

Method and System for Adaptively Managing a Request for Insurance

5 FIELD OF INVENTION

[1] The present disclosure relates broadly, but not exclusively, to methods and systems for adaptively managing a request for insurance.

10 BACKGROUND

[2] Purchasing insurance is a common way of protecting life's uncertainties using an amount of money that the user pays to an insurance company typically in a periodic manner. In the digital world where everything is online, the problem of underwriting and issuing policy has also become digital. A user can now purchase insurance via online platforms with greater ease and accessibility, where the eligibility to buy a policy is evaluated automatically and upon payment (also performed automatically via digital payment mechanisms), the policy is issued to the user.

[3] However, technology comes with its own challenges. For example, system failures can prevent users from being able to apply for policies online. This can be problematic where there is a need for insurance to be obtained on an expedited basis, for example for an on-demand service such as ride-hailing, or food or package delivery.

[4] Common underwriting solutions for selling insurance digitally include, for example, a single underwriting system which sells one particular type of insurance product based on business contractual agreement, and an insurance arrangement in which multiple insurers have quota shares in generated revenue from sales of insurance policies. However, both of these arrangements are heavily dependent on the service providers being technically available and stable.

30

[5] A need therefore exists to provide methods and systems that seek to overcome or at least minimize the above mentioned challenges.

SUMMARY

[7] According to a first aspect of the present disclosure, there is provided a method for adaptively managing a request for insurance, comprising: sending, by a switching server, the request to a first one of a plurality of service provider servers; determining, by the switching server, whether a response is received from the first one of the plurality of service provider servers, and sending, by the switching server, the request to a second one of the plurality of service provider servers based on the determination of whether a response is received from the first one of the plurality of service provider servers.

[8] According to a second aspect of the present disclosure, there is provided a system for adaptively managing a request for insurance, comprising: at least one processor; and at least one memory including computer program code; the at least one memory and the computer program code configured to, with the at least one processor, cause the system at least to: send the request to a first one of a plurality of service provider servers when the request for insurance is determined to be eligible; determine whether a response is received from the first one of the plurality of service provider servers, and send the request to a second one of a plurality of service provider servers based on the determination of whether a response is received from the first one of the plurality of service provider servers.

BRIEF DESCRIPTION OF THE DRAWINGS

[10] Embodiments and implementations are provided by way of example only, and will be better understood and readily apparent to one of ordinary skill in the art from the following written description, read in conjunction with the drawings, in which:

[11] Fig. 1 illustrates a system for adaptively managing a request for insurance according to various embodiments of the present disclosure.

[12] Fig. 2 is a schematic diagram of how an insurance product may be provided by more than one service provider, according to various embodiments of the present disclosure.

[13] Fig. 3 is an overview of a process for adaptively managing a request for insurance, according to various embodiments.

[14] Fig. 4 is an overview of a process when a service provider is not available, according to various embodiments.

[15] Fig. 5 is an overview of a process for recovering a service provider that was not available, according to various embodiments.

[16] Fig. 6 illustrates an example flow diagram for adaptively managing a request for insurance according to various embodiments.

[17] Fig. 7 illustrates an example flow diagram of how a service provider may be reinstated after a response is not received according to various embodiments.

[18] Figs. 8A and 8B form a schematic block diagram of a general purpose computer system upon which the transaction processing server of Fig. 1 can be practiced.

[19] Fig. 8C is a schematic block diagram of a general purpose computer system upon which the switching server of Fig. 1 can be practiced.

[20] Fig. 8D is a schematic block diagram of a general purpose computer system upon which a combined transaction processing server and switching server of Fig. 1 can be practiced.

5 **[21]** Fig. 9 shows an example of a computing device to realize the transaction processing server shown in Fig. 1.

[22] Fig. 10 shows an example of a computing device to realize the switching server shown in Fig. 1.

10

[23] Fig. 11 shows an example of a computing device to realize a combined transaction processing and switching server shown in Fig. 1.

[24] Skilled artisans will appreciate that elements in the figures are illustrated for simplicity and clarity and have not necessarily been depicted to scale. For example, the dimensions of some of the elements in the illustrations, block diagrams or flowcharts may be exaggerated in respect to other elements to help to improve understanding of the present embodiments.

20 DETAILED DESCRIPTION

[25] Embodiments will be described, by way of example only, with reference to the drawings. Like reference numerals and characters in the drawings refer to like elements or equivalents.

25 **[26]** Some portions of the description which follows are explicitly or implicitly presented in terms of algorithms and functional or symbolic representations of operations on data within a computer memory. These algorithmic descriptions and functional or symbolic representations are the means used by those skilled in the data processing arts to convey most effectively the substance of their work to others skilled in the art. An algorithm is
30 here, and generally, conceived to be a self-consistent sequence of steps leading to a desired result. The steps are those requiring physical manipulations of physical quantities, such as electrical, magnetic or optical signals capable of being stored, transferred, combined, compared, and otherwise manipulated.

[27] Unless specifically stated otherwise, and as apparent from the following, it will be appreciated that throughout the present specification, discussions utilizing terms such as “monitoring”, “utilizing”, “retrieving”, “providing”, “generating”, “quantifying”, “calculating”, “outputting”, “optimising”, “rebuilding”, “storing”, “mapping”, “checking”, “identifying”, “collecting”, “searching”, “conducting”, “cross-checking”, “aggregating”, “determining”, “regenerating”, “updating”, “comparing”, “adjusting”, “compiling”, “performing”, “obtaining”, “predicting” or the like, refer to the action and processes of a computer system, or similar electronic device, that manipulates and transforms data represented as physical quantities within the computer system into other data similarly represented as physical quantities within the computer system or other information storage, transmission or display devices.

[28] In addition, the present specification also implicitly discloses a computer program, in that it would be apparent to the person skilled in the art that the individual steps of the method described herein may be put into effect by computer code. The computer program is not intended to be limited to any particular programming language and implementation thereof. It will be appreciated that a variety of programming languages and coding thereof may be used to implement the teachings of the disclosure contained herein. Moreover, the computer program is not intended to be limited to any particular control flow. There are many other variants of the computer program, which can use different control flows without departing from the scope of the specification.

[29] Furthermore, one or more of the steps of the computer program may be performed in parallel rather than sequentially. Such a computer program may be stored on any computer readable medium. The computer readable medium may include storage devices such as magnetic or optical disks, memory chips, or other storage devices suitable for interfacing with a computer. The computer readable medium may also include a hard-wired medium such as exemplified in the Internet system, or wireless medium such as exemplified in the GSM mobile telephone system. The computer program when loaded and executed on such a computer effectively results in an apparatus that implements the steps of the preferred method.

[30] An insurance may be referred to as an arrangement by which a person or an entity such as company or state undertakes to provide a guarantee of compensation for specified loss, damage, illness, or death in return for payment of a specified premium. An insurance plan or insurance product is accordingly referred to as a plan or product that provides such

an arrangement. Insurance may be purchased to protect against events such as accidents, illnesses, death, loss of luggage, damage to property, house fires and other similar events. An insurance product may be provided by a service provider such as an insurer, underwriter, or other similar entity for providing an insurance product. Nowadays,
 5 it is possible to purchase insurance via, for example, a booking application that may provide services across different verticals such as travel, food, mart, insurance and other similar verticals. In some cases, booking applications can provide different insurance products from different service providers, and may be known as insurance aggregators.

