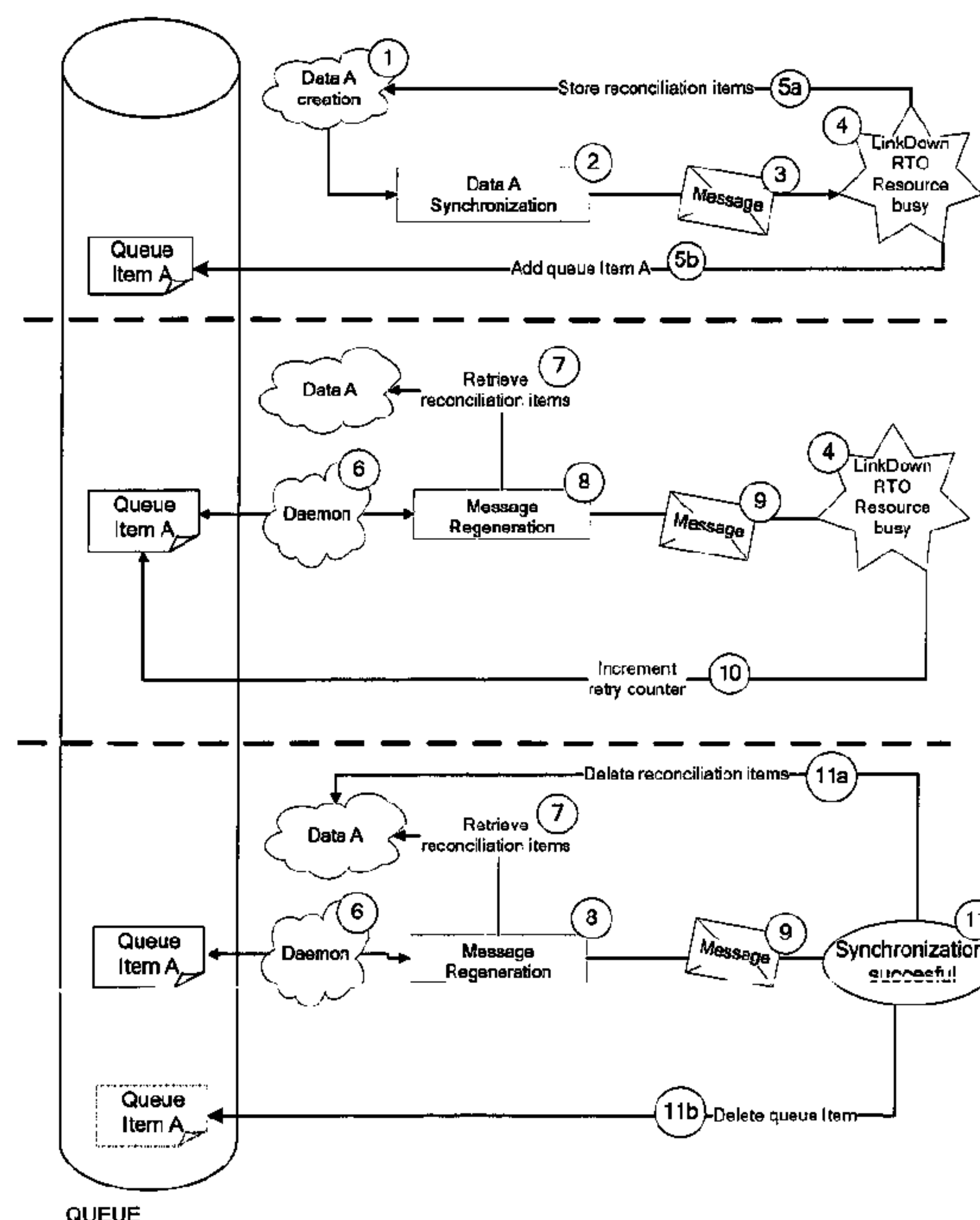




(86) Date de dépôt PCT/PCT Filing Date: 2010/10/11
(87) Date publication PCT/PCT Publication Date: 2011/04/14
(45) Date de délivrance/Issue Date: 2018/03/06
(85) Entrée phase nationale/National Entry: 2012/03/30
(86) N° demande PCT/PCT Application No.: EP 2010/006186
(87) N° publication PCT/PCT Publication No.: 2011/042203
(30) Priorité/Priority: 2009/10/09 (EP09305960.8)

(51) Cl.Int./Int.Cl. *G06F 11/14* (2006.01),
G06F 17/30 (2006.01)
(72) Inventeurs/Inventors:
SOROKIN, KONSTANTIN, FR;
MAI, HUONG-LY, FR;
BRAGANTI, HERVE, FR;
PRIoux, SARAH, FR;
HABERMACHER, ERIC, FR;
WHITFIELD, ANNICK, FR
(73) Propriétaire/Owner:
AMADEUS S.A.S., FR
(74) Agent: LESPERANCE & MARTINEAU S.E.N.C.

