

(19) World Intellectual Property Organization  
International Bureau



(43) International Publication Date  
22 May 2009 (22.05.2009)

(10) International Publication Number  
WO 2009/065049 A1

- (51) International Patent Classification:  
G06Q 50/00 (2006.01)
- (21) International Application Number:  
PCT/US2008/083654
- (22) International Filing Date:  
14 November 2008 (14.11.2008)
- (25) Filing Language: English
- (26) Publication Language: English
- (30) Priority Data:  
60/988,735 16 November 2007 (16.11.2007) US  
12/163,572 27 June 2008 (27.06.2008) US
- (71) Applicant (for all designated States except US): MICROSOFT CORPORATION [US/US]; One Microsoft Way, Redmond, Washington 98052-6399 (US).
- (72) Inventors: CHKODROV, Gueorgui; One Microsoft Way, Redmond, Washington 98052-6399 (US). HILL, Richard D.; One Microsoft Way, Redmond, Washington 98052-6399 (US). CRITCHLEY, Craig A.; One Microsoft Way, Redmond, Washington 98052-6399 (US).

SRINIVASAN, Krishnan; One Microsoft Way, Redmond, Washington 98052-6399 (US). TARNAVSKI, Tihomir; One Microsoft Way, Redmond, Washington 98052-6399 (US). MORRIS, Mitchell G.; One Microsoft Way, Redmond, Washington 98052-6399 (US). GURUNATH, Pramod; One Microsoft Way, Redmond, Washington 98052-6399 (US).

(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, 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, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH,

[Continued on next page]

(54) Title: DISTRIBUTED MESSAGING SYSTEM WITH CONFIGURABLE ASSURANCES

(57) Abstract: The present invention extends to methods, systems, and computer program products for configuring assurances within distributed messaging systems. A defined set of message log and cursor components are configurably activatable and deactivatable to compose a variety of different capture assurances, transfer assurances, and delivery assurances within a distributed messaging system. A composition of a capture assurance, a transfer assurance, and a delivery assurance can provide an end-to-end assurance for a messaging system. End-to-end assurances can include one of best effort, at-most-once, at-least-once, and exactly once and can include one of: durable or non-durable. Using a defined set of activatable and deactivatable message log and cursor components facilities more efficient transitions between desired assurances. In some embodiments, a composition of a capture assurance, a transfer assurance, and a delivery assurance provides durable exactly-once message delivery.

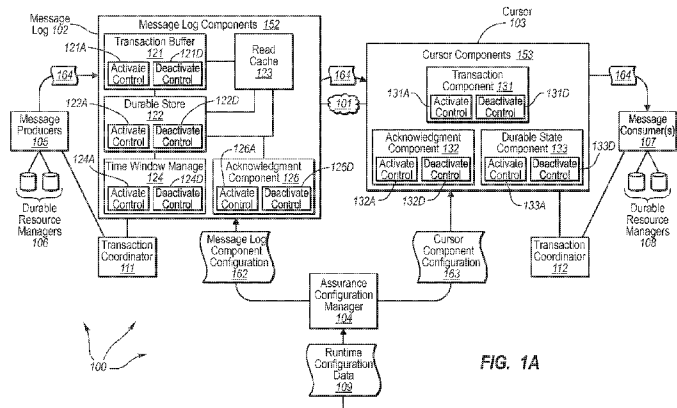


FIG. 1A

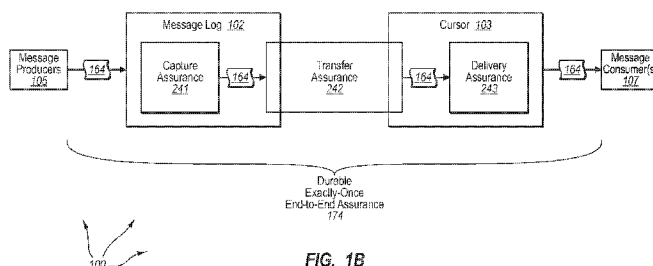


FIG. 1B

WO 2009/065049 A1



GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MT, NL, NO, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

**Published:**

- *with international search report*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

**Declarations under Rule 4.17:**

- *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*
- *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

## DISTRIBUTED MESSAGING SYSTEM WITH CONFIGURABLE ASSURANCES

### BACKGROUND

#### Background and Relevant Art

5 [0001] Computer systems and related technology affect many aspects of society. Indeed, the computer system's ability to process information has transformed the way we live and work. Computer systems now commonly perform a host of tasks (e.g., word processing, scheduling, accounting, etc.) that prior to the advent of the computer system were performed manually. More recently, computer systems have  
10 been coupled to one another and to other electronic devices to form both wired and wireless computer networks over which the computer systems and other electronic devices can transfer electronic data. Accordingly, the performance of many computing tasks are distributed across a number of different computer systems and/or a number of different computing environments.

15 [0002] Distributed messaging systems can implement queuing and publish/subscribe ("pub/sub") message patterns. A queue is essentially a one-to-one consumer message exchange. A message producer inserts a message into the queue and a message consumer takes the message out of the queue. On the other hand, a pub/sub system is a one-to-many message exchange. A message producer  
20 publishes a message and multiple message consumers subscribe to and receive the published message.

[0003] Distributed messaging systems, including those that implement queuing and pub/sub patterns, are typically backed by a message store. In these distributed messaging systems, the state of individual consumers who subscribe for messages  
25 from a topic or who consume from a queue is collocated with the message store. Consumers (e.g., user code applications) typically use distributed messaging systems to ensure that the message is delivered exactly once from the message log to the consumer. Consumers use transactions to retrieve messages because application state is inherently tied with the cursor state. Unfortunately, the use of  
30 transactions does not scale for large numbers of consumers receiving messages from a queue or topic.

[0004] Typical distributed messages systems also have a number of other limitations. For example, a distributed messages system is often tightly coupled to a limited number of, or even a single, network topology. Thus, there is limited, if any, ability to modify a distributed message system for use in other network topologies. For example, some distributed message systems cannot be used over a topology where consumers are separated from the queue or topic via a firewall.

[0005] Further, distributed messaging systems are typically limited in the message assurances they provide. For example, a distributed messaging system that provides best effort message delivery is generally not designed to ensure that no messages are ever dropped. Even distributed messaging systems that provide high assurances typically can not be modified to provide low assurances. For example, a distributed messaging system that provides exactly once assurances typically can not be easily modified to provide at most once, at least once, or best effort delivery.

[0006] Similar limitations are also associated with message durability within distributed messaging systems. That is, distributed messaging systems are typically limited in the type of message storage they provide. Further, similar to the limitations associated with assurances, even distributed messaging systems providing increased reliability of message storage can not be modified to provide lesser reliable message storage. For example, a distributed messaging system that provides durable message storage typically can not be easily modified to provide volatile or non-transaction message storage.

[0007] These assurance and message log limitations can be imposed due to the network topology of a distributed message system or can result from hard coded instructions within the software of the distributed message system. For example, a developer that designs a durable queue may have no reason (and from the perspective of his employer it may be a waste of resources) to include, or even consider, options for lesser reliable message storage. Likewise, a developer of a pub/sub messaging system with best effort delivery would waste significant resources adding options for more reliable message delivery, when the more reliable message delivery is not necessarily needed.

[0008] However, as with most computerized systems, the requirements of a distributed messaging system can change over time. For example, a distributed message system can gain more users, the types of users can change (e.g., moving from a freeware to a revenue model), government regulations can change, strategic business decisions can be made to cut costs, etc. In view of changed requirements, corresponding changes to message delivery assurances and message storage reliability may be warranted.

[0009] For example, when moving from freeware to a revenue model, customers may demand increased message delivery assurances (e.g., from best effort to exactly once). Alternately, to cut costs an IT department can determine it is more cost efficient to reduce message delivery assurances (e.g., from at least once to best effort). Another typical reason is the business application(s) running over the distributed message system may have been changed or replaced, resulting in a change of its requirements of the distributed messaging system. Similarly, the failure rate of the underlying network may have changed, making applying assurances at a higher-level less cost-efficient. Unfortunately, there is typically no efficient way to alter (either increase or decrease) existing message delivery assurances and message storage reliability in a distributed message system.