10 **[31]** A user may be any suitable type of entity, which may include a person, a consumer looking to purchase a good or service via a transaction processing server, a seller looking to sell a good or service via the transaction processing server, a motorcycle driver or pillion rider in a case of the user looking to book or provide a motorcycle ride via the transaction processing server, a car driver or passenger in a case of the user looking to book or
 15 provide a car ride via the transaction processing server, and other similar entity. In the present disclosure, the user is also able to purchase an insurance product via the transaction processing server. For example, a user may be a regular customer of the concerned booking application who also decides to purchase an insurance product using the booking application via the transaction processing server. The user may also be a
 20 seller or driver providing a good or service via the booking application and who decides to purchase an insurance product using the booking application via the transaction processing server. A user who is registered to the transaction processing server or switching server will be called a registered user. A user who is not registered to the transaction processing server or switching server will be called a non-registered user.
 25 Depending on the booking application, an unregistered user may or may not be able to enjoy certain services provided by the booking application, or may or may not be able to be a deliverer or driver of the concerned booking application, or other similar disadvantages. The term user will be used to collectively refer to both registered and non-registered users. A user may at various places within this specification be referred to as a
 30 requestor (e.g. a person who requests for a good or service) or a provider (e.g. a person who provides the requested good or service to the requestor).

[32] A switching server may be a server that hosts software application programs for adaptively managing a request for insurance. The switching server may be implemented
 35 as shown in Figs. 10 or 11 for adaptively managing a request for insurance.

[33] The transaction processing server may be a server that hosts software application programs for processing payment transactions for, for example, purchasing of an insurance product by a user. The transaction processing server may communicate with one or more other servers (e.g., a switching server) concerning processing payment transactions for insurance products. The transaction processing server may communicate with a switching server to facilitate purchase of an insurance product. The transaction processing server may use a variety of different protocols and procedures in order to process the payment.

[34] Transactions that may be performed via a transaction processing server further include product or service purchases, credit purchases, debit transactions, fund transfers, account withdrawals, etc. Transaction processing servers may be configured to process transactions via cash-substitutes, which may include payment cards, letters of credit, checks, payment accounts, etc.

[35] The transaction processing server is usually managed by a service provider that may be an entity (e.g. a company or organization) which operates to process user requests for insurance. The transaction processing server may include one or more computing devices that are used for processing user requests for insurance.

[36] A transaction account may be an account of a user who is registered at a transaction processing server. The user can be a customer, a ride-hailing provider (e.g., a driver), or any third parties such as a delivery service provider (e.g., a courier) who want to use the transaction processing server. In certain circumstances, the transaction account is not required to use the switching server. A transaction account includes details (e.g., name, address, vehicle, face image, etc.) of a user. The transaction processing server manages the transaction accounts of users and the interactions between users and other external servers.

[37] Fig. 1 illustrates a block diagram of an example system 100 for adaptively managing a request for insurance. Further, the system 100 enables a payment transaction for an insurance product.

[38] The example system 100 comprises a requestor device 102, a provider device 104, an acquirer server 106, a transaction processing server 108, an issuer server 110, a switching server 140 and service provider servers 150A to 150N.

5 **[39]** The requestor device 102 is in communication with a provider device 104 via a connection 112. The connection 112 may be an ad hoc connection (e.g., via NFC communication, Bluetooth, etc.) or over a network (e.g., the Internet). The requestor device 102 is also in communication with the switching server 140 via a connection 121 (e.g., the Internet). The requestor device 102 may also be connected to a cloud that
10 facilitates the system 100 for adaptively managing a request for insurance. For example, the requestor device 102 can send a signal or data to the cloud directly via an ad hoc connection (e.g., via NFC communication, Bluetooth, etc.) or over a network (e.g., the Internet).

15 **[40]** The provider device 104 is in communication with the requestor device 102 as described above, usually via the transaction processing server 108. The provider device 104 is, in turn, in communication with an acquirer server 106 via a connection 114. The provider device 104 is also in communication with the switching server 140 via a connection 123. The connections 114 and 123 may be network connections (e.g., the
20 Internet). The provider device 104 may also be connected to a cloud that facilitates the system 100 for adaptively managing a request for insurance. For example, the provider device 104 can send a signal or data to the cloud directly via an ad hoc connection (e.g., via NFC communication, Bluetooth, etc.) or over a network (e.g., the Internet).

25 **[41]** The acquirer server 106, in turn, is in communication with the transaction processing server 108 via a connection 116. The transaction processing server 108, in turn, is in communication with an issuer server 110 via a connection 118. The connections 116 and 118 may each be a network connection (e.g., the Internet).

30 **[42]** The transaction processing server 108 is further in communication with the switching server 140 via a connection 120. The connection 120 may be over a network (e.g., a local area network, a wide area network, the Internet, etc.). In one arrangement, the transaction processing server 108 and the switching server 140 are combined and the connection 120 may be an interconnected bus.

[43] The switching server 140, in turn, is in communication with the service provider servers 150A to 150N via respective connections 122A to 122N. The connections 122A to 122N may each be network connections (e.g., the Internet). The switching server 140 may also be connected to a cloud that facilitates the system 100 for adaptively managing a request for insurance. For example, the switching server 140 can send a signal or data to the cloud directly via an ad hoc connection (e.g., via NFC communication, Bluetooth, etc.) or over a network (e.g., the Internet).

[44] The service provider servers 150A to 150N are collectively referred to herein as service provider servers 150. In an example, the service provider server 150 may be one managed by a service provider such as an insurer, underwriter, or other similar entity for providing an insurance product and the switching server 140 is a server that adaptively manages user requests for insurance and decides which of the service provider servers 150 to forward a user request for insurance.

[45] In the illustrative embodiment, each of the devices 102, 104, and the servers 106, 108, 110, 140, and 150 provides an interface to enable communication with other connected devices 102, 104, 142 and/or servers 106, 108, 110, 140, and 150. Such communication is facilitated by an application programming interface ("API"). Such APIs may be part of a user interface that may include graphical user interfaces (GUIs), Web-based interfaces, programmatic interfaces such as application programming interfaces (APIs) and/or sets of remote procedure calls (RPCs) corresponding to interface elements, messaging interfaces in which the interface elements correspond to messages of a communication protocol, and/or suitable combinations thereof.

[46] Use of the term 'server' herein can mean a single computing device or a plurality of interconnected computing devices which operate together to perform a particular function. That is, the server may be contained within a single hardware unit or be distributed among several or many different hardware units.

[47] The switching server 140 is associated with an entity (e.g. a company or organization or moderator of the service). In one arrangement, the switching server 140 is owned and operated by the entity operating the transaction processing server 108. In such an arrangement, the switching server 140 may be implemented as a part (e.g., a

computer program module, a computing device, etc.) of the transaction processing server 108.

[48] The transaction processing server 108 may also be configured to manage the registration of users. A registered user has a transaction account (see the discussion above) which includes details of the user. The registration step is called on-boarding. A user may use either the requestor device 102 or the provider device 104 to perform on-boarding to the transaction processing server 108.

[49] It may not be necessary to have a transaction account at the transaction processing server 108 to access the functionalities of the transaction processing server 108. However, depending on the booking application, an unregistered user may have access to only a subset of the functionality available to registered users. For example, an unregistered user may or may not be able to enjoy certain services provided by the booking application, or may or may not be able to be a deliverer or driver of the concerned booking application.