(54) Titre : LIVRAISON AVEC RECONCILIATION DU COTE CLIENT
(54) Title: DELIVERY WITH RECONCILIATION ON CLIENT SIDE



(57) Abrégé/Abstract:

The invention relates to a system for data synchronization between two or more computer terminals comprising, at least one client terminal, a server terminal, a communication network connecting said client and server terminals, a data string being created on said client terminal, said client terminal being configured to send the data string to the server terminal for synchronization between the two terminals, characterized in that upon synchronization failure, reconciliation data with the latest synchronization information including said data string, is configured to be stored in a database on the client terminal and resent later according to a retry counter.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau(43) International Publication Date
14 April 2011 (14.04.2011)(10) International Publication Number
WO 2011/042203 A1(51) International Patent Classification:
G06F 11/14 (2006.01) *G06F 17/30* (2006.01)[FR/FR]; 15, allée des chênes, Les hauts de Valbonne,
F-06560 Valbonne (FR).(21) International Application Number:
PCT/EP2010/006186(74) Agent: **HAUTIER, Nicolas**; Cabinet Hautier, 20 rue de
la Liberté, F-06000 Nice (FR).(22) International Filing Date:
11 October 2010 (11.10.2010)(81) Designated States (*unless otherwise indicated, for every
kind of national protection available*): AE, AG, AL, AM,
AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ,
CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO,
DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT,
HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP,
KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD,
ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI,
NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD,
SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR,
TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09305960.8 9 October 2009 (09.10.2009) EP(71) Applicant (*for all designated States except US*):
AMADEUS S.A.S [FR/FR]; 485 Route du Pin Montard,
Sophia Antipolis, F-06410 Biot (FR).

(72) Inventors; and

(75) Inventors/Applicants (*for US only*): **SOROKIN, Kon-**
stantin [RU/FR]; Résidence Aurélia, Bât 3, 741 Chemin
des moyennes breguières, F-06600 Antibes (FR). **MAI,**
Huong-Ly [FR/FR]; C/O Cabinet Hautier, 20 rue de la
Liberté, F-06000 Nice (FR). **BRAGANTI, Hervé**
[FR/FR]; 122 T Corniche des Oliviers, F-06000 Nice
(FR). **PRIOUX, Sarah** [FR/FR]; 121, chemin des basses
ribes, F-06130 Grasse (FR). **HABERMACHER, Eric**
[FR/FR]; Le Tivoli de l'Estérel, 9, impasse des figuiers,
F-06150 Cannes la Bocca (FR). **WHITFIELD, Annick**(84) Designated States (*unless otherwise indicated, for every
kind of regional protection available*): ARIPO (BW, GH,
GM, KE, LR, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG,
ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ,
TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK,
EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU,
LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK,
SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ,
GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: DELIVERY WITH RECONCILIATION ON CLIENT SIDE

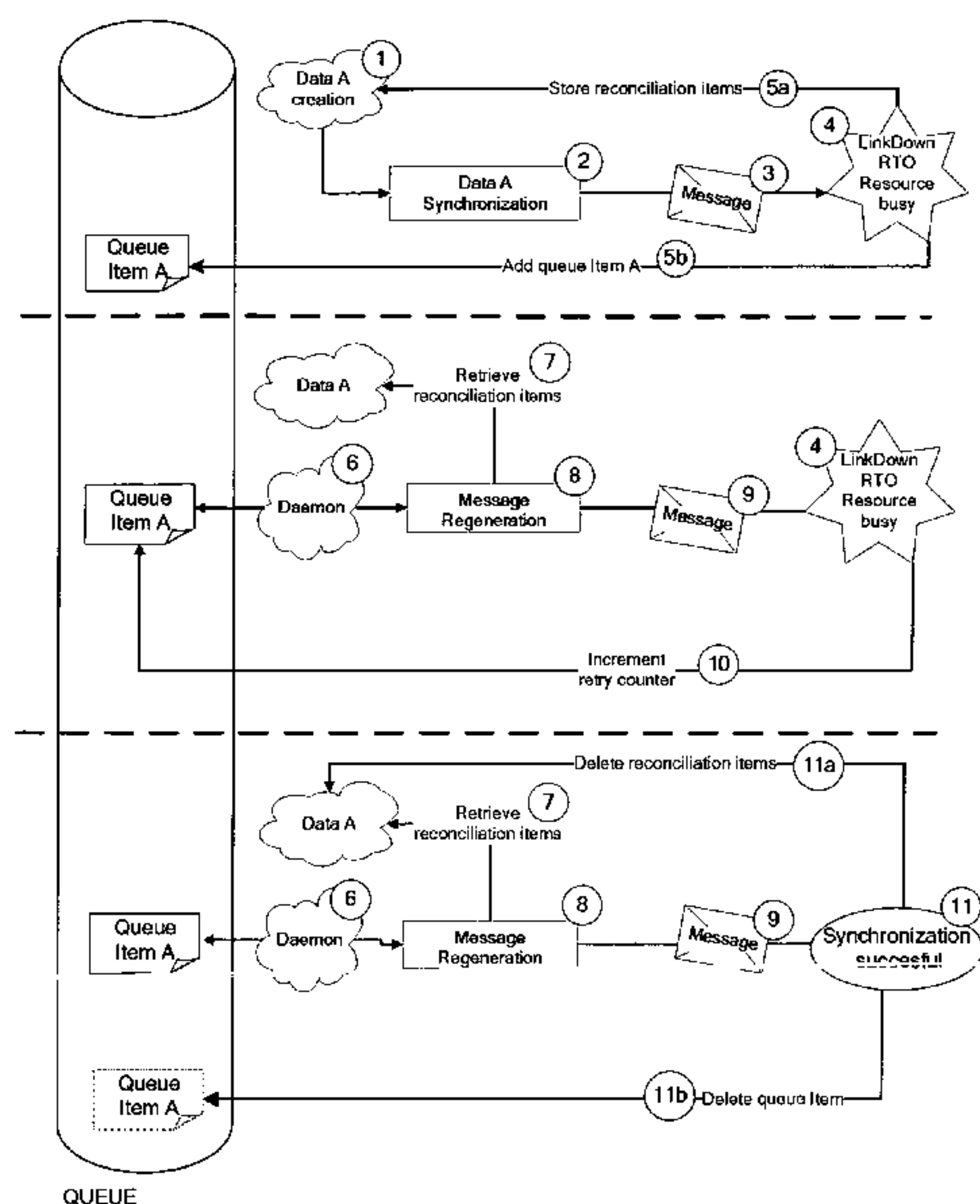
(57) Abstract: The invention relates to a system for data synchronization
between two or more computer terminals comprising, at least one client
terminal, a server terminal, a communication network connecting said
client and server terminals, a data string being created on said client ter-
minal, said client terminal being configured to send the data string to the serv-
er terminal for synchronization between the two terminals, characterized in
that upon synchronization failure, reconciliation data with the latest syn-
chronization information including said data string, is configured to be
stored in a database on the client terminal and resent later according to a
retry counter.