[0010] One option is to rebuild code to include desired delivery assurances and storage reliability mechanisms. However, this can be time consuming, costly and resource intensive for developers. Another option is to purchase a new distributed message system that includes the desired delivery assurance and storage reliability. However, this can also be costly and can lead to compatibility difficulties during integration with existing components and messages.

25

### **BRIEF SUMMARY**

[0011] The present invention extends to methods, systems, and computer program products for configuring assurances within distributed messaging systems. A componentized messaging system facilitates configuring any of a variety of different capture protocols, transfer protocols, and delivery protocols to interact with one another to provide desired message assurances. Accordingly, a business

30

can more efficiently configure (and change configuration of) their message system to align with (potentially varied and dynamic) business logic for the business.

[0012] In some embodiments, a componentized message system includes a message log and a cursor. Generally, the message log is configured to store messages. The message log also includes message log components for composing any of an extensible set of capture protocols to capture messages from message producers allowing the producers to hand-off a message to a distributed messaging system with different assurances and over different protocols. For example any request-response protocol can be used to achieve At-Least-Once (ALO) assurance. One way protocols, on the other hand, can only do At-Most-Once (AMO), because there is no feedback if a message is lost or a crash occurs. The extensible set of capture protocols can provide any of a variety of different message capture assurances from within a set of message capture assurances. Different message capture assurances are facilitated through the use of different combinations of message log components, including utilizing one or more of a durable store and transactions.

[0013] Generally, the cursor is configured to track the progress of consumers through messages in the message log. The cursor also includes cursor components for composing any of an extensible set of delivery protocols and/or message exchange patterns (MEPs) to deliver message to message consumers. MEPs typically identify who sends data to whom first and what the data contains (e.g. One-Way vs. Request-Reply). The delivery protocols indicate how the data appears on the wire (e.g. XML, byte-packets, etc.). The extensible set of delivery protocols can provide any of a variety of different message delivery assurances from within a set of message delivery assurances. Different message delivery assurances are facilitated through the use of different combinations of cursor components, including utilizing one or more of a durable store and transactions.

[0014] The message log and the cursor each also include further components for interoperating with one another to compose any of an extensible set of transfer protocols to transfer messages from the message log to the separate cursor. The extensible set of transfer protocols can provide any of a variety of different

message transfer assurances from within a set of message transfer assurances. Different message transfer assurances are facilitated through the use of different combinations of message log and cursor components, including utilizing one or more of a time window and varying in the use of acknowledged delivery.

5 [0015] Accordingly, various combinations of capture assurances, transfer assurances, and delivery assurances can be combined to provide different end-to-end assurances, such as, for example, being either best effort, at-most-once, at-least-once or exactly once and either durable or volatile.

[0016] Additional features and advantages of the invention will be set forth in the  
10 description which follows, and in part will be obvious from the description, or may be learned by the practice of the invention. The features and advantages of the invention may be realized and obtained by means of the instruments and combinations particularly pointed out in the appended claims. These and other features of the present invention will become more fully apparent from the  
15 following description and appended claims, or may be learned by the practice of the invention as set forth hereinafter.

#### **BRIEF DESCRIPTION OF THE DRAWINGS**

[0017] In order to describe the manner in which the above-recited and other advantages and features of the invention can be obtained, a more particular  
20 description of the invention briefly described above will be rendered by reference to specific embodiments thereof which are illustrated in the appended drawings. Understanding that these drawings depict only typical embodiments of the invention and are not therefore to be considered to be limiting of its scope, the invention will be described and explained with additional specificity and detail  
25 through the use of the accompanying drawings in which:

[0018] Figure 1A illustrates an example computer architecture that facilitates configuring assurances within a distributed messaging system.

[0019] Figure 1B illustrates an example computer architecture configured to provide durable exactly-once end-to-end message transfer assurance with a  
30 distributed messaging system.

[0020] Figure 2 illustrates various matrices of assurances that can be configurably composed for a distributed messaging system.

[0021] Figure 3 illustrates a flow chart of an example method for configuring message assurances within a distributed message system.

5

### **DETAILED DESCRIPTION**

[0022] The present invention extends to methods, systems, and computer program products for configuring assurances within distributed messaging systems. A componentized messaging system facilitates configuring any of a variety of different capture protocols, transfer protocols, and delivery protocols to interact with one another to provide desired message assurances. Accordingly, a business can more efficiently configure (and change configuration of) their message system to align with (potentially varied and dynamic) business logic for the business.

[0023] In some embodiments, a componentized message system includes a message log and a cursor. Generally, the message log is configured to store messages. The message log also includes message log components for composing any of an extensible set of capture protocols to capture messages from message producers. The extensible set of capture protocols can provide any of a variety of different message capture assurances from within a set of message capture assurances. Different message capture assurances are facilitated through the use of different combinations of message log components, including utilizing one or more of a durable store and transactions.

[0024] Generally, the cursor is configured to track the progress of consumers through messages in the message log. The cursor also includes cursor components for composing any of an extensible set of delivery protocols to deliver message to message consumers. The extensible set of delivery protocols can provide any of a variety of different message delivery assurances from within a set of message delivery assurances. Different message delivery assurances are facilitated through the use of different combinations of cursor components, including utilizing one or more of a durable store and transactions.

[0025] The message log and the cursor each also include further components for interoperating with one another to compose any of an extensible set of transfer



protocols to transfer messages from the message log to the separate cursor. The extensible set of transfer protocols can provide any of a variety of different message transfer assurances from within a set of message transfer assurances.

Different message transfer assurances are facilitated through the use of different combinations of message log and cursor components, including utilizing one or more of a time window and varying in the use of acknowledged delivery.

5 [0026] Accordingly, various combinations of capture assurances, transfer assurances, and delivery assurances can be combined to provide different end-to-end assurances, such as, for example, best effort, at-most-once, at-least-once and exactly once.

[0027] Embodiments of the present invention may comprise or utilize a special purpose or general-purpose computer including computer hardware, as discussed in greater detail below. Embodiments within the scope of the present invention also include physical and other computer-readable media for carrying or storing computer-executable instructions and/or data structures. Such computer-readable media can be any available media that can be accessed by a general purpose or special purpose computer system. Computer-readable media that store computer-executable instructions are physical storage media. Computer-readable media that carry computer-executable instructions are transmission media. Thus, by way of example, and not limitation, embodiments of the invention can comprise at least two distinctly different kinds of computer-readable media: physical storage media and transmission media.

[0028] Physical storage media includes RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage or other magnetic storage devices, or any other medium which can be used to store desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer.

[0029] A “network” is defined as one or more data links that enable the transfer of electronic data between computer systems and/or modules and/or other electronic devices. When information is transferred or provided over a network or another communications connection (either hardwired, wireless, or a combination of

hardwired or wireless) to a computer, the computer properly views the connection as a transmission medium. Transmission media can include a network and/or data links which can be used to carry desired program code means in the form of computer-executable instructions or data structures and which can be accessed by a general purpose or special purpose computer. Combinations of the above should also be included within the scope of computer-readable media.

[0030] Further, upon reaching various computer system components, program code means in the form of computer-executable instructions or data structures can be transferred automatically from transmission media to physical storage media (or vice versa). For example, computer-executable instructions or data structures received over a network or data link can be buffered in RAM within a network interface module (e.g., a “NIC”), and then eventually transferred to computer system RAM and/or to less volatile physical storage media at a computer system. Thus, it should be understood that physical storage media can be included in computer system components that also (or even primarily) utilize transmission media.

[0031] Computer-executable instructions comprise, for example, instructions and data which cause a general purpose computer, special purpose computer, or special purpose processing device to perform a certain function or group of functions. The computer executable instructions may be, for example, binaries, intermediate format instructions such as assembly language, or even source code. Although the subject matter has been described in language specific to structural features and/or methodological acts, it is to be understood that the subject matter defined in the appended claims is not necessarily limited to the described features or acts described above. Rather, the described features and acts are disclosed as example forms of implementing the claims.