[50] The on-boarding process for a user is typically performed by the user through one of the requestor device 102 or the provider device 104. In one arrangement, the user downloads an app (which includes, or otherwise provides access to, the API to interact with the transaction processing server 108) to the requestor device 102 or the provider device 104. In another arrangement, the user accesses a website (which includes, or otherwise provides access to, the API to interact with the transaction processing server 108) on the requestor device 102 or the provider device 104. The user is then able to interact with the switching server 140. The user may be a requestor or a provider associated with the requestor device 102 or the provider device 104, respectively.

[51] Details of the registration may include, for example, name of the user, address of the user, emergency contact, blood type or other healthcare information, next-of-kin contact, permissions to retrieve data and information from the requestor device 102 and/or the provider device 104 for insurance purchase or underwriting purposes, such as permission to use a camera of the requestor device 102 and/or the provider device 104 to take a picture of the requestor's or provider's face, wherein the picture may be used for insurance purchase purposes. Alternatively, another mobile device may be selected instead of the requestor device 102 and/or the provider device 104 for retrieving the

picture. Once on-boarded, the user would have a transaction account that stores all the details.

[52] The requestor device 102 is associated with a customer (or requestor) who is a party to a request (e.g. a ride-hailing request or a delivery request) that occurs between the requestor device 102 and the provider device 104. The requestor device 102 may be a computing device such as a desktop computer, an interactive voice response (IVR) system, a smartphone, a laptop computer, a personal digital assistant computer (PDA), a mobile computer, a tablet computer, and the like.

[53] The requestor device 102 includes transaction credentials (e.g., a payment account) of a requestor to enable the requestor device 102 to be a party to a payment transaction, for example for purchasing an insurance product. If the requestor has a transaction account, the transaction account may also be included (i.e., stored) in the requestor device 102. For example, a mobile device (which is a requestor device 102) may have the transaction account of the requestor stored in the mobile device.

[54] In one example arrangement, the requestor device 102 is a computing device in the form of a watch or similar wearable and is fitted with a wireless communications interface (e.g., a NFC interface). The requestor device 102 can then electronically communicate with the switching server 140 and the transaction processing server 108 regarding a user request for insurance, or electronically communicate with the provider device 104 for a transaction request for a good or service. The requestor uses the watch or similar wearable to make a user request or a transaction request by pressing a button on the watch or wearable.

[55] The provider device 104 is associated with a provider who is also a party to the transaction request that occurs between the requestor device 102 and the provider device 104. The provider device 104 can also be party to a payment transaction for purchasing an insurance product. The provider device 104 includes transaction credentials (e.g., a payment account) of a provider to enable the provider device 104 to be a party to a payment transaction, for example for purchasing an insurance product. If the provider has a transaction account, the transaction account may also be included (i.e., stored) in the provider device 104. For example, a mobile device (which is a provider device 104) may have the transaction account of the provider stored in the mobile device.

[56] The provider device 104 may be a computing device such as a desktop computer, an interactive voice response (IVR) system, a smartphone, a laptop computer, a personal digital assistant computer (PDA), a mobile computer, a tablet computer, and the like. The provider device 104 can then electronically communicate with the switching server 140 and the transaction processing server 108 regarding a user request for insurance, or electronically communicate with the requestor device 102 for a transaction request for a good or service.

[57] Hereinafter, the term “provider” refers to any third party associated with providing a good or service for purchase, or a travel or ride or delivery service via the provider device 104. Therefore, the transaction account of a provider refers to both the transaction account of a provider and the transaction account of a third party (e.g., a travel co-ordinator or merchant) associated with the provider.

[58] If the provider has a transaction account, the transaction account may also be included (i.e., stored) in the provider device 104. For example, a mobile device (which is a provider device 104) may have the transaction account of the provider stored in the mobile device.

[59] In one example arrangement, the provider device 104 is a computing device in a watch or similar wearable and is fitted with a wireless communications interface (e.g., a NFC interface). The provider device 104 can then electronically communicate with the switching server 140 and the transaction processing server 108 regarding a user request for insurance, or electronically communicate with the requestor to make a transaction request by pressing a button on the watch or wearable.

[60] The acquirer server 106 is associated with an acquirer who may be an entity (e.g. a company or organization) which issues (e.g. establishes, manages, administers) a payment account (e.g. a financial bank account) of a provider or a service provider. An acquirer may be a bank or other financial institution. As discussed above, the acquirer server 106 may include one or more computing devices that are used to establish communication with another server (e.g., the transaction processing server 108) by exchanging messages with and/or passing information to the other server. The acquirer

server 106 forwards the payment transaction relating to a user request for insurance or relating to a transaction request to the transaction processing server 108.

[61] The transaction processing server 108 is configured to carry out processes relating to a transaction account by, for example, forwarding data and information associated with the transaction to the other servers in the system 100 such as the switching server 140. In an example, the transaction processing server 108 may provide confirmation of payment for a user request for insurance to the switching server 140.

[62] The issuer server 110 is associated with an issuer and may include one or more computing devices that are used to perform a payment transaction. The issuer may be an entity (e.g. a company or organization) which issues (e.g. establishes, manages, administers) a transaction credential or a payment account (e.g. a financial bank account) associated with the owner of the requestor device 102. As discussed above, the issuer server 110 may include one or more computing devices that are used to establish communication with another server (e.g., the transaction processing server 108) by exchanging messages with and/or passing information to the other server.

[63] The service provider server 150 is a server associated with an entity (e.g. a company or organization) which manages (e.g. establishes, administers) and provides insurance products for sale to a user. In one arrangement, the entity is a service provider such as an insurer or underwriter that sells insurance products.

[64] Switching between service providers, or between servers thereof where a service provider maintains multiple servers (e.g. service provider servers 150), increases the availability and resilience of systems providing insurance products. It remains a challenge to determine how the request distribution should be handled. To address this, in some embodiments, a weighted distribution of traffic and falling back to available servers in case of failures can be used. Possible benefits include the ability to provide customers with the best prices with support for discounts, to provide a better user experience to customers by increasing availability of service providers to process user requests for insurance, and to abstract customers from what is happening behind the scenes during an insurance purchase. For insurance aggregators, it can also help in better resilience of the system.

[65] To increase the availability, resilience, and robustness of insurance aggregators, a strategy to merge service providers for similar insurance products and coverage types along with weighted traffic distribution and fallback system is proposed in the present disclosure. All the three steps mentioned above can be implemented to automatically
 5 mitigate the failures of downstream service of the system.

[66] For example, insurance of a same type from different service providers or underwriters will be merged into one insurance product, from the perspective of possible buyers of insurance, while configuring the insurance products into the system. During
 10 processing of a user request for insurance, the system will determine which service provider the request should be routed to, for example based on the current state of service provider servers 150. Failures from service providers (e.g. a service provider server 150 failing to send a response to the switching server 140 to indicate that a user request for insurance is successfully received) will be taken into consideration for dynamically
 15 controlling the traffic distribution percentage.

[67] Fig. 2 is a schematic diagram 200 of how an insurance product may be provided by more than one service provider, according to various embodiments of the present disclosure. The process begins during configuration of insurance products in the system.
 20 Each insurance product which a user can purchase may represent an insurance plan from each of a plurality of different service providers. For example, a service provider 202 may offer an insurance plan 204, while another service provider 206 may offer an insurance plan 208. The insurance plans 204 and 208 may be merged and presented in a booking application as a single insurance product for purchase. The merging of insurance plans
 25 across different service providers may be based on factors such as coverage type, price, and other similar factors, and insurance plans that are merged into a single insurance product are similar to one another. Each plan 204 and 208 has an associated cost (premium) which the user must pay during purchase. In this case, the price for the insurance product can be the same as the price of each of plan 204 and 208 (e.g. SGD
 30 2).