FIG 1

WO 2011/042203 A1

WO 2011/042203 A1



Published:

— *with international search report (Art. 21(3))*

5

10

15

Delivery with reconciliation on client side

20

The present invention relates to a data synchronization system between terminals, namely, client terminals and server terminals. In more specific terms, the present invention relates to a data synchronization system between a reservation system RS and an Airline Inventory System AIS. The present invention further relates to a method of data synchronization between these

25

terminals.

The term terminal used in this application should be interpreted by its broadest definition, comprising any computing device or any entrance or exit node for input or output.

30

As an example of a reservation system, let's consider Amadeus Reservation System that provides a traveler with a combined view of possible

travel solutions from one point to another. The chosen travel is sent by a reservation system to a corresponding airline. The reservation system further provides the confirmation of the booking and stores passenger data in a dedicated data string called Passenger Name Record (PNR) or Structured
5 Booking Record (SBR). Thus such system has passenger oriented architecture.

AIS is a system that is dedicated to store and process flight information of an airline. Hence it has a flight and date oriented architecture. The communication between both systems is done through an on-line link using
10 EDIFACT messages (Electronic Data Interchange For Administration, Commerce and Transport) on every transaction on the itinerary data or passenger data on the reservation system.

A number of RSs can be connected on-line to a single airline inventory
15 system and thereby representing multiple clients on a single server, where RSs are considered clients and AIS the server.

AIS should be notified instantly each time a creation or modification of a booking gets committed. The key information that should be communicated to
20 AIS is:

- flight related information
- overall trip related information
- passenger related information
- special service requests that are done by a passenger

25

On such notifications, AIS may provide supplementary data in the response that corresponds to a booking. This information is stored on the client terminal for passenger notification.

30 The same booking can be modified several times in a short period of time generating high traffic between both systems which may restrict the

performance of the Airline Inventory System. This may lead to an on-line link disruption, which results in a loss of synchronization between the systems.

Thus a powerful mechanism should be put in place between the systems to guarantee correct booking synchronization. Such mechanisms are implemented on a server terminal and thereby contributing to their complexity and increase the traffic between the systems.

When data synchronization between a client and a server is done through on-line links, the message exchange can fail because of time out, link down issue or other possible errors (resource busy on server side, data not yet available on server side...). The synchronization information needs to be resent later.

The objective of invention is to assure synchronization between the systems, while reducing complexity from the server terminal without important traffic increase between the systems.

To solve said problem of complexity while assuring synchronization between the systems, the invention provides a system where all synchronization responsibilities between two systems are moved to a client side only. Client guarantees to a server that it sends the most recent booking image with all actions that have been taken since the last successful synchronization between two systems on a concerned booking.

25

One and only one client can be a master of a booking thus it is aware of all modifications that are subject to synchronization with a server.

To move responsibilities to a client it is necessary to

30 - Have a dedicated storage for each modification done on a particular booking.

