D1 does not appear to disclose the additional features of the method of dependent claim 2. Thus, the first invention is defined by these features. In view of the originally filed description ([0015], [0016]), the additional features of said method solve the technical problem of providing an alternative manner to inform standby nodes of the active node's log's state.

7 With regard to the second invention, D1 discloses the following features of the database system of claim 10:

7.1 A database system, comprising:

Figure 3, ref. sign 300

7.2 a plurality of nodes configured to implement a database; and

Figure 3, ref. signs 320a-c, 322a-c

7.3 a distributed storage accessible to the plurality of nodes and configured to store data of the database;

Figure 3, ref. sign 310

7.4 wherein a first of the plurality of nodes is configured to: receive a request to perform a first transaction that includes committing data to the distributed storage; for the first transaction, store a first set of data in the distributed storage; and

see §3.3

7.5 store a first record of the first transaction in a transaction log maintained by the distributed storage,

[0054]: "database engine head node 320a may also include transaction log 340"
[0089]: "the primary node (client side driver) may generate the log record, which may be indicative of the change to the data record"

The primary node generates a log record which means that the log record must be at least temporarily stored in the primary node's memory. As the primary node is part of the distributed storage (see §3.1, last sentence), the log record is maintained by the distributed storage.

7.6 ~~wherein the transaction log defines an ordering in which transactions are performed with respect to the database.~~

8 The database system of claim 10 therefore differs from the database system of D1 in the following distinguishing feature:

8.1 the transaction log defines an ordering in which transactions are performed with respect to the database

9 This distinguishing feature therefore defines the second invention. In view of the originally filed description [0025], said feature solves the technical problem of permitting the claimed database system to revert to one of its previous states.

10 Still with regard to the second invention, D1 discloses the following features of the method of claim 18:

10.1 A method, comprising: maintaining, by a first of a plurality of database nodes of a database system, a cache for data stored in a distributed storage shared among the plurality of database nodes,

Figure 3, ref. signs 310, 320a-c, 322a-c, 326a, 335: shows that the distributed database service is accessible to primary/head nodes and read replica nodes, and thus shared among these nodes. The nodes, including read replica nodes (one of which is the first node), have a cache for the data in the distributed storage service.

10.2 wherein the cache includes an entry for a first key-value pair;

[0091]: "after updating its cache, [...] the read replica may remove the stale cache indication (e.g., from the data structure maintaining a list of the stale cache entries) for that particular data record."
[0100]: "At 820, the read replica's cache may be updated with versions of the data pages stored in the primary node's cache."

The primary/head and read replica nodes have a cache that includes stored data records. Data records must be, implicitly, readable into the memory of the node. The data record's memory address is said record's key, and the data record's content is said record's value.

10.3 reading, by the first database node, a transaction log~~,~~ ~~wherein the transaction log identifies an ordering in which transactions of the database system are committed to the distributed storage~~;

[0088]: "a write request that specifies a modification to a data record stored by a database service may be received. [...] The write request may specify a modification to be made to a given data record stored in a database table."
[0089]: "a log record representing the modification to be made to the data record may be generated."
[0091]: "a cache invalidation indication may be sent [...] to a plurality of read replicas) indicating that a cached version of the given data record stored in the read replica's cache is stale. [...] the cache invalidation [...] may also include the actual log record that was sent to the storage service (e.g., for application by the read replica to its cached version of the data record)."

A log record (part of a transaction log) is sent to a read replica node (first node) for updating its cache. Thus, the read replica node (first node) must read the received log record (part of the transaction log).

10.4 based on the reading, the first database node determining that a second of the plurality of database nodes has committed, to the distributed storage, a first transaction that modifies a value of the first key-value pair; and

see §10.3: after writing (committing) data to the distributed storage, the primary node generates a log record and sends it to the read replica node (first node) which determines that its cached data entries (data records/key value pairs reflecting the data in the distributed storage) are stale (value is modified).

10.5 in response to the determining, the first database node updating the entry included in the cache based on the modified value of the first key-value pair.

[0091]: "the read replica will know that the date is stale and retrieve it from the storage service instead of from its cache. In an embodiment in which the cache invalidation indication includes the actual log record, the read replica may apply the modification specified by the log record to its cache. After doing so, that cache entry (and data record) may no longer be indicated as stale."

The read replica node (first node) updates the stale (modified) cache entry (data record/key value pair).

11 Thus, the method of claim 18 differs from the method of D1 in the following distinguishing feature (the same applies to independent claims 24 and 25):

11.1 the transaction log identifies an ordering in which transactions of the database system are committed to the distributed storage

12 This appears to be the same feature as the distinguishing feature of independent claim 10 (§8.1). Thus, claims 18, 24, 25 and 10 appear to share the same special technical feature and belong to the second invention, wherein this special technical feature defines the second invention.