[68] To merge the plans 204 and 208 into an insurance product, a unique id can be assigned to each plan 204 and 208. Each plan 204 and 208 may also have an identifier which can be used to recognize the same plan from different underwriters. For example,
 35 the insurance product that is created based on the merger of plans 204 and 208 may be

titled as “Plan A” and may be associated with an identifier such as “djdfh1233”. Table 210 depicts an example of how insurance product Plan A may be registered in a data store. For example, plans 204 and 208 can be input as two different rows in table 210, along with their corresponding description details in the middle column, and an indication of associated service providers 202 and 206 in the right column. The table 210 may be used to show metadata (for example such as a name or title of the insurance product, details of insurance coverage provided by the insurance product such as what events are covered and the maximum cover for each event, price, etc) for each insurance product on a front end such as a user interface for purchasing the insurance product. The metadata to be shown for each insurance product can be from any of the associated service providers (e.g. in the case of insurance product Plan A, it would be from associated service providers 202 and 204) which can be configuration based.

[69] Fig. 3 is an illustration 300 of a process for adaptively managing a request for insurance, according to various embodiments. A user 302 may decide on an insurance product to purchase from a booking application and send a request for insurance to the switching server 140 to purchase the insurance product. The request may be sent via a user device 304, which may be the requestor device 102 or the provider device 104. Based on the request, underwriting checks may be performed to check if the user is eligible for the insurance product. For example, an underwriting process may consider information relating to the user 302 such as age, gender, occupation, medical record, financial history, amount of credit stored with the booking application, and other similar information. Once the underwriting checks are cleared and the user is found to be eligible, the switching server 140 sends the request to a first one of a plurality of service providers (the choice of service provider typically being determined according to selection logic as detailed below), and determines whether a response is received from the first one of the plurality of service providers. The determination of whether a response is received may be in accordance with a predetermined threshold, the threshold being one that is determined based on a service level agreement (SLA). For example, the threshold may be a maximum number of times that a response is not received from the first one of the plurality of service providers, a number of times that a response is not received within a predetermined period of time from the first one of the plurality of service providers, or other similar type of threshold that is configured based on the SLA between the concerned first one of the plurality of service providers and the entity that owns and/or operates the booking application.

[70] In this example, the concerned insurance product may be a merger of three insurance plans from service providers A, B and C respectively which run service provider servers 306, 308 and 310 respectively. The switching server 140 may determine a plurality of service provider servers (e.g. the plurality of service provider servers 306, 308 and 310 run by service providers A, B and C respectively) for providing the insurance plan indicated in the request for insurance. The plurality of service provider servers may be determined based on data identifying the insurance product, for example data in the form of table 210 which indicates the associated service providers that can provide the insurance plans that are merged into the concerned insurance product. The switching server 140 may also be configured to determine an identifier associated with the insurance product based on the data, wherein determining the plurality of service provider servers is based on the identifier. For example, the request may indicate a title, name, ID number, or other similar type of identifier associated with the concerned insurance product, and the switching server may reference the data based on the identifier (e.g. referencing data wherein the identifier indicated in the left column of table 210 is the same as the identifier indicated in the request).

[71] As there are a plurality of service providers A, B and C that can provide the concerned insurance product, the switching server 140 has to select one of these service providers for sending the request. In one implementation, the switching server 140 may determine the first one of the plurality of service provider servers based on a weight assigned to each of the plurality of service provider servers. The assigned weight for each of the plurality of service provider servers designates a priority to each service provider server for selection by the switching server 140. The priority may be based on factors such as service level agreements (SLAs) and/or contractual agreements between the entity that owns and/or operates the concerned booking application and the respective service providers, current availability of each service provider server for handling requests for insurance, and other similar factors.

[72] In an implementation, a weighted distribution of requests wherein the weights can initially be manually configured in the system. For example, there may be 10 requests for an insurance product and 3 service provider servers (e.g. service provider servers 306, 308 and 310 run by service providers A, B and C respectively) that can provide the concerned insurance product. The weights can be configured such that service provider A serves 4 out of the 10 requests, service provider B serves 1 out of the 10 requests, and

service provider C serves 5 out of the 10 requests respectively. The configuration can be based on factors such as service level agreements (SLAs) and/or contractual agreements between the entity that owns and/or operates the concerned booking application and the respective service providers, current availability of each service provider server for handling requests for insurance, and other similar factors. The operation of this weighted distribution system for 10 requests and 3 service provider servers may be represented in the following equation:

$$\text{Max} = \text{Ratio(A)} + \text{Ratio(B)} + \text{Ratio(C)}$$

$$\text{e.g. } 10 = 4 + 1 + 5$$

[73] Based on an implementation of the above weighted distribution system, when a request for insurance is received, a random number may be generated between [1, 10]. The generated number corresponds to one of the 3 service providers A, B and C which would serve the request for insurance. For example, a generated random number between [1, 4] would mean that the request is to be served by service provider A, and the switching server 140 sends the request to the service provider server corresponding to service provider A (e.g. service provider server 306). If the generated random number is 5, service provider B will handle the request, and the switching server 140 sends the request to the service provider server corresponding to service provider B (e.g. service provider server 308). If the generated random number is between [6,10], service provider C will handle the request, and the switching server 140 sends the request to the service provider server corresponding to service provider C (e.g. service provider server 310). Probability of each number generated from the random number generator becomes equal when implemented on a larger scale e.g. for a larger number of requests. It will be appreciated that other implementations for distributing requests for insurance to an appropriate service provider are also possible. For example, round-robin allocation to servers may be performed up until the point at which respective servers have received their full allocation for a given period. In the example above, a first round of allocations may occur, such that each of servers A, B, and C receive one request. In the next round, since B has already received its single request, only A and C receive requests, and so on.

[74] Fig. 4 is an illustration 400 of a switching process that is carried out when a service provider server is not available, according to various embodiments. In this case, the service provider server 306 may be unavailable to receive and serve any request for

insurance that may be sent by the switching server 140, and accordingly is unable to provide any response to the switching server 140 to indicate that a request for insurance is received. Assuming that the switching server 140 has sent a request for insurance to service provider server 306, the switching server 140 then determines whether a response is received from the service provider server 306 in accordance with a predetermined threshold, the threshold being one that is determined based on a service level agreement. For example, the threshold may be a maximum number of times that a response is not received from the concerned service provider server 306, a number of times that a response is not received within a predetermined period of time from the concerned service provider 306, or other similar type of threshold that is configured based on the SLA between the concerned service provider A that owns and/or operates service provider server 306 and the entity that owns and/or operates the booking application.

[75] The switching server 140 may determine whether to send the request to a second one of the plurality of service provider servers based on the determination of whether a response is received from the first one of the plurality of service provider servers. Since no response has been received from the service provider server 306 to indicate that the request for insurance is duly received, the switching server 140 may send the request to either service provider server 308 or service provider server 310 instead. Further, the weights among the plurality of service provider servers may also be rebalanced or reassigned without the first one of the plurality of service provider servers (e.g. without service provider 306) if it is determined that the response is not received. The determination may be based on the threshold, for example the reassignment of weights may be done after the number of times that no response was received from service provider server 306 has exceeded a threshold number. Thereafter, no requests will be sent to service provider server 306 and instead be distributed between service provider server 308 and service provider server 310 based on the reassigned weights. The rebalanced or reassigned weights may be represented in the following equation:

$$\text{Max} = \text{Ratio(B)} + \text{Ratio(C)}$$

[76] In the present case, the value for 'Max' may be 6 since weights associated with service provider server 306 are to be omitted. When a further request for insurance is received, the switching server 140 may determine whether service provider server 308 or 310 should receive the request based on the reassigned weights, and then send the

request based on the determination. This advantageously ensures that even in the event of a disruption of service of any service provider server, there will still be an available service provider to handle a request for insurance.