[0032] Those skilled in the art will appreciate that the invention may be practiced in network computing environments with many types of computer system configurations, including, personal computers, desktop computers, laptop computers, message processors, hand-held devices, multi-processor systems, microprocessor-based or programmable consumer electronics, network PCs,

minicomputers, mainframe computers, mobile telephones, PDAs, pagers, routers, switches, and the like. The invention may also be practiced in distributed system environments where local and remote computer systems, which are linked (either by hardwired data links, wireless data links, or by a combination of hardwired and wireless data links) through a network, both perform tasks. In a distributed system environment, program modules may be located in both local and remote memory storage devices.

[0033] Figure 1A illustrates an example computer architecture 100 that facilitates configuring assurances within a distributed messaging system. As depicted, computer architecture 100 includes message producers 105, durable resource managers 106, message log 102, cursor 103, durable resource managers 108, message consumer(s) 107, and assurance configuration manager 104. Each of the depicted computer systems can be connected to one another over (or be part of) a network, such as, for example, a Local Area Network ("LAN"), a Wide Area Network ("WAN"), and even the Internet. Accordingly, each of the depicted components as well as any other connected components, can create message related data and exchange message related data (e.g., Internet Protocol ("IP") datagrams and other higher layer protocols that utilize IP datagrams, such as, Transmission Control Protocol ("TCP"), Hypertext Transfer Protocol ("HTTP"), Simple Mail Transfer Protocol ("SMTP"), etc.) over the network.

[0034] Generally, message producers 105 send messages (e.g., message 164) to message log 102. The messages are based on resources from durable resources 106. Message log 102 invokes desired messaging behavior to store messages (with essentially no knowledge of potential message consumers), such as, for example, functioning as a queue or publisher in a pub/sub environment. When appropriate, messages (e.g., message 164) are sent from message log 102 over network 101 to cursor 103. Cursor 103 then invokes desired messaging behavior, such as, for example, receiving a queued message or functioning as a subscriber in a pub/sub environment. Cursor 103 then sends messages (e.g., message 164) to message consumer(s) 107, which can store the messages in durable resource managers 108 or perform other operations with message content. Cursor 103 can store the

progress of a message consumer or group of message consumers in a message sequence.

[0035] Message log 102 and cursor 103 each include activatable and deactivatable components that can be activated and deactivated in various different combinations to compose desired message assurances or portions thereof. For example, message  
5 log components 152 include transaction buffer 121, read cache 123, durable store 122, time window manager 124, and acknowledgement component 126. Generally, message log components 152 can be activated and deactivated to compose any of an extensible set of capture protocols. Each capture protocol can be configured to  
10 provide a different capture assurance. For example, capture protocols can vary in the use of durable store 122 and participation in transactions involving transaction coordinator 111 to provide different capture assurances.

[0036] When activated, transaction buffer 121 keeps track of the messages for transactions that are in process. Transaction buffer 121 can enlist in a transaction  
15 as a volatile resource manager with the option to enlist as a durable resource manager during prepare (“EDPR”). When a message producer initiates commit, the transaction buffer receives prepare notification. Transaction buffer 121 can then enlist durable store 122 in the transaction (when activated) and save the message. When transaction buffer 121 receives committed it moves the messages  
20 to the read cache 123. Thus a message can be available on a short term bases without the need to read back from the Durable Store.

[0037] Similarly, cursor components 153 include transaction component 131, acknowledgement component 132, and durable store component 133. Generally, cursor components 153 can be activated and deactivated to compose any of an  
25 extensible set of delivery protocols. Each delivery protocol can be configured to provide a different delivery assurance. For example, delivery protocols can vary in the use of transaction component 131 and durable state component 133 to provide different delivery assurances. When activated, transaction component 131 and durable state component 133 function similarly to transaction buffer 121 and  
30 durable store 122 respectively to delivery a message to message consumer(s) 107.

[0038] Further, various message log components 152 and cursor components 153 can be activated and deactivated and interoperate to compose any of an extensible set of transfer protocols. Each transfer protocol can be configured to provide a different transfer assurance. For example, transfer protocols can vary in the used of acknowledgment components 126 and 132 and time window manager 124 to provide different transfer assurances.

[0039] For example, when activated, time window manager 124 can cause durable store 122 to keep a given window of messages (e.g., one day). Acknowledgement components 126 and 132 can interoperate to provide message acknowledgements from cursor 103 to message log 102. For example, when a message or range of messages is received at cursor 103, acknowledgement component 133 can acknowledge receipt of the message or range of messages to acknowledgement components 126. Upon receiving acknowledgment, appropriate messages can be discarded from message log 102. When activated, this interoperation causes durable store 122 to keep messages until acknowledged. Thus, there is an assurance that once a subscription is created a consumer is not likely to miss any messages.

[0040] Generally, assurance configuration manager 104 can send commands to message log 102 and cursor 103 to activate and deactivate appropriate components of message log components 152 and cursor components 153. Activation and deactivation of appropriate message log components 152 and cursor components 153 can result in composition of desired capture, transfer, and delivery protocols. The desired capture, transfer, and delivery protocols can then interoperate to provide desired end-to-end message assurance.

[0041] In some embodiments, assurance configuration manager 104 receives (e.g., system or other low-level) runtime configuration data 109. Runtime configuration data 109 indicates a combination of a capture assurance, a transfer assurance, and a delivery assurance that are to provide a desired end-to-end message assurance for message transfer in computer architecture 100.

[0042] From runtime configuration data 109, assurance configuration manager 104 can formulate a message log component configuration. A message log

component configuration is for activating and deactivating appropriate message log components to compose the capture assurance and a first portion of the transfer assurance. For example, message log component configuration 162 is for activating and deactivating transaction buffer 121, durable store 122, time window manager 125, and acknowledgement component 126 to compose the capture assurance and a first portion of the transfer assurance that can interoperate with cursor components 153.

[0043] From runtime configuration data 109, assurance configuration manager 104 can also formulate a cursor component configuration. A cursor component configuration is for activating and deactivating appropriate cursor components to compose the delivery assurance and a second portion of the transfer assurance. For example, cursor component configuration 163 is for activating and deactivating transaction component 131, durable state component 133, and acknowledgement component 132 to compose the delivery assurance and the second portion of the transfer assurance that can interoperate with message log components 152.

[0044] Assurance configuration manager 104 can instruct components of computer architecture 100 to compose the capture assurance, the transfer assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance. Assurance configuration manager 104 can send message log component configuration 162 to message log 102 to instruct message log 102 to compose the capture assurance and the first portion of the transfer assurance. Similarly, assurance configuration manager 104 can send cursor component configuration 163 to cursor 103 to instruct cursor 103 to compose the delivery assurance and the second portion of the transfer assurance.

[0045] As previous described, message log components and cursor components can be activate and deactivate to provide a desired end-to-end message assurance. In some embodiments, each component is individually activatable and deactivatable. For example, each component can include an activation control and a deactivation control. For example, at message log 102, transaction buffer 121 includes activation control 121A and deactivation control 121D, durable store 122 includes activation control 122A and deactivation control 122D, time window

manager includes activation control 124A and deactivation control 124D, and acknowledgement component 126 includes activation control 126A and deactivation control 126D. Similarly, at cursor 103, transaction component 131 includes activation control 131A and deactivation control 131D, durable state component 133 includes activation control 133A and deactivation control 133D, and acknowledgement component 132 includes activation control 132A and deactivation control 132D. Individually activatable and deactivatable components provide a high degree of granularity. This granularity facilitates configuration of a wide variety of different sets of message capture assurances, message transfer assurances, and message delivery assurances.

[0046] As such, message log component configuration 162 can include instructions for individually activating and deactivating message log components 102. Similarly, cursor component configuration 163 can include instructions for individually activating and deactivating cursor components 103. In combination, message log component configuration 162 and cursor component configuration 163 instruct message log 102 and cursor 103 to provide desired end-to-end message assurance.

[0047] Figure 2 illustrates various matrices 200 of assurances that can be configurably composed for a distributed messaging system. Matrices 200 will be described with reference to the components of computer architecture 100 in Figure 1A.