- Define the rules that express the importance of each type of modification and their interactions.
- Allow to implement a reconciliation process on a client side for all modifications that were not previously synchronized.

5

Thus enabling complexity reduction on a server side without any increase of traffic between two systems.

To solve the technical problem above, the present invention provides a
10 system for data synchronization between two or more computer terminals comprising, at least one client terminal, a server terminal, a communication network connecting said client and server terminals, a data string being created on said client terminal, said client terminal being configured to send the data string to the server terminal for synchronization between the two terminals,
15 characterized in that upon synchronization failure, reconciliation data with the latest synchronization information including said data string, is configured to be stored in a database on the client terminal and resent later.

In one example the client terminal is a reservation system (RS) and the server terminal is an Airline Inventory System (AIS). In one embodiment the
20 data string includes PNR information.

A synchronization failure includes network errors, online link unavailability, lack of confirmation response from the server terminal, the server terminal being busy
25

The stored reconciliation data will always be modified to contain the latest synchronization information. After modification, the stored reconciliation data is placed in a queue containing unsynchronized data.

A daemon is configured to determine if the unsynchronized queued data
30 is eligible for synchronization.

The modification of the reconciliation data is based upon a reconciliation process including adding the modified data string to the original data string, merging the modified data string to the original data string or replacing the original data string by the modified data string.

5

Furthermore the counter is configured to reinitiate synchronization according to an exponential backoff algorithm.

Finally, the reconciliation data is removed from the database on the client terminal after a successful synchronization with the server terminal.

The present invention also describes a method for data synchronization for a system as already described above where a successful synchronization is reached without subsequent modification of the reconciliation data includes the steps of:

- Creating a new or modifying an existing data string on the client terminal
- Triggering the synchronization process
- Generating a synchronization message
- Synchronization failing
- Reconciliation information being stored in a database on the client terminal
- Adding a queue item in the queue
- The daemon determining if the queue item is eligible for synchronization
- Retrieving the reconciliation information
- Regenerating the previous message
- Incrementing a retry counter
- Synchronizing successfully
- Removing the reconciliation data
- Deleting the queue item.

Another object of the present invention is a method for data synchronization for a system as already described above where a successful synchronization with subsequent modification of the reconciliation data includes:
5 the following steps performed with at least one data processor:

- Creating a new or modifying an existing data string on the client terminal
- Triggering the synchronization process
- 10 -Generating a synchronization message
- Synchronization failing
- Reconciliation information being stored in a database on the client terminal
- Said data string subsequent modification
- 15 -Reconciling the reconciliation item with the current data
- Regenerating the synchronization message containing updated reconciliation information
- Incrementing a retry counter
- Synchronizing successfully
- 20 -Removing the reconciliation data
- Deleting the queue item.

The invention also relates to a method for data synchronization for a system as already described above where an unsuccessful synchronization without subsequent modification of the reconciliation data includes the following
25 steps performed with at least one data processor:

- Creating a new or modifying an existing data string on the client terminal
- 30 -Triggering the synchronization process
- Generating a synchronization message
- Synchronization failing

- Reconciliation information being stored in a database on the client terminal
- Adding a queue item in the queue
- The daemon determining if the queue item is eligible for synchronization
- Retrieving the reconciliation information
- Regenerating the previous message
- Incrementing a retry counter
- Reaching the maximum number of retries

10

Another subject matter of the present invention is a method as set forth in any one of the two preceding paragraphs, where the method is performed as a result of execution of computer software by at least one data processor, the computer software being stored in a non-transitory computer-readable memory medium.

15

Another subject matter of the present invention is a computer program product stored on a non-transitory computer readable storage medium, comprising computer readable code means for causing at least one computer to operate the method according to any one of the three preceding paragraphs.

20

A preferred embodiment of the invention will now be described in further details according to the drawings.

25

Figure 1 shows a schematic view of the invention process without subsequent modification on the data, ended by a successful synchronization

Figure 2 shows a schematic view of the invention process with subsequent modification on the data, ended by a successful synchronization

30

Figure 3 shows a schematic view of the invention process without subsequent modification on the data, ended by an unsuccessful synchronization

As an example for fig. 1 a PNR (Passenger name record) with booking and passenger data is created and committed on the reservation system (1). This triggers a synchronization process (2) and a synchronization message is generated (3). The message contains information about the booking itself and the actions on the PNR called transactional information. Synchronization can fail (4) for several reasons. Reasons could be that the sending of the synchronization message fails due to on-line link unavailability, that the message is sent but no confirmation response is received from the server terminal or even that the resource on the server terminal is busy and the server
10 replies with a given error code.

If a synchronization failure is identified, the PNR message gets into a synchronization pending state. The reservation system stores the data in a dedicated database as reconciliation items (5a). Reconciliation items are linked
15 to the PNR and contain information to identify the data that were not yet synchronized and the information that is needed for reconciliation process. A queue item is also added in a dedicated queue (5b). A daemon (6) is set up against a dedicated queue. It analyses each queue item. According to the retry
20 counter and the time stamp of the item, it determines whether synchronization should be triggered. If yes, the reconciliation item is retrieved (7), a message regeneration process (8) builds a synchronization message (9) and sends it again to the server.