[77] Fig. 5 is an illustration 500 of a process for recovering a service provider server that was not available, according to various embodiments. To bring back service provider server 306 to serve further requests for insurance, it is possible to work along the lines of a circuit breaker pattern. For example, a duration may be kept from a last time that a response was not received from the service provider server 306 (this can be as per defined SLA between the concerned service provider A that owns and/or operates service provider server 306 and the entity that owns and/or operates the booking application), after which a certain number of requests that are subsequently received may be sent to service provider server 306. Accordingly, the switching server 140 may send one or more other requests to the first one of the plurality of service provider servers (e.g. service provider server 306) at a predetermined period of time after the determination that the response (e.g. the last response wherein weights are reassigned and service provider server 306 is omitted from receiving further requests) is not received, the predetermined period of time being one that is determined based on the service level agreement. The switching server 140 may then determine whether a response is received for each of the one or more other requests from the service provider server 306 in accordance with the predetermined threshold. If, for example, a response is received for each of these one or more other requests indicating that these requests are successfully received by the service provider server 306, service provider server 306 may be reinstated to receive requests for insurance again. Further, weights among the plurality of service provider servers 306, 308 and 310 will be reassigned, for example back to the original weight distribution system based on $\text{Max} = \text{Ratio}(A) + \text{Ratio}(B) + \text{Ratio}(C)$. Advantageously, this enables the system to ensure that all service provider servers are regularly checked for availability to handle further requests for insurance, even for service provider servers that are presumed to be unavailable due to previous transmission failures.

[78] Fig. 6 illustrates an example flow diagram for adaptively managing a request for insurance according to various embodiments. In a step 602, a request for insurance is sent by a switching server to a first one of a plurality of service provider servers. The switching server may determine whether the request is eligible based on an underwriting process, wherein the request is sent to the first one of the plurality of service provider

servers when the request is determined to be eligible. In a step 604, it is determined by the switching server whether a response is received from the first one of the plurality of service provider servers. The determination may be in accordance with a predetermined threshold, the predetermined threshold being based on a service level agreement, the predetermined threshold being a maximum number of times that a response is not received, or a number of times that a response is not received within a predetermined period of time. It may be determined whether to send the request to a second one of the plurality of service provider servers based on the determination of whether a response is received from the first one of the plurality of service provider servers. The plurality of service provider servers may be determined for providing an insurance plan indicated in the request, the plurality of service provider servers being based on data identifying the insurance plan. An identifier associated with the insurance plan based on the data may be determined, wherein determining the plurality of service provider servers is based on the identifier. In a step 606, the request may be sent by the switching server to a second one of the plurality of service provider servers based on the determination of whether a response is received from the first one of the plurality of service provider servers.

[79] Fig. 7 illustrates an example flow diagram of how a service provider server may be reinstated after a response is not received according to various embodiments. In a step 702, the first one of the plurality of service provider servers is determined by the switching server based on a weight assigned to each of the plurality of service provider servers. Weights among the plurality of service provider servers may be reassigned without the first one of the plurality of service provider servers if it is determined that the response is not received. In a step 704, one or more other requests are sent to the first one of the plurality of service provider servers at a predetermined period of time after the determination that the response is not received, the predetermined period of time being one that is determined based on the service level agreement. In a step 706, it is determined whether a response is received for each of the one or more other requests from the first one of the plurality of service provider servers in accordance with the predetermined threshold, the predetermined threshold further being a maximum number of times that a response is received, or a number of times that a response is received within a predetermined period of time. In a step 708, weights among the plurality of service provider servers are reassigned with the first one of the plurality of service provider servers if it is determined that the response is received for each of the one or more other requests.

[80] Figs. 8A and 8B form a schematic block diagram of a general purpose computer system upon which the transaction processing server of Fig. 1 can be practiced.

[81] As seen in Fig. 8A, the computer system 1300 includes a computer module 1301.

5 An external Modulator-Demodulator (Modem) transceiver device 1316 may be used by the computer module 1301 for communicating to and from a communications network 1320 via a connection 1321. The communications network 1320 may be a wide-area network (WAN), such as the Internet, a cellular telecommunications network, or a private WAN. Where the connection 1321 is a telephone line, the modem 1316 may be a
10 traditional “dial-up” modem. Alternatively, where the connection 1321 is a high capacity (e.g., cable) connection, the modem 1316 may be a broadband modem. A wireless modem may also be used for wireless connection to the communications network 1320.

[82] The computer module 1301 typically includes at least one processor unit 1305,
15 and a memory unit 1306. For example, the memory unit 1306 may have semiconductor random access memory (RAM) and semiconductor read only memory (ROM). The computer module 1301 also includes an interface 1308 for the external modem 1316. In some implementations, the modem 1316 may be incorporated within the computer module 1301, for example within the interface 1308. The computer module 1301 also has
20 a local network interface 1311, which permits coupling of the computer system 1300 via a connection 1323 to a local-area communications network 1322, known as a Local Area Network (LAN). As illustrated in Fig. 8A, the local communications network 1322 may also couple to the wide network 1320 via a connection 1324, which would typically include a so-called “firewall” device or device of similar functionality. The local network
25 interface 1311 may comprise an Ethernet circuit card, a Bluetooth® wireless arrangement or an IEEE 802.11 wireless arrangement; however, numerous other types of interfaces may be practiced for the interface 1311.

[83] The I/O interfaces 1308 may afford either or both of serial and parallel connectivity,
30 the former typically being implemented according to the Universal Serial Bus (USB) standards and having corresponding USB connectors (not illustrated). Storage devices 1309 are provided and typically include a hard disk drive (HDD) 1310. Other storage devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may also be used. An optical disk drive 1312 is typically provided to act as a non-volatile
35 source of data. Portable memory devices, such optical disks, USB-RAM, portable,

external hard drives, and floppy disks, for example, may be used as appropriate sources of data to the system 1300.

[84] The components 1305 to 1312 of the computer module 1301 typically communicate via an interconnected bus 1304 and in a manner that results in a conventional mode of operation of the computer system 1300 known to those in the relevant art. For example, the processor 1305 is coupled to the system bus 1304 using a connection 1318. Likewise, the memory 1306 and optical disk drive 1312 are coupled to the system bus 1304 by connections 1319. Examples of computers on which the described arrangements can be practised include IBM-PC's and compatibles, Sun Sparcstations, Apple or like computer systems.

[85] The steps of the methods 600 and 700 in Figs. 6 and 7 facilitated by the transaction processing server 108 may be implemented using the computer system 1300. For example, the steps of the methods 600 and 700 may be implemented as one or more software application programs 1333 executable within the computer system 1300. In particular, the steps of the methods 600 and 700 as facilitated by the transaction processing server 108 are effected by instructions 1331 (see Fig. 6B) in the software 1333 that are carried out within the computer system 1300. The software instructions 1331 may be formed as one or more code modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part and the corresponding code modules facilitate the steps of the methods 600 and 700 and a second part and the corresponding code modules manage a user interface between the first part and the user.

[86] The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer system 1300 from the computer readable medium, and then executed by the computer system 1300. A computer readable medium having such software or computer program recorded on the computer readable medium is a computer program product. The use of the computer program product in the computer system 1300 preferably effects an advantageous apparatus for a transaction processing server 108.