[0048] Capture assurances matrix 201 represents an example of the properties of various different capture assurances message store 102 can use to capture messages from message producers 105. Within capture assurances matrix 201, each capture assurance indicates assurance properties that the capture assurance does or does not utilize. Each capture assurance can be implemented through the activation and deactivation of different combinations of message log components 152.

[0049] For example, capture assurance 211 does not utilize a durable store nor transactions. Thus, sending a message does not care about transactions and consumers can see messages despite commit/abort status of transactions. Further messages are lost when a crash or power failure occurs. Within computer

architecture 100, capture assurance 211 can be implemented by deactivating durable store 122 and deactivating transaction buffer 121.

[0050] Capture assurance 221 does not utilize a durable store but does utilize transactions. Thus, consumers see messages for committed transactions (e.g.,  
5 coordinated by transaction coordinator 111). However, messages are lost when a crash or power failure occurs. Within computer architecture 100, capture assurance 221 can be implemented by deactivating durable store 122 and activating transaction buffer 121.

[0051] Capture assurance 231 utilizes a durable store but does not utilize  
10 transactions. Thus, sending a message does not care about transactions and consumers can see messages despite commit/abort status of transactions. However, messages are durably stored (e.g., on disk) so that messages are not lost when a crash or power failure occurs. Within computer architecture 100, capture assurance 231 can be implemented by activating durable store 122 and deactivating  
15 transaction buffer 121.

[0052] Capture assurance 241 utilizes a durable store and transactions. Thus, consumers see messages for committed transactions and messages are not lost in case of a crash or power failure. Within computer architecture 100, capture assurance 241 can be implemented by activating durable store 122 and activating  
20 transaction buffer 121.

[0053] Transfer assurances matrix 202 represents an example of the properties of various different transfer assurances that can be used to transfer a message from message store 102 to cursor 103. Within transfer assurances matrix 202, each transfer assurance indicates assurance properties that the transfer assurance does or  
25 does not utilize. Each transfer assurance can be implemented through the activation and deactivation of different combinations of message log components 152 and cursor components 153.

[0054] For example, transfer assurance 212 does not utilize acknowledged delivery nor a time window. Transfer assurance 212 essentially represents best  
30 effort delivery. Within computer architecture 100, transfer assurance 212 can be



implemented by deactivating time window manager 124 and deactivating acknowledgment components 126 and 132.

5 [0055] Transfer assurance 222 does utilize acknowledged delivery but does not utilize a time window. Transfer assurance 222 keeps message until acknowledged (potentially forever). Transfer assurance 222 provides assurance that once a subscription is created, a consumer is unlikely to miss a message. However, since no time window is utilized, a subscription can not obtain older messages. Within computer architecture 100, transfer assurance 222 can be implemented by deactivating time window manager 124 and activating acknowledgment components 126 and 132.

10 [0056] Transfer assurance 232 does not utilize acknowledged delivery but does utilize a time window. Transfer assurance 232 guarantees to keep given time window of messages (e.g. One Day). Thus, existing subscribers are not likely to miss messages unless they fall behind the online-window. New subscribers have a substantially high likelihood of receiving a given range of past messages. Within computer architecture 100, transfer assurance 232 can be implemented by activating time window manager 124 and deactivating acknowledgment components 126 and 132.

20 [0057] Transfer assurance 242 utilizes acknowledged delivery and utilizes a time window. Transfer assurance 242 is essentially a logical AND of transfer assurance 222 and transfer assurance 232. Thus, existing subscribers are not likely to miss messages and new subscribers have a substantial likelihood of receiving a given range of past messages. Within computer architecture 100, transfer assurance 242 can be implemented by activating time window manager 124 and activating acknowledgment components 126 and 132.

25 [0058] Depending on the retention policy, message log 102 can behave differently when a quota is reached. For example, when utilizing transfer assurance 212, message log 102 can simply drop messages when a quota is reached. Message log 102 does not send error messages to message producers 105 when a message is dropped. As such, message producers 105 experience little if any delays.

30

[0059] When utilizing transfer assurances 222, 232, or 242, message log 102 can delay sending responses to producers to give further time for consumers to drain messages. Thus, message log 102 can compensate to some extent to avoid timeouts and message retries. However, message log 102 may eventually hit a timeout  
5 causing message producers to receive an error and possibly retry.

[0060] Delivery assurances matrix 203 represents an example of the properties of various different delivery assurances that can be used to transfer a message from cursor 103 to message consumer(s) 107. Within delivery assurances matrix 203, each delivery assurance indicates assurance properties that the delivery assurance  
10 does or does not utilize. Each delivery assurance can be implemented through the activation and deactivation of different combinations of cursor components 153.

[0061] For example, delivery assurance 213 does not utilize durable state nor transactions. Delivery assurance 213 essentially represents publish/subscribe functionality. Within computer architecture 100, delivery assurance 213 can be  
15 implemented by deactivating durable state component 133 and transaction component 131.

[0062] Delivery assurance 223 does not utilize durable state but does utilize transactions. Thus, consumers use volatile transactions as a mechanism to enforce atomic changes. Cursor 103 participates in transactions (e.g., coordinated by  
20 transaction coordinator 112) and provides exactly-once delivery into the state that is committed. Within computer architecture 100, delivery assurance 223 can be implemented by deactivating durable state component 133 and activating transaction component 131.

[0063] Delivery assurance 233 utilizes a durable state but does not utilize  
25 transactions. Thus, cursor 103 is durable but message consumer(s) 107 do not use transactions. The properties of the provided assurance can depend on the usage pattern (protocols) implemented between durable state component 133 and message consumers(s) 107. For example, a first usage pattern includes using a locking protocol to lock messages (and thus prevents other consumers from accessing the  
30 message) and an acknowledge protocol to inform the cursor that delivery has completed. In the first usage pattern, the consumer (107) utilizes the

acknowledgment protocol as soon as it has received the message but before it has processed it. Subsequently, after message acknowledgement, the first usage pattern includes doing work based on the message. Message consumer(s) 107 using the first usage pattern are provided with At-Most-Once message delivery.

5 [0064] A second usage pattern includes using a locking protocol to lock messages and an acknowledge protocol to inform the cursor that delivery has completed. In the second usage pattern, the consumer (107), after doing work based on the message, includes using the acknowledgment protocol to inform the cursor that it has received the message. Message consumer(s) 107 using the second usage  
10 pattern are provided with At-Least-Once message delivery.

[0065] Within computer architecture 100, delivery assurance 233 can be implemented by activating durable state component 133 and deactivating transaction component 131. Different usage patterns (including the first and second usage patterns previously described) can provide delivery assurance 233 with  
15 different properties, such as, for example, At-Most-Once and At-Least-Once message delivery. In any event, delivery assurance 233 provides durable message state and recovery from a crash or power failure is likely.

[0066] Delivery assurance 243 utilizes durable state and transactions. Accordingly, delivery assurance 243 provides durable Exactly-Once delivery to  
20 message consumer(s) 107. Within computer architecture 100, delivery assurance 243 can be implemented by activating durable state component 133 and activating transaction component 131.

[0067] As previously described, various assurances can be expressed in (e.g., low-level) runtime configuration data. While possible, the runtime configuration data  
25 does not necessarily represent end-user intent. Since many end-users will lack a full understanding of the different assurances, a higher level model can provide protection from inefficient (e.g., wasted resources) or otherwise inappropriate combinations of assurances. For example, the higher level model can prevent the combination of a capture assurance that provides durable, transaction capture but a  
30 transfer assurance that is best-effort. The higher level model can utilize runtime

configuration data for selecting appropriate combinations of assurances to provide an end-to-end assurance.

[0068] Figure 3 illustrates a flow chart of an example method 300 configuring message assurances within a distributed message system. The method 300 will be described with respect to the components and data depicted in computer architecture 100 in Figure 1A.

[0069] Method 300 includes an act of receiving runtime configuration data indicating a combination of a capture assurance, a transfer assurance, and a delivery assurance that are to provide a desired end-to-end message assurance for message transfer in the distributed messaging system (act 301). For example, assurance configuration manger 104 can receive runtime configuration data 109. Runtime configuration data 109 can indicate a capture assurance, a transfer assurance, and a delivery assurance that are to provide a desired end-to-end message assurance for message transfer in computer architecture 100.