If the synchronization was not successful, the retry counter of the queue
25 will be incremented (10). The PNR data remains in the synchronization pending state and the synchronization will then be retried according to an exponential back-off algorithm.

Upon a successful synchronization (11), the reconciliation data and the queue item are removed from the client terminal. The PNR data is not in the
30 synchronization pending state anymore.

To avoid infinite retries, when the maximum number of retries is reached and the synchronization still fails, the queued item is moved to a rejected item queue as on figure 3. However the reconciliation item is kept and the PNR data remains in the synchronization pending state. Thus in case of a subsequent
5 modification, for example a subsequent synchronization, the information that is still not synchronized with the server terminal can be reconciled with the synchronization message and resent.

This mechanism ensures that the modifications are sent to the server
10 terminal even though the first attempt fails and no subsequent synchronization is triggered.

Other issues may also rise for example where PNR data is in the synchronization pending state. If no modification of the PNR is introduced then
15 the synchronization continues according to the above and as described in fig 1 and 3. However, if a modification of the PNR is presented before the synchronization is triggered, then that leads to a second synchronization process shown in fig 2.

20 In fact, each time synchronization is triggered, reconciliation items are stored in the dedicated database. These items are analyzed and compared to the modifications that have been done to the PNR (12) in the current transaction. Based on earlier defined reconciliation rules, a decision is taken deciding if the corresponding item should be added, removed or merged (7) to
25 the content of the synchronization message that will be sent to the server terminal. Thus a synchronization message (8) containing the most recent PNR and all actions not yet synchronized with the server.

Thanks to the reconciliation rules, the client terminal decides the
30 pertinence and the coherence of the message content, so that the message reflects all the information on the server terminal to be synchronized. This way,

the logic on the server terminal to handle synchronization failure is fairly reduced.

CLAIMS

1. A system for data synchronization between one or more reservation systems and an airline inventory system connected by a communication network, such that the one or more reservation systems are responsible for the data synchronization, the system comprising:

at least one processor; and

a memory storing instructions that, when executed by the at least one processor, cause the system to:

create a first data string at one of the one or more reservation systems, the first data string including information relating to a creation or modification of a passenger name record on the one reservation system;

in response to the creation of the first data string, trigger a first synchronization attempt between the one reservation system and the airline inventory system;

in response to the first synchronization attempt being triggered, generate a synchronization message based on the first data string;

send the synchronization message from the one reservation system to the airline inventory system;

determine that the first synchronization attempt is unsuccessful; and

in response to determining that the first synchronization attempt is unsuccessful:

store, at the one reservation system, reconciliation data that identifies data relating to the passenger name record that is not yet synchronized;

add a queue item to a first queue;

trigger one or more additional synchronization attempts based on the queue item and the reconciliation data according to a retry counter;

in response to one of the one or more additional synchronization attempts being successful, remove the reconciliation data from the one reservation system and the queue item from the first queue; and

in response to reaching a maximum number of retries:

move the queue item to a second queue for rejected items,

after the queue item is moved to the second queue for rejected items, modify the first data string on the one reservation system,

update the reconciliation data based on the modified data string,

generate a third synchronization message based on the updated reconciliation data, and

send the third synchronization message from the one reservation system to the airline inventory system.

2. The system according to claim 1, wherein said first synchronization attempt is unsuccessful because of a reason selected from the group consisting of a network error, online link unavailability, lack of confirmation response from the airline inventory system, and the airline inventory system being busy.

3. The system according to claim 1, wherein the instructions further cause the system to:

in response to determining that the first synchronization attempt is unsuccessful, place the reconciliation data in a third queue containing unsynchronized data.

4. The system according to claim 3, wherein a daemon is configured to determine if the unsynchronized data in the third queue is eligible for synchronization.

5. The system according to claim 1, wherein prior to the reconciliation data being updated, the reconciliation data includes the first data string, and the instructions cause the system to update the reconciliation data based on the modified data string by causing the system to add the modified data string to the first data string included in the reconciliation data, merge the modified data string to the first data string included in the reconciliation data, or replace the first data string included in the reconciliation data by the modified data string.

6. The system according to claim 1, wherein the instructions that cause the system to trigger one or more additional synchronization attempts based on the queue item and the reconciliation data according to a retry counter comprise instructions that cause the system to:

initiate the one or more additional synchronization attempts according to an exponential backoff algorithm.