[87] The software 1333 is typically stored in the HDD 1310 or the memory 1306. The software is loaded into the computer system 1300 from a computer readable medium, and

executed by the computer system 1300. Thus, for example, the software 1333 may be stored on an optically readable disk storage medium (e.g., CD-ROM) 1325 that is read by the optical disk drive 1312. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer
 5 program product in the computer system 1300 preferably effects an apparatus for a transaction processing server 108.

[88] In some instances, the application programs 1333 may be supplied to the user encoded on one or more CD-ROMs 1325 and read via the corresponding drive 1312, or
 10 alternatively may be read by the user from the networks 1320 or 1322. Still further, the software can also be loaded into the computer system 1300 from other computer readable media. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computer system 1300 for execution and/or processing. Examples of such storage media include floppy disks,
 15 magnetic tape, optical disk, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 1301. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data
 20 to the computer module 1301 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

[89] The second part of the application programs 1333 and the corresponding code modules mentioned above may be executed to implement one or more graphical user interfaces (GUIs) to be rendered or otherwise represented upon a display. Through manipulation of typically a keyboard and a mouse, a user of the computer system 1300 and the application may manipulate the interface in a functionally adaptable manner to
 25 provide controlling commands and/or input to the applications associated with the GUI(s). Other forms of functionally adaptable user interfaces may also be implemented, such as an audio interface utilizing speech prompts output via loudspeakers and user voice commands input via a microphone.
 30

[90] Fig. 8B is a detailed schematic block diagram of the processor 1305 and a “memory” 1334. The memory 1334 represents a logical aggregation of all the memory modules (including the HDD 1309 and semiconductor memory 1306) that can be accessed by the computer module 1301 in Fig. 8A.

5

[91] When the computer module 1301 is initially powered up, a power-on self-test (POST) program 1350 executes. The POST program 1350 is typically stored in a ROM 1349 of the semiconductor memory 1306 of Fig. 8A. A hardware device such as the ROM 1349 storing software is sometimes referred to as firmware. The POST program 1350 examines hardware within the computer module 1301 to ensure proper functioning and typically checks the processor 1305, the memory 1334 (1309, 1306), and a basic input-output systems software (BIOS) module 1351, also typically stored in the ROM 1349, for correct operation. Once the POST program 1350 has run successfully, the BIOS 1351 activates the hard disk drive 1310 of Fig. 8A. Activation of the hard disk drive 1310 causes a bootstrap loader program 1352 that is resident on the hard disk drive 1310 to execute via the processor 1305. This loads an operating system 1353 into the RAM memory 1306, upon which the operating system 1353 commences operation. The operating system 1353 is a system level application, executable by the processor 1305, to fulfil various high level functions, including processor management, memory management, device management, storage management, software application interface, and generic user interface.

[92] The operating system 1353 manages the memory 1334 (1309, 1306) to ensure that each process or application running on the computer module 1301 has sufficient memory in which to execute without colliding with memory allocated to another process. Furthermore, the different types of memory available in the system 1300 of Fig. 8A must be used properly so that each process can run effectively. Accordingly, the aggregated memory 1334 is not intended to illustrate how particular segments of memory are allocated (unless otherwise stated), but rather to provide a general view of the memory accessible by the computer system 1300 and how such is used.

[93] As shown in Fig. 8B, the processor 1305 includes a number of functional modules including a control unit 1339, an arithmetic logic unit (ALU) 1340, and a local or internal memory 1348, sometimes called a cache memory. The cache memory 1348 typically includes a number of storage registers 1344 - 1346 in a register section. One or more

35

internal busses 1341 functionally interconnect these functional modules. The processor 1305 typically also has one or more interfaces 1342 for communicating with external devices via the system bus 1304, using a connection 1318. The memory 1334 is coupled to the bus 1304 using a connection 1319.

5

[94] The application program 1333 includes a sequence of instructions 1331 that may include conditional branch and loop instructions. The program 1333 may also include data 1332 which is used in execution of the program 1333. The instructions 1331 and the data 1332 are stored in memory locations 1328, 1329, 1330 and 1335, 1336, 1337, respectively. Depending upon the relative size of the instructions 1331 and the memory locations 1328-1330, a particular instruction may be stored in a single memory location as depicted by the instruction shown in the memory location 1330. Alternately, an instruction may be segmented into a number of parts each of which is stored in a separate memory location, as depicted by the instruction segments shown in the memory locations 1328 and 1329.

15

[95] In general, the processor 1305 is given a set of instructions which are executed therein. The processor 1305 waits for a subsequent input, to which the processor 1305 reacts to by executing another set of instructions. Each input may be provided from one or more of a number of sources, including data generated by one or more of the input devices 1302, 1303, data received from an external source across one of the networks 1320, 1302, data retrieved from one of the storage devices 1306, 1309 or data retrieved from a storage medium 1325 inserted into the corresponding reader 1312, all depicted in Fig. 8A. The execution of a set of the instructions may in some cases result in output of data. Execution may also involve storing data or variables to the memory 1334.

20

25

[96] The disclosed transaction processing server 108 arrangements use input variables 1354, which are stored in the memory 1334 in corresponding memory locations 1355, 1356, 1357. The transaction processing server 108 arrangements produce output variables 1361, which are stored in the memory 1334 in corresponding memory locations 1362, 1363, 1364. Intermediate variables 1358 may be stored in memory locations 1359, 1360, 1366 and 1367.

30

[97] Referring to the processor 1305 of Fig. 8B, the registers 1344, 1345, 1346, the arithmetic logic unit (ALU) 1340, and the control unit 1339 work together to perform

35

sequences of micro-operations needed to perform “fetch, decode, and execute” cycles for every instruction in the instruction set making up the program 1333. Each fetch, decode, and execute cycle comprises:

- 5 a fetch operation, which fetches or reads an instruction 1331 from a memory location 1328, 1329, 1330;

a decode operation in which the control unit 1339 determines which instruction has been fetched; and

10

an execute operation in which the control unit 1339 and/or the ALU 1340 execute the instruction.

- 15 **[98]** Thereafter, a further fetch, decode, and execute cycle for the next instruction may be executed. Similarly, a store cycle may be performed by which the control unit 1339 stores or writes a value to a memory location 1332.

- 20 **[99]** Each step or sub-process in the processes as performed by the transaction processing server 108, is associated with one or more segments of the program 1333 and is performed by the register section 1344, 1345, 1347, the ALU 1340, and the control unit 1339 in the processor 1305 working together to perform the fetch, decode, and execute cycles for every instruction in the instruction set for the noted segments of the program 1333.

- 25 **[100]** It is to be understood that the structural context of the computer system 1300 (i.e., the transaction processing server 108) is presented merely by way of example. Therefore, in some arrangements, one or more features of the server 1300 may be omitted. Also, in some arrangements, one or more features of the server 1300 may be combined together. Additionally, in some arrangements, one or more features of the server 1300 may be split
30 into one or more component parts.

- [101]** Fig. 9 shows an alternative implementation of the transaction processing server 108 (i.e., the computer system 1300). In the alternative implementation, the transaction processing server 108 may be generally described as a physical device comprising at
35 least one processor 802 and at least one memory 804 including computer program codes.

The at least one memory 804 and the computer program codes are configured to, with the at least one processor 802, cause the transaction processing server 108 to facilitate the operations described in the methods 600 and 700. The transaction processing server 108 may also include a transaction processing module 806. The memory 804 stores computer program code that the processor 802 compiles to have each of the modules 806 and 808 perform their respective functions.