[0070] Method 300 includes an act of formulating a message log component configuration from the runtime configuration data, the message log component configuration for activating and deactivating appropriate message log components to compose the capture assurance and to compose a first portion of the transfer assurance to interoperate with a corresponding second portion of the transfer assurance provided by cursor components (act 302). For example, assurance configuration manager 104 can formulate message log component configuration 162. Message log component configuration 162 can include instructions for activating and deactivating appropriate message log components 152 to compose the capture assurance. Message log component configuration 162 can also include instructions for activating and deactivating appropriate message log components 152 to compose a first portion of the transfer assurance to interoperate with a corresponding second portion of the transfer assurance provided by cursor components 103.

[0071] Method 300 includes an act of formulating a cursor component configuration from the runtime configuration data, the cursor component configuration for activating and deactivating appropriate cursor components to

compose the delivery assurance and to compose the corresponding second portion of the transfer assurance to interoperate with the first portion of the transfer assurance provided by the message log components (act 303). For example, assurance configuration manager 104 can formulate cursor component configuration 163. Cursor component configuration 163 can include instructions for activating and deactivating appropriate cursor components 153 to compose the delivery assurance assurance. Cursor component configuration 163 can also include instructions for activating and deactivating appropriate cursor components 153 to compose the second portion of the transfer assurance to interoperate with the first portion of the transfer assurance provided by message log components 102.

[0072] Method 300 includes an act of instructing the distributed messaging system to compose the capture assurance, the transfer assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance for the distributed messaging system (act 304). For example, assurance configuration manger can instruct computer architecture 100 to compose the capture assurance, the transfer assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance for computer architecture 100.

[0073] Act 304 can include an act of sending the message log component configuration to the message log to instruct the message log to compose the capture assurance and the first portion of transfer assurance (act 305). For example, assurance configuration manager 104 can send message log component configuration 162 to message log 102 to instruct message log 102 to compose the capture assurance and the first portion of transfer assurance for computer architecture 100. Act 304 can also include an act of sending the cursor component configuration to the cursor to instruct the cursor to compose the delivery assurance and the second portion of the transfer assurance (act 306). For example, assurance configuration manager 104 can send cursor component configuration 163 to cursor 103 to instruct cursor 103 to compose the delivery assurance and the second portion of transfer assurance for computer architecture 100.

[0074] Message log 102 and cursor 103 can implement the instructions of message log component configuration 162 and cursor component configuration 163 respectively. From the instructions, message log 102 and cursor 103 can activate and deactivate appropriate message log components 152 and cursor components 5 153 respectively to compose the capture assurance, the transfer assurance, and the delivery assurance. Together, the capture assurance, the transfer assurance, and the delivery assurances provide the desired end-to-end message assurance for computer architecture 100.

[0075] Referring now to Figure 1B, Figure 1B illustrates computer architecture 10 100 configured to provide durable exactly-once end-to-end message transfer assurance. The assurances depicted in Figure 1B can result from applying method 300 within computer architecture 100. As depicted, message log 102 composes capture assurance 241 to provide durable transaction-based message capture from message producers 105. Message log 102 and cursor 103 interoperate to compose 15 transfer assurance 242 to provide time window retention and message acknowledgment transfer between message log 102 and cursor 103. Cursor 103 composes delivery assurance 243 to provide durable transaction based message delivery to message consumer(s) 107. Together, the composition of capture assurance 241, transfer assurance 242, and delivery assurance 243 provide durable 20 exactly-once end-to-end assurance 174.

[0076] As such, a message from message producers 105 (e.g., message 164) is essentially guaranteed to be received exactly once at message consumer(s) 107 even in the event of system crashes and/or power failures.

[0077] Accordingly, embodiments of the invention facilitate the configuration of a 25 variety of different messaging assurances using a defined set of messaging components that are activatable and deactivatable. Embodiments of the invention can thus be implemented to simulate queues and publish/subscribe messaging systems having any of the variety of different messaging assurances. Since a defined set of components is used, configuring assurances and changing the 30 configuration of assurances within of a messaging system is significantly more efficient.

[0078] Thus, embodiments of the present invention permit a business owner to balance the need for various message assurances against the cost to provide such message assurances. For example, resources can be conserved by using best effort or at-most once delivery when business rules do not require that a consumer receive every message (e.g., streaming media). Further, when business rules for a business change messaging systems can be more easily transitioned to comply with the new business rules. For example, to conserve resources, a messaging system can be transitioned from some other type of delivery assurance to best effort. On the other hand, if business rules transition to a revenue model, assurances can correspondingly be transitioned to insure that consumers (e.g., paying customers) get a message at least once.

[0079] The present invention may be embodied in other specific forms without departing from its spirit or essential characteristics. The described embodiments are to be considered in all respects only as illustrative and not restrictive. The scope of the invention is, therefore, indicated by the appended claims rather than by the foregoing description. All changes which come within the meaning and range of equivalency of the claims are to be embraced within their scope.

## CLAIMS

What is claimed:

1. A componentized messaging system (100) with configurable assurances, comprising:

5 a message log (102) configured to store messages, the message log including one or more processors, system memory, and message log components (152) for composing any of an extensible set of capture protocols to capture messages (164) from message producers (105), the extensible set of capture protocols configured to provide any of a variety of different message capture assurances from within a set  
10 of message capture assurances (201), the set of message capture assurances varying in the use of a durable store (122) and varying in the use of transactions;

a cursor (103) configured to track the progress of consumers (107) through messages (164) in the message log (102), the cursor (103) including one or more processors, system memory, and cursor components (153) for composing any of an  
15 extensible set of delivery protocols to deliver messages (164) to message consumers (107), the extensible set of delivery protocols configured to provide any of a variety of different message delivery assurances from within a set of message delivery assurances (203), the set of message delivery assurances varying in the use of a durable store (122) and varying in the use of transactions; and

20 wherein the message log (102) and the cursor (103) each also include further components for interoperating with one another to compose any of an extensible set of transfer protocols to transfer messages from the message log (102) to the separate cursor (103), the extensible set of transfer protocols configured to provide any of a variety of different message transfer assurances from within a set of  
25 message transfer assurances (202), the set of message transfer assurances varying in the use a time window and varying in the use of acknowledged delivery.

2. The system as recited in claim 1, wherein the message log components for implementing an extensible set of capture protocols include:

a transaction buffer, the transaction buffer configurably activatable and  
30 deactivatable to assist in composing a selected capture protocol, from among the set of extensible capture protocols, the transaction buffer activated to provide



transactional behavior to a selected capture protocol and the transaction buffer deactivated to provide non-transactional behavior to a selected capture protocol; and

5 a durable store, the durable store configurably activatable and deactivatable to assist in implementing a selected capture protocol, from among the set of extensible capture protocols, the durable store activated to provide durable storage to a selected capture protocol and the durable store deactivated to provide volatile storage to a selected capture protocol.

3. The system as recited in claim 2, wherein one of the transaction buffer and durable store is activated and one of the transaction buffer and durable store is deactivated.

4. The system as recited in claim 1, wherein the cursor components for implementing an extensible set of delivery protocols include:

15 a transaction component, the transaction component configurably activatable and deactivatable to assist in composing a selected delivery protocol, from among the set of extensible delivery protocols, the transaction component activated to provide transactional behavior to a selected delivery protocol and the transaction component deactivated to provide non-transactional behavior to a selected delivery protocol; and

20 a durable state component, the durable state component configurably activatable and deactivatable to assist in implementing a selected delivery protocol, from among the set of extensible delivery protocols, the durable state component activated to provide durable cursor state to a selected delivery protocol and the durable state component deactivated to provide volatile cursor state to a selected delivery protocol.

5. The system as recited in claim 4, wherein one of the transaction component and durable state component is activated and one of the transaction component and durable state component is deactivated.