13 The special technical features of the first and second invention are not the same (§6 versus §8.1/11.1). Additionally, these features solve different technical problems (§6, 9) and therefore cannot be corresponding.

14 Thus, the two inventions do not share any of the same or corresponding special technical features (Rule 13.2 PCT). Consequently, these inventions are not linked by a single general inventive concept and the requirement of unity of invention is not met (Rule 13.1 PCT).

15 The remainder of this opinion is limited to the first invention.

Re Item V

Reasoned statement with regard to novelty, inventive step or industrial applicability; citations and explanations supporting such statement

16 **Summary**

16.1 With regard to Article 33(1) PCT, the claimed subject matter is not new in the sense of Article 33(2) PCT and does not involve an inventive step in the sense of Article 33(3) PCT.

17 **Prior Art**

17.1 Reference is made to the following documents:

D1 US 2014/324785 A1 (GUPTA ANURAG WINDLASS [US] ET AL) 30 October 2014 (2014-10-30)

D2 Anonymous: "Polling (computer science)", Wikipedia, 11 July 2017 (2017-07-11), pages 1-3, XP055507238, Retrieved from the Internet: URL:https://en.wikipedia.org/w/index.php?title=Polling_(computer_science)&oldid=790118174 [retrieved on 2018-09-14]

18 **Novelty (Article 33(2) PCT)**

18.1 With regard to Article 33(1) PCT, the claimed subject matter is not new in the sense of Article 33(2) PCT for the following reasons.

18.2 As elaborated above, D1 anticipates the subject matter of claims 1, 3-9, and 17 (see §3-5.7).

19 **Inventive Step (Article 33(3) PCT)**

19.1 With regard to Article 33(1) PCT, the claimed subject matter does not involve an inventive step within the meaning of Article 33(3) PCT for the following reasons.

19.2 Claim 2 defines the following additional features:

19.2.1 monitoring, by one of the standby nodes, a catalog of the database system that identifies new transactions that have been committed to the transaction log,

19.2.2 wherein the catalog stores a database schema for the database system; and

19.2.3 prior to reading the transaction log, the standby node determining that the catalog identifies a new transaction committed to the distributed storage.

19.3 The method of claim 2 therefore differs from the closest prior art in that the standby nodes (read replica nodes) monitor a catalog for newly committed transactions (new writes). When a new transaction is committed (new write performed), the standby node (read replica node) learns it from the catalog and reads the transaction log (log records). Additionally, the catalog stores a database schema.

19.4 First, it is not apparent what credible technical effect is supposed to be produced by the feature that the catalog stores a database schema (§19.2.2). The claimed database schema is nowhere used and it is not readily apparent, even in light of the description, what its function is. Thus, said feature appears to be an arbitrary modification of the prior art that cannot support an inventive step in the sense of Article 33(3) PCT (PCT-EPO Guidelines G-VII, 10.1).

19.5 The remaining distinguishing features (§19.2.1, 19.2.3) produce the technical effect that the active node (primary node) does not need to send a notification to the standby nodes (read replica nodes) in case of a newly committed transaction (write operation). Thus, and in view of the originally filed description ([0015], [0016]), these distinguishing features solve the technical problem of how to, in the context of the method of D1, provide an alternative manner to inform standby nodes of the active node's log's state.

19.6 In the method of D1, informing read replica nodes (standby nodes) of updates is implemented using the widely known observer pattern. Multiple read replica nodes "observe" a primary node, meaning that the primary node notifies the read replica nodes of updates. A well known alternative to the observer pattern is polling. In this commonly used implementation alternative, the read replica nodes (observers) periodically "poll" (monitor) the primary node for changes.

19.7 Polling is common general knowledge of the skilled person and for instance described in Wikipedia article D2.

p.1, first paragraph: "Polling, or polled operation, in computer science, refers to actively sampling the status of an external device by a client program as a synchronous activity."

19.8 For these reasons, when confronted with the aforementioned objective technical problem, the skilled person would adapt the method of D1 and configure the read replica nodes (standby nodes) to poll (monitor) the primary node (active node) for any updates and read its transaction log in case of new write operations (committed transactions, updates). Thus, the skilled person would adapt the method of D1 and arrive at the subject matter of claim 2 without exercising any inventive skill. For these reasons, said subject matter does not involve an inventive step within the meaning of Article 33(3) PCT.

Re Item VII
Certain defects in the international application

20 Contrary to the requirements of Rule 5.1(a)(ii) PCT, the relevant background art disclosed in D1 is not mentioned in the description, nor is this document identified therein.

21 The claims do not contain reference signs to the drawings as required by Rule 6.2(b) PCT.