[102] With reference to Figs. 1 to 7, the transaction processing module 806 performs the function of communicating with the requestor device 102 and the provider device 104; and the acquirer server 106 and the issuer server 110 to respectively receive and transmit a request for insurance, as well as a transaction or travel request message. Further, the transaction processing module 806 may provide data and information associated with the request for insurance that are used for adaptively managing the request for insurance by the switching server 140, for example providing a confirmation that payment has been made for the request for insurance, providing data and information relating to the user making the request for insurance for underwriting purposes, or other similar types of data and information.

[103] Fig. 8C depicts a general-purpose computer system 1400, upon which the switching server 140 described can be practiced. The computer system 1400 includes a computer module 1401. An external Modulator-Demodulator (Modem) transceiver device 1416 may be used by the computer module 1401 for communicating to and from a communications network 1420 via a connection 1421. The communications network 1420 may be a wide-area network (WAN), such as the Internet, a cellular telecommunications network, or a private WAN. Where the connection 1421 is a telephone line, the modem 1416 may be a traditional “dial-up” modem. Alternatively, where the connection 1421 is a high capacity (e.g., cable) connection, the modem 1416 may be a broadband modem. A wireless modem may also be used for wireless connection to the communications network 1420.

[104] The computer module 1401 typically includes at least one processor unit 1405, and a memory unit 1406. For example, the memory unit 1406 may have semiconductor random access memory (RAM) and semiconductor read only memory (ROM). The computer module 1401 also includes an interface 1408 for the external modem 1416. In some implementations, the modem 1416 may be incorporated within the computer

module 1401, for example within the interface 1408. The computer module 1401 also has a local network interface 1411, which permits coupling of the computer system 1400 via a connection 1423 to a local-area communications network 1422, known as a Local Area Network (LAN). As illustrated in Fig. 8C, the local communications network 1422 may also couple to the wide network 1420 via a connection 1424, which would typically include a so-called “firewall” device or device of similar functionality. The local network interface 1411 may comprise an Ethernet circuit card, a Bluetooth® wireless arrangement or an IEEE 802.11 wireless arrangement; however, numerous other types of interfaces may be practiced for the interface 1411.

[105] The I/O interfaces 1408 may afford either or both of serial and parallel connectivity, the former typically being implemented according to the Universal Serial Bus (USB) standards and having corresponding USB connectors (not illustrated). Storage devices 1409 are provided and typically include a hard disk drive (HDD) 1410. Other storage devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may also be used. An optical disk drive 1412 is typically provided to act as a non-volatile source of data. Portable memory devices, such optical disks, USB-RAM, portable, external hard drives, and floppy disks, for example, may be used as appropriate sources of data to the system 1400.

[106] The components 1405 to 1412 of the computer module 1401 typically communicate via an interconnected bus 1404 and in a manner that results in a conventional mode of operation of the computer system 1400 known to those in the relevant art. For example, the processor 1405 is coupled to the system bus 1404 using a connection 1418. Likewise, the memory 1406 and optical disk drive 1412 are coupled to the system bus 1404 by connections 1419. Examples of computers on which the described arrangements can be practiced include IBM-PC's and compatibles, Sun Sparcstations, Apple or like computer systems.

[107] The methods 600 and 700, where performed by the switching server 140 may be implemented using the computer system 1400. The processes may be implemented as one or more software application programs 1433 executable within the computer system 1400. In particular, the methods 600 and 700 are effected by instructions (see corresponding component 1331 in Fig. 8B) in the software 1433 that are carried out within the computer system 1400. The software instructions may be formed as one or more code

modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part and the corresponding code modules performs the methods and a second part and the corresponding code modules manage a user interface between the first part and the user.

5

[108] The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer system 1400 from the computer readable medium, and then executed by the computer system 1400. A computer readable medium having such software or computer program recorded on the computer readable medium is a computer program product. The use of the computer program product in the computer system 1400 preferably effects an advantageous apparatus for a switching server 140.

[109] The software 1433 is typically stored in the HDD 1410 or the memory 1406. The software is loaded into the computer system 1400 from a computer readable medium, and executed by the computer system 1400. Thus, for example, the software 1433 may be stored on an optically readable disk storage medium (e.g., CD-ROM) 1425 that is read by the optical disk drive 1412. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer system 1400 preferably effects an apparatus for a switching server 140.

[110] In some instances, the application programs 1433 may be supplied to the user encoded on one or more CD-ROMs 1425 and read via the corresponding drive 1412, or alternatively may be read by the user from the networks 1420 or 1422. Still further, the software can also be loaded into the computer system 1400 from other computer readable media. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computer system 1400 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, optical disc, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 1401. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computer module 1401 include radio or infra-red transmission channels as well as

a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

5 **[111]** The second part of the application programs 1433 and the corresponding code modules mentioned above may be executed to implement one or more graphical user interfaces (GUIs) to be rendered or otherwise represented upon a display. Through manipulation of typically a keyboard and a mouse, a user of the computer system 1400 and the application may manipulate the interface in a functionally adaptable manner to
10 provide controlling commands and/or input to the applications associated with the GUI(s). Other forms of functionally adaptable user interfaces may also be implemented, such as an audio interface utilizing speech prompts output via loudspeakers and user voice commands input via a microphone.

15 **[112]** It is to be understood that the structural context of the computer system 1400 (i.e., the switching server 140) is presented merely by way of example. Therefore, in some arrangements, one or more features of the computer system 1400 may be omitted. Also, in some arrangements, one or more features of the computer system 1400 may be combined together. Additionally, in some arrangements, one or more features of the
20 computer system 1400 may be split into one or more component parts.

[113] Fig. 10 shows an alternative implementation of the switching server 140 (i.e., the computer system 1400). In the alternative implementation, switching server 140 may be generally described as a physical device comprising at least one processor 902 and at
25 least one memory 904 including computer program codes. The at least one memory 904 and the computer program codes are configured to, with the at least one processor 902, cause the switching server 140 to perform the operations described in the methods 600 and 700. The switching server 140 may also include an underwriting module 906, a data module 908, an insurance plan module 910, a calculation module 912 and a SLA module
30 914. The memory 904 stores computer program code that the processor 902 compiles to have each of the modules 906 to 914 perform their respective functions.

[114] With reference to Figs. 1 to 7, the SLA module 914 comprises information relating to SLAs for each of the plurality of service provider servers. This information can be utilized
35 by the calculation module 912 for determining a threshold and/or assigned weights for use

in adaptively managing a request for insurance. The SLA module 914 facilitates the function of determining whether a response is received from a first one of the plurality of service provider servers in accordance with a predetermined threshold, the threshold being one that is determined based on a service level agreement. The SLA module 914 further facilitates the function of sending one or more other requests to the first one of the plurality of service provider servers at a predetermined period of time after the determination that the response is not received, the predetermined period of time being one that is determined based on the service level agreement; and determining whether a response is received for each of the one or more other requests from the first one of the plurality of service provider servers in accordance with the predetermined threshold.

[115] With reference to Figs. 1 to 7, the underwriting module 906 performs the function of underwriting for a request for insurance based on data and information associated with the user making the request and the insurance product indicated in the request. The switching server 140 may proceed with adaptively managing the request for insurance upon confirmation by the underwriting module 906 that the user is eligible for the insurance product indicated in the request for insurance.