6. The system as recited in claim 1, wherein the message log and the separate cursor components for interoperating with one another to implement an extensible set of transfer protocols include:

30

message acknowledgment components, the message acknowledgement components configurably activatable and deactivatable to assist in implementing a selected transfer protocol, from among the set of extensible transfer protocols, the message acknowledgment components activated to provide message  
5 acknowledgment behavior for a selected transfer protocol and the message acknowledgement components deactivated to prevent message acknowledgement behavior for a selected transfer protocol; and

time window components, the time window components configurably activatable and deactivatable to assist in implementing a selected transfer protocol,  
10 from among the set of extensible transfer protocols, the time window components activated to provide retention of a window of messages for a selected transfer protocol and the time window components deactivated to prevent retention of a window of messages for a selected transfer protocol.

7. The system as recited in claim 6, wherein one of the message  
15 acknowledgment components and time window components is activated and one of the message acknowledgment components and time window components is deactivated.

8. The system as recited in claim 1, wherein the message log components and cursor components are configurably activated and deactivated to provide one of:  
20 best effort message delivery, at-most-once message delivery, at-least-once message delivery, and exactly-once message delivery.

9. The system as recited in claim 1, wherein the message log components and cursor components are configurably activated and deactivated to provide one of:  
durable message delivery or volatile message delivery.

25 10. The system as recited in claim 1, wherein the message log components and cursor components are configurably activated and deactivated to provide durable exactly-once message delivery.

11. A componentized messaging system with configurable assurances, the componentized messaging system configured to provide durable, exactly once  
30 delivery of messages to a plurality of clients, the system comprising:

a message log, the message log including one or more processors, system memory, and configurable message storage components for implementing any of extensible set of capture protocols to capture messages from message producers, the message log components configured to implement a durable transaction-based capture protocol for capturing messages from message producers;

a separate client-side cursor, the separate client-side cursor separate from the message log, the separate cursor including one or more processors, system memory, and cursor components for implementing an extensible set of delivery protocols to deliver messages to message consumers, the cursor components configured to implement a durable transaction-based delivery protocol for delivering messages to message consumers; and

wherein the message log and the separate cursor each also include components for interoperating with one another to implement an extensible set of transfer protocols to transfer messages from the message log to the separate cursor, the interoperating components configured to implement a transfer protocol providing message acknowledgment behavior and retention of a window of messages.

12. The system as recited in claim 11, wherein the message storage components for implementing the durable transaction-based capture protocol include:

a transaction buffer, the transaction buffer configurably activatable and deactivatable to assist in implementing a selected capture protocol, from among the set of extensible capture protocols, the transaction buffer activated to provide transactional behavior for the durable transaction-based capture protocol; and

a durable store, the durable store configurably activatable and deactivatable to assist in implementing a selected capture protocol, from among the set of extensible capture protocols, the durable store activated to provide durable storage for the durable transaction-based capture protocol.

13. The system as recited in claim 11, wherein the separate client-side components for implementing the durable transaction-based delivery protocol include:

a transaction component, the transaction component configurably activatable and deactivatable to assist in implementing a selected delivery protocol, from among the set of extensible delivery protocols, the transaction component activated to provide transactional behavior to the durable transaction-based delivery protocol;  
5 and

a durable state component, the durable state component configurably activatable and deactivatable to assist in implementing a selected delivery protocol, from among the set of extensible delivery protocols, the durable state component activated to provide durable cursor state to the durable transaction-based delivery  
10 protocol.

14. The system as recited in claim 11, wherein interoperating components for implementing a transfer protocol providing message acknowledgment behavior and retention of a window of messages include:

message acknowledgment components, the message acknowledgement  
15 components configurably activatable and deactivatable to assist in implementing a selected transfer protocol, from among the set of extensible transfer protocols, the message acknowledgment components activated to provide message acknowledgement behavior for the transfer protocol; and

time window components, the time window components configurably  
20 activatable and deactivatable to assist in implementing a selected transfer protocol, from among the set of extensible transfer protocols, the time window components activated to provide retention of a window of messages for the transfer protocol.

15. In a distributed messaging system (100) including a message log (102) and a cursor (103), a method for configuring an end-to-end message assurance for  
25 message transfer from message providers (106) to message consumers (107), the method comprising:

an act of receiving runtime configuration data (109) indicating a combination of a capture assurance, a transfer assurance, and a delivery assurance that are to provide a desired end-to-end message assurance for message transfer in  
30 the distributed messaging system (100);

an act of formulating a message log component configuration (162) from the runtime configuration data (109), the message log component configuration (162) for activating and deactivating appropriate message log components (152) to compose the capture assurance and to compose a first portion of the transfer assurance to interoperate with a corresponding second portion of the transfer assurance provided by cursor components;

an act of formulating a cursor component configuration (163) from the runtime configuration data (109), the cursor component configuration (163) for activating and deactivating appropriate cursor components (153) to compose the delivery assurance and to compose the corresponding second portion of the transfer assurance to interoperate with the first portion of the transfer assurance provided by the message log components; and

an act of instructing the distributed messaging system (101) to compose the capture assurance, the transfer assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance for the distributed messaging system, including:

an act of sending the message log component configuration (162) to the message log (102) to instruct the message log (102) to compose the capture assurance and the first portion of transfer assurance; and

an act of sending the cursor component configuration (163) to the cursor (103) to instruct the cursor (103) to compose the delivery assurance and the second portion of the transfer assurance.

16. The method as recited in claim 15, wherein the act of receiving runtime configuration data indicating a combination of a capture assurance, a transfer assurance, and a delivery assurance comprise receiving runtime configuration data indicating a combination of a capture assurance, a transfer assurance, and a delivery assurance that are to provide durable exactly-once message delivery.

17. The method as recited in claim 15, wherein an act of instructing the distributed messaging system to compose the capture assurance, the transfer assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance for the distributed messaging

system comprises an act of instructing the distributed messaging system to compose the capture assurance, the transfer assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance for a message queuing system.

5 18. The method as recited in claim 15, wherein an act of instructing the distributed messaging system to compose the capture assurance, the transfer assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance for the distributed messaging system comprises an act of instructing the distributed messaging system to  
10 compose the capture assurance, the transfer assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance for a publish/subscribe messaging system.

19. The method as recited in claim 15, wherein an act of instructing the distributed messaging system to compose the capture assurance, the transfer  
15 assurance, and the delivery assurance at the message log and cursor so as to implement the desired end-to-end message assurance for the distributed messaging system comprises an act of instructing the distributed messaging system to compose the capture assurance, the transfer assurance, and the delivery assurance at  
20 the message log and cursor so as to implement one of: best effort message delivery, at-most-once message delivery, at-least-once message delivery, and exactly-once message delivery and one of: durable message delivery and volatile message delivery.

20. The method as recited in claim 15, further comprising:  
an act of the message log activating and deactivating the message log  
25 components as instructed in the message log component configuration to compose the capture protocol and the first portion of the transfer protocol; and

an act of the cursor activating and deactivating the cursor components as instructed in the cursor component configuration to compose the delivery protocol and the second portion of the transfer protocol.