7. A method for data synchronization between one or more reservation systems and an airline inventory system connected by a communication network, such that the one or more reservation systems are responsible for the data synchronization, the method comprising:

creating, by at least one processor, a first data string at one of the one or more reservation systems, the first data string including information relating to a creation or modification of a passenger name record on the one reservation system;

in response to the creation of the first data string, triggering, by the at least one processor, a first synchronization attempt between the one reservation system and the airline inventory system;

in response to the first synchronization attempt being triggered, generating, by the at least one processor, a first synchronization message based on the first data string;

sending, by the at least one processor, the first synchronization message from the one reservation system to the airline inventory system;

determining that the first synchronization attempt is unsuccessful; and

in response to determining that the first synchronization attempt is unsuccessful:

storing, by the at least one processor and at the one reservation system, reconciliation data that identifies data relating to the passenger name record that is not yet synchronized;

adding, by the at least one processor, a queue item to a first queue;

triggering, by the at least one processor, one or more additional synchronization attempts based on the queue item and the reconciliation data according to a retry counter;

in response to one of the one or more additional synchronization attempts being successful, removing, by the at least one processor, the reconciliation data from the one reservation system and the queue item from the first queue; and

in response to reaching a maximum number of retries:

moving, by the at least one processor, the queue item to a second queue for rejected items,

after the queue item is moved to the second queue for rejected items, modifying, by the at least one processor, the first data string on the one reservation system,

updating the reconciliation data based on the modified data string,

generate a third synchronization message based on the updated reconciliation data, and

send the third synchronization message from the one reservation system to the airline inventory system.

8. A computer program product for data synchronization between one or more reservation systems and an airline inventory system connected by a communication network, such that the one or more reservation systems are responsible for the data synchronization, the computer program product comprising:

a non-transitory computer readable storage medium; and

computer readable code stored on the storage medium and that, when executed by at least one processor, causes the at least one processor to:

create a data string at one of the one or more reservation systems, the data string including information relating to a creation or modification of a passenger name record on the one reservation system;

in response to the creation of the data string, trigger a first synchronization attempt between the one reservation system and the airline inventory system;

in response to the first synchronization attempt being triggered, generate a synchronization message based on the data string;

send the synchronization message from the one reservation system to the airline inventory system;

determine that the first synchronization attempt is unsuccessful; and

in response to determining that the first synchronization attempt is unsuccessful:

store, at the one reservation system, reconciliation data that identifies data relating to the passenger name record that is not yet synchronized,

add a queue item to a first queue;

trigger one or more additional synchronization attempts based on the queue item and the reconciliation data according to a retry counter;

in response to one of the one or more synchronization attempts being successful, remove the reconciliation data from the one reservation system and the queue item from the first queue; and

in response to reaching a maximum number of retries:

move the queue item to a second queue for rejected items,

after the queue item is moved to the second queue for rejected items, modify the first data string on the one reservation system,

update the reconciliation data based on the modified data string,

generate a third synchronization message based on the updated reconciliation data, and

send the third synchronization message from the one reservation system to the airline inventory system.

9. The method according to claim 7, wherein triggering the one or more additional synchronization attempts based on the queue item and the reconciliation data according to the retry counter comprises:

determining, by a daemon, if the queue item is eligible for synchronization; and

in response to the daemon determining that the queue item is eligible for synchronization:

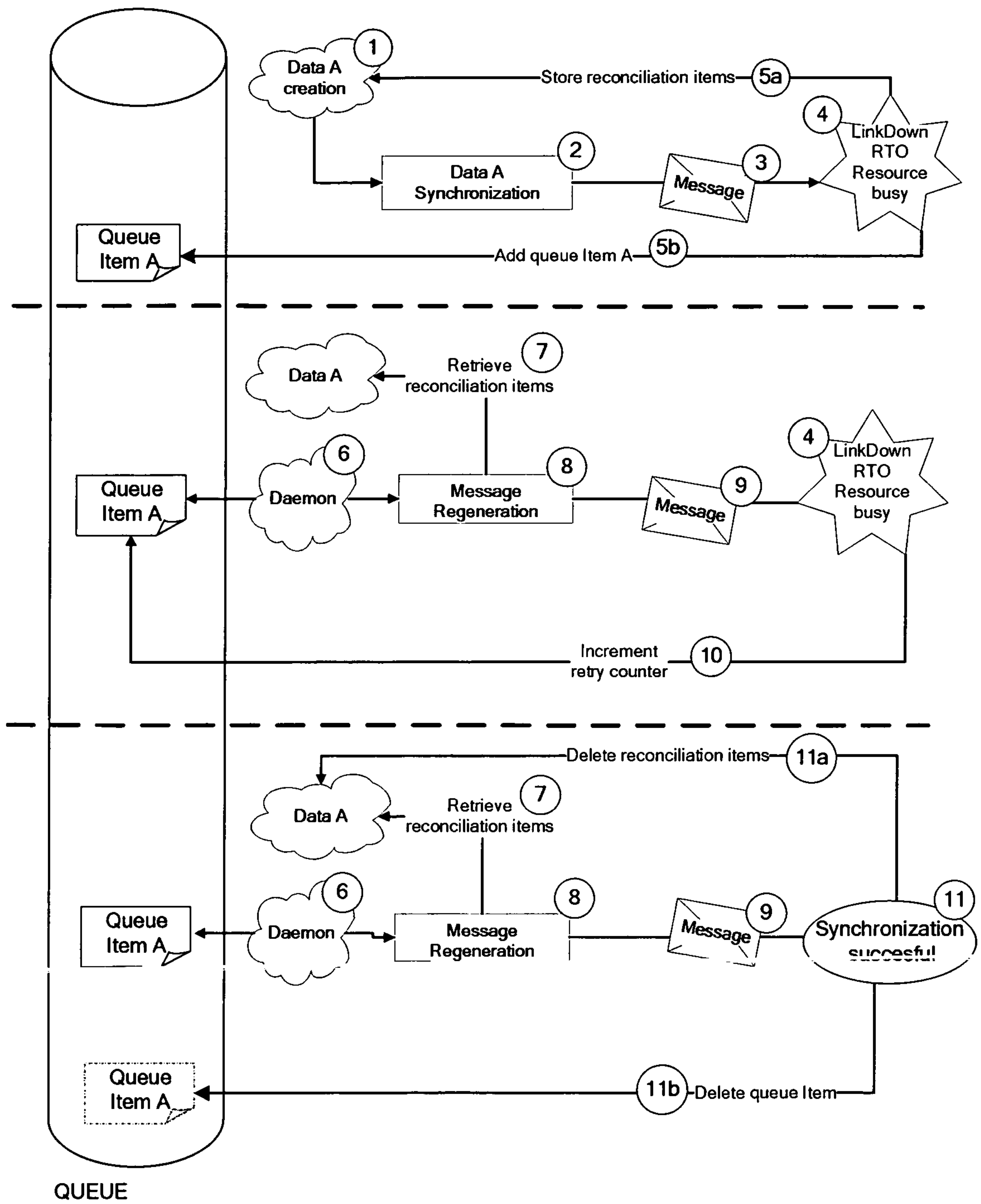
retrieving the reconciliation data;