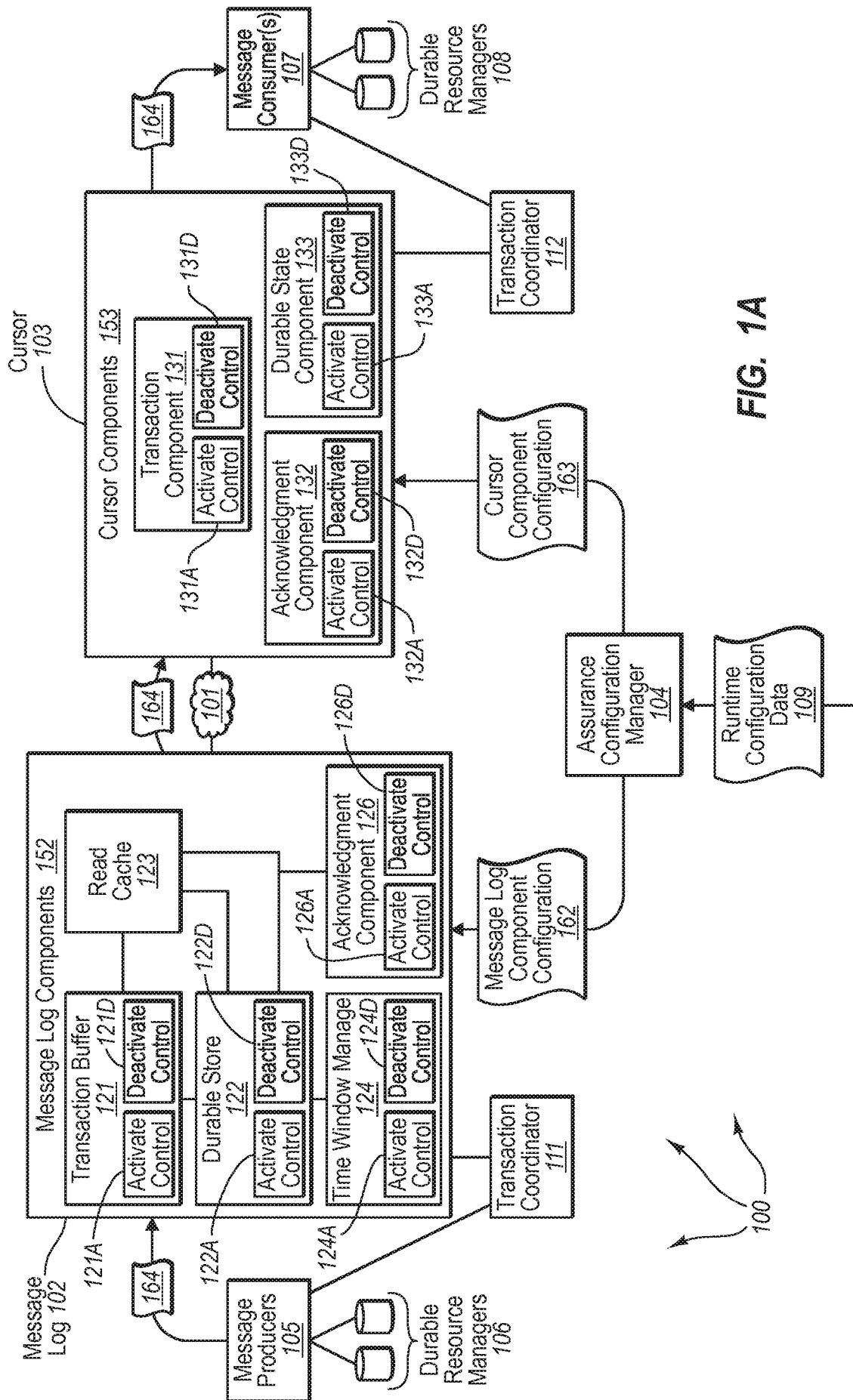
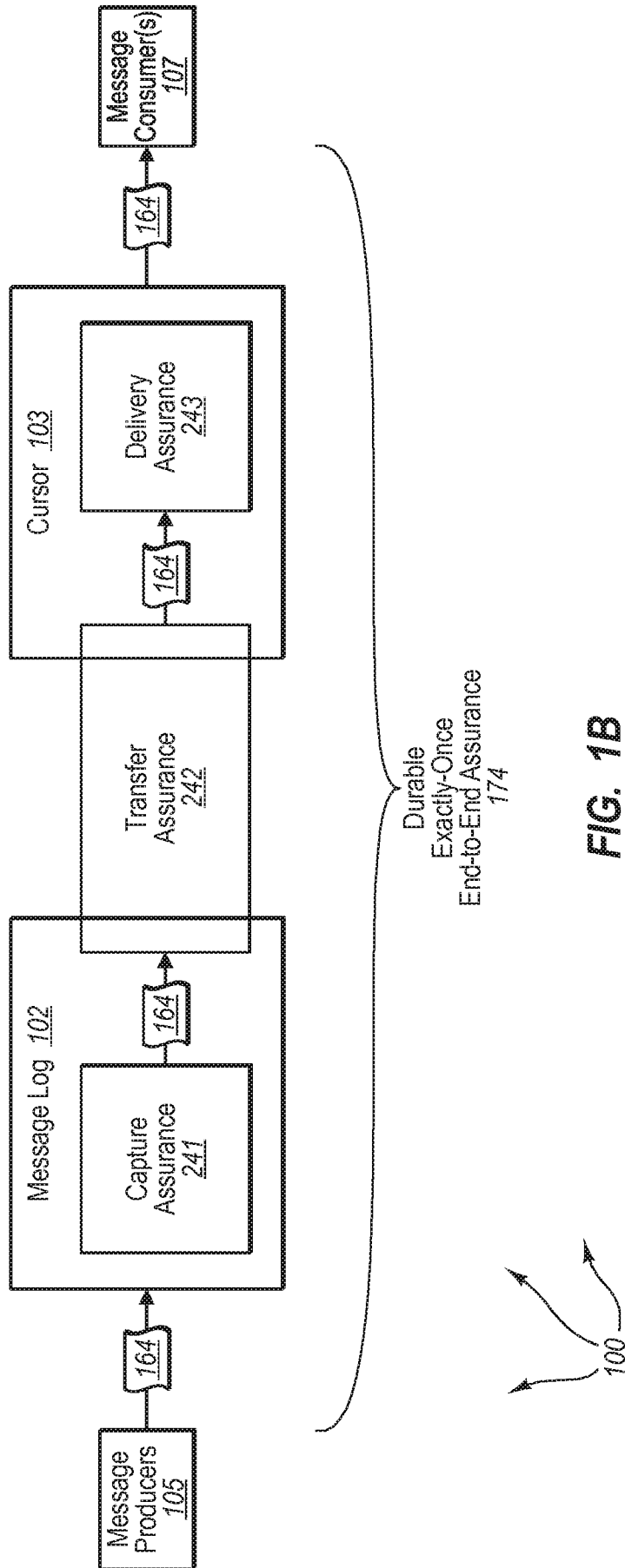


FIG. 1A



**FIG. 1B**



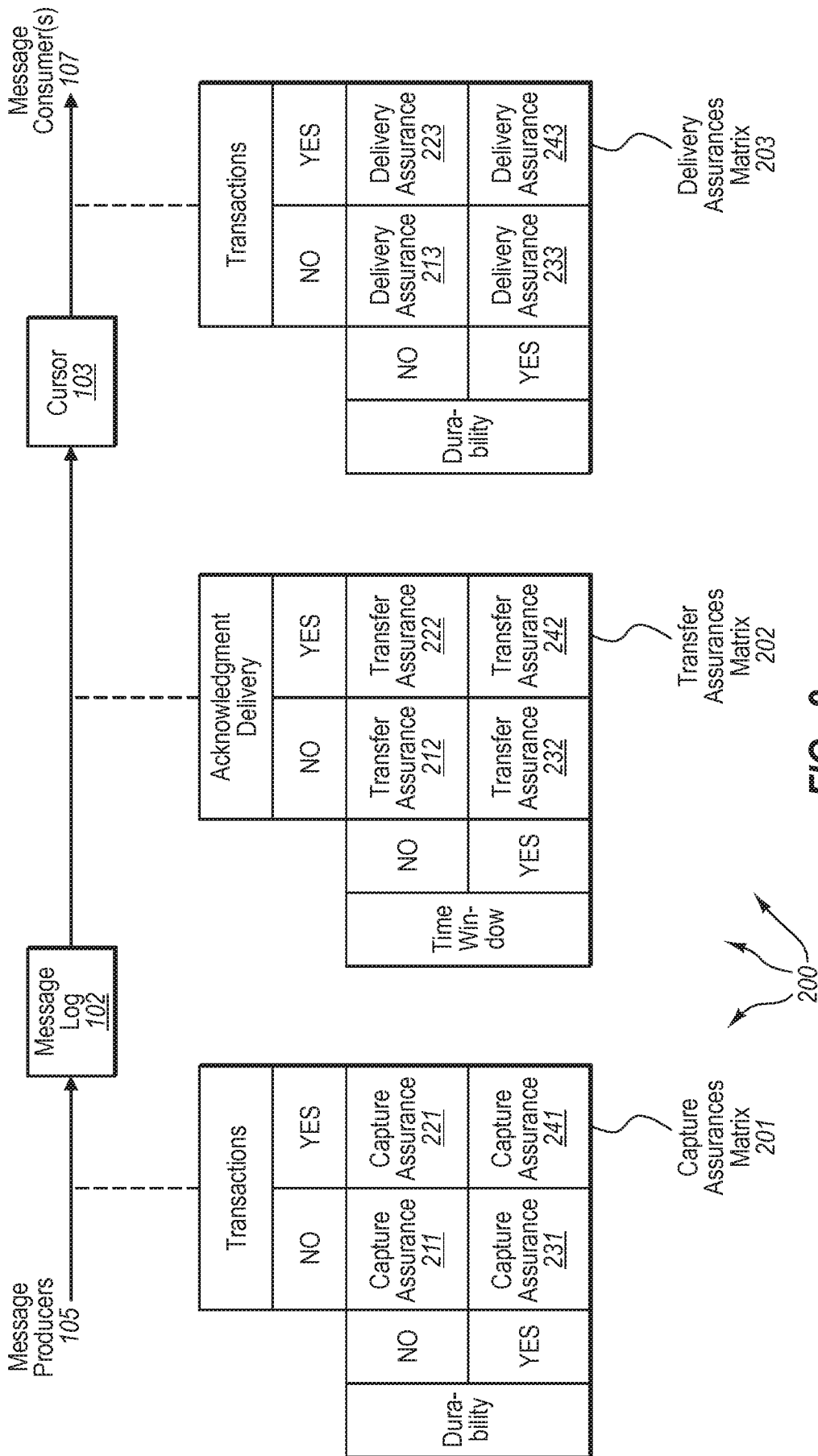


FIG. 2

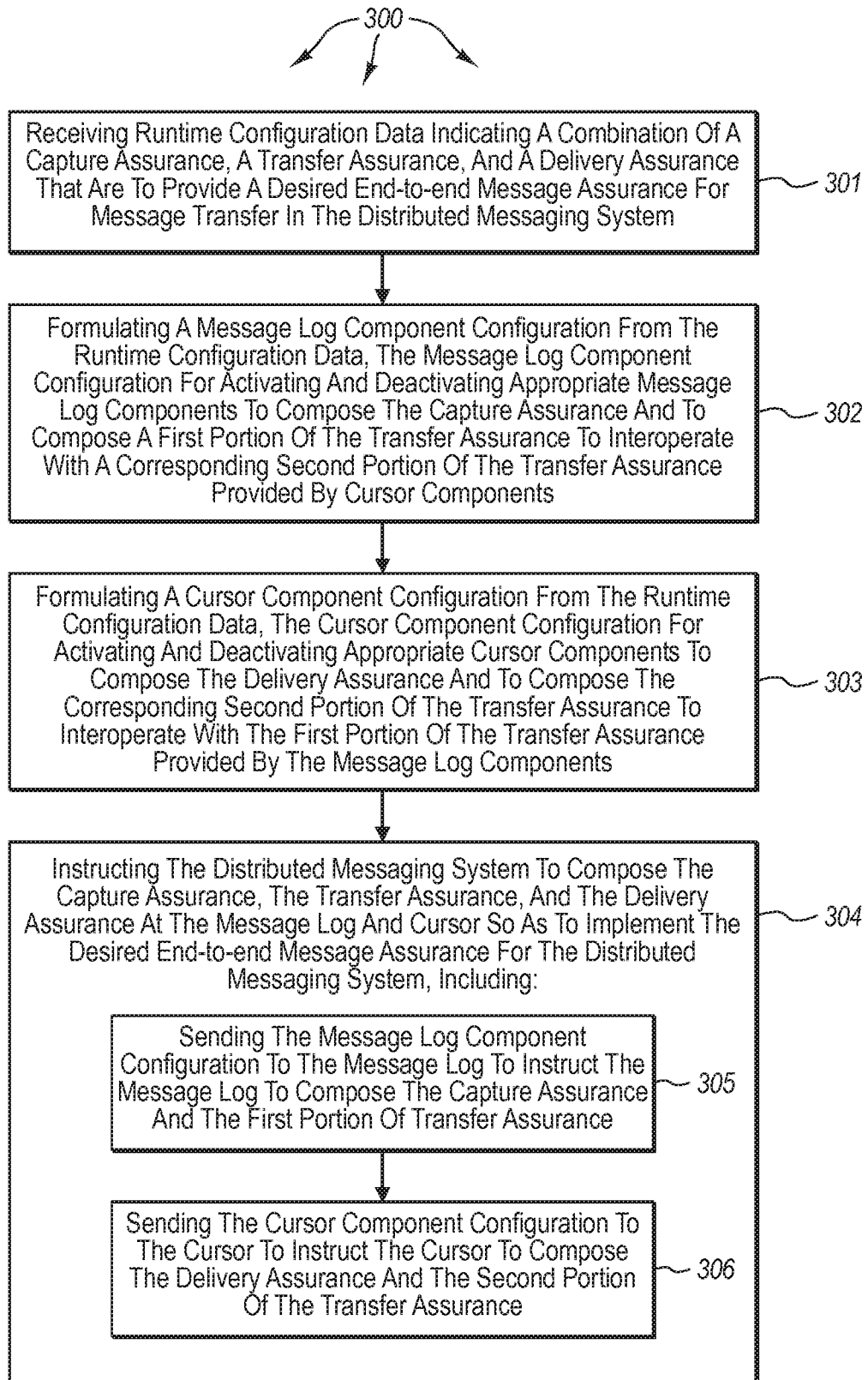


FIG. 3

**A. CLASSIFICATION OF SUBJECT MATTER****G06Q 50/00(2006.01)i**

According to International Patent Classification (IPC) or to both national classification and IPC

**B. FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

IPC8 G06F19/00, G06F17/00, G06Q10/00-99/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Korean Utility models and applications for Utility models since 1975

Japanese Utility models and applications for Utility models since 1975

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

PAJ, FPD, USPAT, eKIPASS "Keyword: distributed, message, assurance, configure, log, cursor, capture"

**C. DOCUMENTS CONSIDERED TO BE RELEVANT**

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 7,162,512 B1 ( NETA AMIT et al. ) 09 January 2007 See the abstract; claim 1; figures 2-3.	1-20
A	US 7,050,432 B1 ( GURUDUTH SOMASEKHARA BANAVAR et al. ) 23 May 2006 See the abstract; claim 1; figure 5.	1-20
A	US 2004-0205781 A1 ( RICHARD D. HILL et al. ) 14 October 2004 See the abstract; claim 1; figure 2.	1-20
A	US 2004-0111652 A1 ( SHAHID SHOAIB et al. ) 10 June 2004 See the abstract; claim 1; figure 1.	1-20

 Further documents are listed in the continuation of Box C. See patent family 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 application or patent 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 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

"&amp;" document member of the same patent family

Date of the actual completion of the international search

28 APRIL 2009 (28.04.2009)

Date of mailing of the international search report

**30 APRIL 2009 (30.04.2009)**

Name and mailing address of the ISA/KR

Korean Intellectual Property Office  
Government Complex-Daejeon, 139 Seonsa-ro, Seo-  
gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Authorized officer

WEE Jae Woo

Telephone No. 82-42-481-8540



**INTERNATIONAL SEARCH REPORT**

Information on patent family members

International application No.

**PCT/US2008/083654**

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 7,162,512 B1	09.01.2007	WO 0165785 A1	07.09.2001
US 7,050,432 B1	23.05.2006	None	
US 2004-205781 A1	14.10.2004	AU 2004-200732 A1 BR 0400775 A CA 2460274 A1 CN 1534923 A EP 1463249 A2 EP 1463249 A3 IL 160572 D0 JP 2004-295895 A KR 10-2004-0086583 A PA 04002731 A NO 20041261 A NZ 531382 A PL 366440 A1 RU 2004109131 A SG 116532 A1 ZA 200401582 A	14.10.2004 26.10.2004 27.09.2004 06.10.2004 29.09.2004 07.06.2006 25.07.2004 21.10.2004 11.10.2004 17.06.2005 28.09.2004 28.10.2005 04.10.2004 10.09.2005 28.11.2005 31.08.2004
US 2004-0111652 A1	10.06.2004	JP 2004-192642 A US 7152180 B2	08.07.2004 19.12.2006