generating a second synchronization message based on the retrieved reconciliation data;

incrementing the retry counter; and

sending the second synchronization message from the one reservation system to the airline inventory system.

1/3



5

FIG 1

2/3

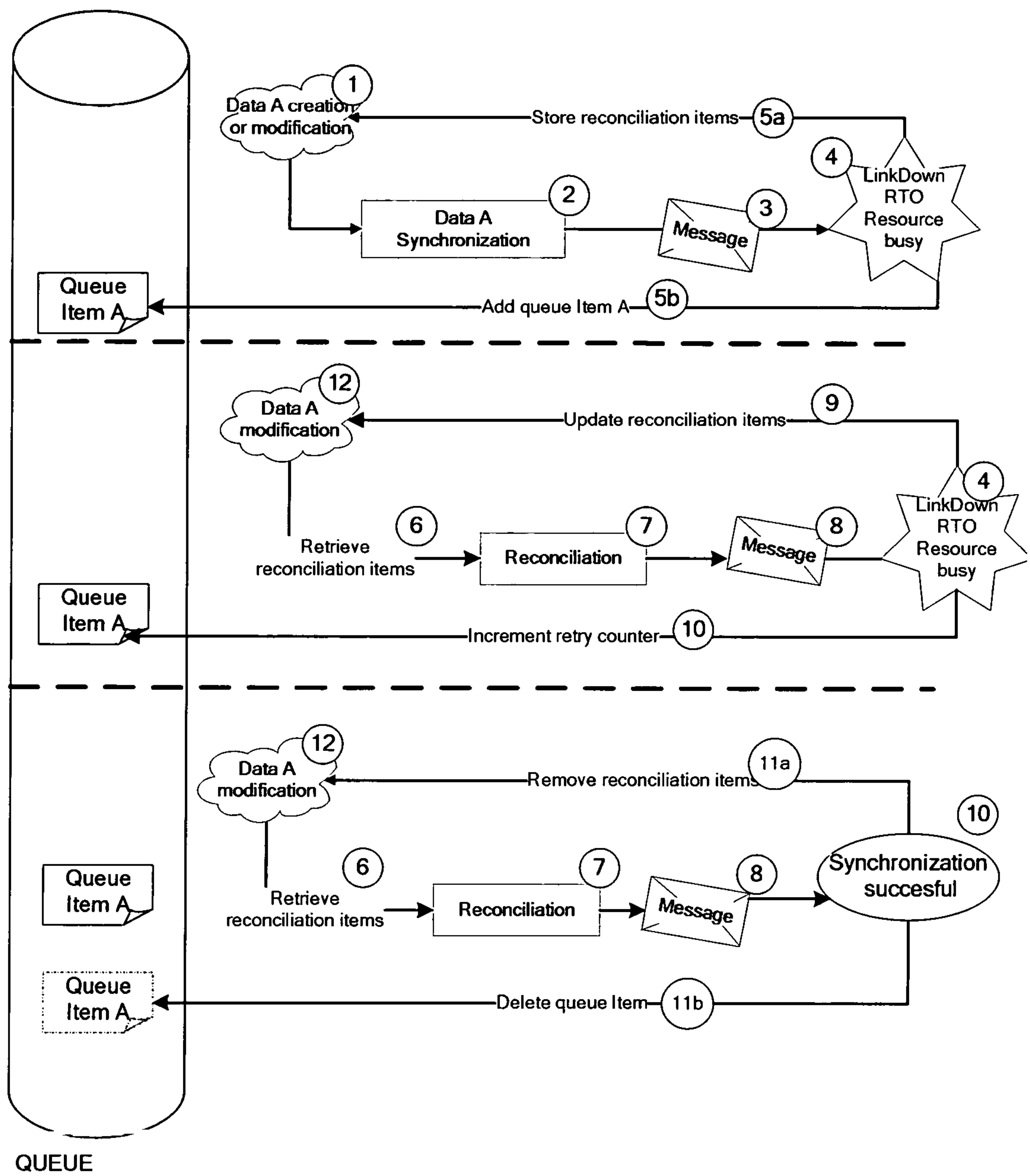


FIG 2

3/3

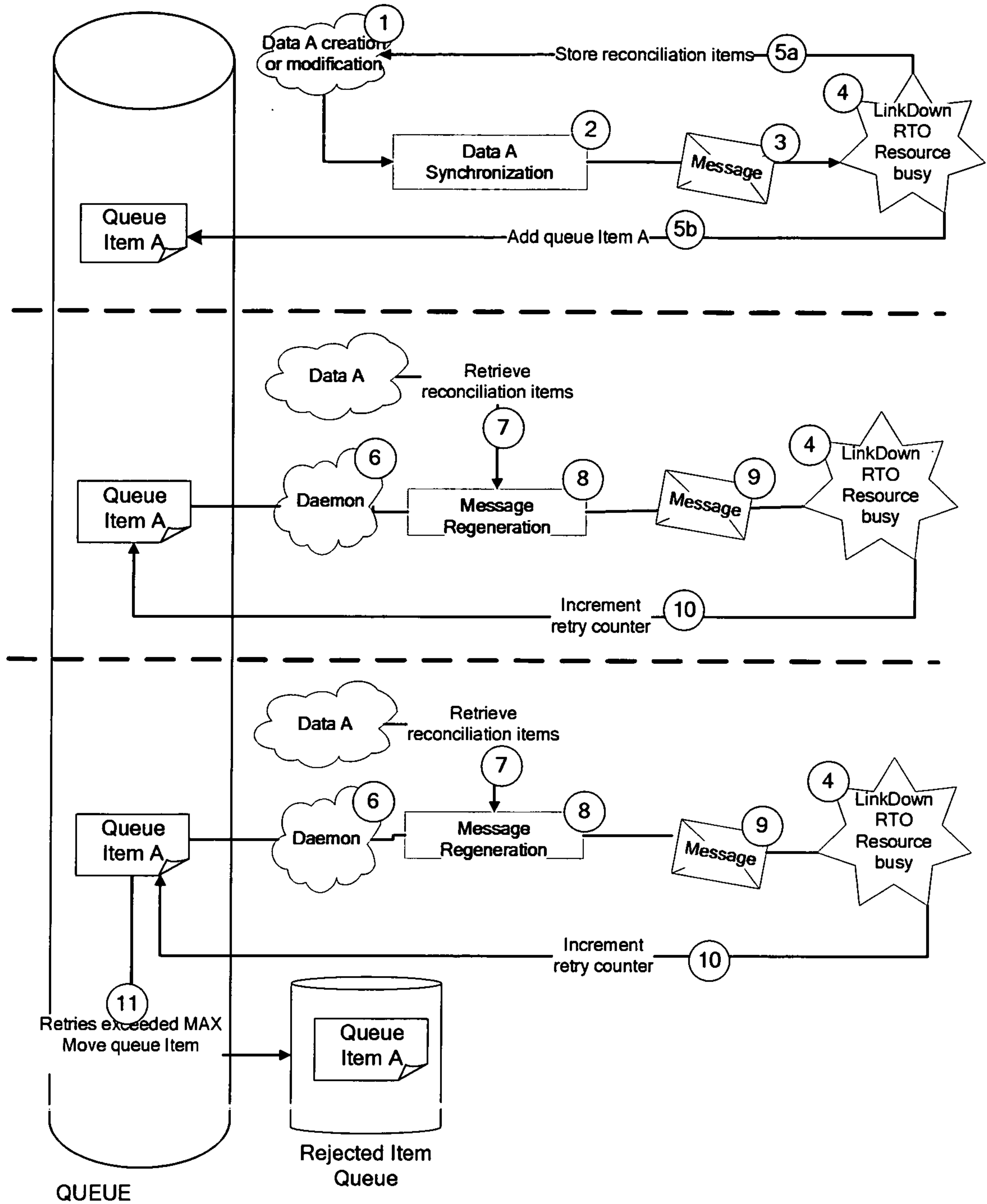


FIG 3