[116] With reference to Figs. 1 to 7, the insurance plan module 910 comprises data and information relating to one or more insurance products, corresponding insurance plans that make up each insurance product, as well as corresponding unique ID, identifier and service provider for each insurance plan (for example in the form of table 210 of Fig. 2). The insurance plan module facilitates the function of determining the plurality of service provider servers for providing an insurance product indicated in the request for insurance, the plurality of service provider servers being based on data identifying the insurance plan, as well as for determining an identifier associated with the insurance plan based on the data, wherein determining the plurality of service provider servers is based on the identifier.

[117] With reference to Figs. 1 to 7, the calculation module 912 performs the function of calculating a threshold and/or assigned weights for each service provider server 150 based on data and information relating to SLA agreements, contractual agreements, current status of each service provider server 150 and other similar types of data and information. The calculation module 912 facilitates the functions of assigning and reassigning weights to each of the plurality of service provider servers 150, as well as

facilitates the function of determining whether a response is received from a first one of the plurality of service provider servers in accordance with a predetermined threshold, the threshold being one that is determined based on a service level agreement. The calculation module 912 further facilitates the function of sending one or more other requests to the first one of the plurality of service provider servers at a predetermined period of time after the determination that the response is not received, the predetermined period of time being one that is determined based on the service level agreement; and determining whether a response is received for each of the one or more other requests from the first one of the plurality of service provider servers in accordance with the predetermined threshold.

[118] With reference to Figs. 1 to 7, the data module 908 performs the functions of receiving data and information from the requestor device 102, provider device 104, transaction processing server 108, a cloud and other sources of information to facilitate the methods 600 and 700. For example, the data module 908 may be configured to receive a request for insurance and/or data and information thereof from the requestor device 102, the provider device 104, transaction processing server 108 or other sources of information. The data module 908 may also be configured to receive a confirmation that payment has been made for the request for insurance, data and information relating to the user making the request for insurance for underwriting purposes, or other similar types of data and information from the transaction processing server 108, the requestor device 102, the provider device 104 or other sources of information.

[119] Fig. 8D depicts a general-purpose computer system 1500, upon which a combined transaction processing server 108 and switching server 140 described can be practiced. The computer system 1500 includes a computer module 1501. An external Modulator-Demodulator (Modem) transceiver device 1516 may be used by the computer module 1501 for communicating to and from a communications network 1520 via a connection 1521. The communications network 1520 may be a wide-area network (WAN), such as the Internet, a cellular telecommunications network, or a private WAN. Where the connection 1521 is a telephone line, the modem 1516 may be a traditional “dial-up” modem. Alternatively, where the connection 1521 is a high capacity (e.g., cable) connection, the modem 1516 may be a broadband modem. A wireless modem may also be used for wireless connection to the communications network 1520.

[120] The computer module 1501 typically includes at least one processor unit 1505, and a memory unit 1506. For example, the memory unit 1506 may have semiconductor random access memory (RAM) and semiconductor read only memory (ROM). The computer module 1501 also includes an interface 1508 for the external modem 1516. In some implementations, the modem 1516 may be incorporated within the computer module 1501, for example within the interface 1508. The computer module 1501 also has a local network interface 1511, which permits coupling of the computer system 1500 via a connection 1523 to a local-area communications network 1522, known as a Local Area Network (LAN). As illustrated in Fig. 8D, the local communications network 1522 may also couple to the wide network 1520 via a connection 1524, which would typically include a so-called "firewall" device or device of similar functionality. The local network interface 1511 may comprise an Ethernet circuit card, a Bluetooth® wireless arrangement or an IEEE 802.11 wireless arrangement; however, numerous other types of interfaces may be practiced for the interface 1511.

[121] The I/O interfaces 1508 may afford either or both of serial and parallel connectivity, the former typically being implemented according to the Universal Serial Bus (USB) standards and having corresponding USB connectors (not illustrated). Storage devices 1509 are provided and typically include a hard disk drive (HDD) 1510. Other storage devices such as a floppy disk drive and a magnetic tape drive (not illustrated) may also be used. An optical disk drive 1512 is typically provided to act as a non-volatile source of data. Portable memory devices, such optical disks, USB-RAM, portable, external hard drives, and floppy disks, for example, may be used as appropriate sources of data to the system 1500.

[122] The components 1505 to 1512 of the computer module 1501 typically communicate via an interconnected bus 1504 and in a manner that results in a conventional mode of operation of the computer system 1500 known to those in the relevant art. For example, the processor 1505 is coupled to the system bus 1504 using a connection 1518. Likewise, the memory 1506 and optical disk drive 1512 are coupled to the system bus 1504 by connections 1519. Examples of computers on which the described arrangements can be practised include IBM-PC's and compatibles, Sun Sparcstations, Apple or like computer systems.

[123] The steps of the methods 600 and 700 performed by the switching 140 and facilitated by the transaction processing server 108 may be implemented using the computer system 1500. For example, the steps of the method 600 and 700 as performed by the switching server 140 may be implemented as one or more software application programs 1533 executable within the computer system 1500. In particular, the steps of the methods 600 and 700 are effected by instructions (see corresponding component 1331 in Fig. 8B) in the software 1533 that are carried out within the computer system 1500. The software instructions may be formed as one or more code modules, each for performing one or more particular tasks. The software may also be divided into two separate parts, in which a first part and the corresponding code modules perform the steps of the methods 600 and 700 and a second part and the corresponding code modules manage a user interface between the first part and the user.

[124] The software may be stored in a computer readable medium, including the storage devices described below, for example. The software is loaded into the computer system 1500 from the computer readable medium, and then executed by the computer system 1500. A computer readable medium having such software or computer program recorded on the computer readable medium is a computer program product. The use of the computer program product in the computer system 1500 preferably effects an advantageous apparatus for a combined transaction processing server 108 and switching server 140.

[125] The software 1533 is typically stored in the HDD 1510 or the memory 1506. The software is loaded into the computer system 1500 from a computer readable medium, and executed by the computer system 1500. Thus, for example, the software 1533 may be stored on an optically readable disk storage medium (e.g., CD-ROM) 1525 that is read by the optical disk drive 1512. A computer readable medium having such software or computer program recorded on it is a computer program product. The use of the computer program product in the computer system 1500 preferably effects an apparatus for a combined transaction processing server 108 and switching server 140.

[126] In some instances, the application programs 1533 may be supplied to the user encoded on one or more CD-ROMs 1525 and read via the corresponding drive 1512, or alternatively may be read by the user from the networks 1520 or 1522. Still further, the software can also be loaded into the computer system 1500 from other computer readable

media. Computer readable storage media refers to any non-transitory tangible storage medium that provides recorded instructions and/or data to the computer system 1500 for execution and/or processing. Examples of such storage media include floppy disks, magnetic tape, optical disc, a hard disk drive, a ROM or integrated circuit, USB memory, a magneto-optical disk, or a computer readable card such as a PCMCIA card and the like, whether or not such devices are internal or external of the computer module 1501. Examples of transitory or non-tangible computer readable transmission media that may also participate in the provision of software, application programs, instructions and/or data to the computer module 1501 include radio or infra-red transmission channels as well as a network connection to another computer or networked device, and the Internet or Intranets including e-mail transmissions and information recorded on Websites and the like.

[127] The second part of the application programs 1533 and the corresponding code modules mentioned above may be executed to implement one or more graphical user interfaces (GUIs) to be rendered or otherwise represented upon a display. Through manipulation of typically a keyboard and a mouse, a user of the computer system 1500 and the application may manipulate the interface in a functionally adaptable manner to provide controlling commands and/or input to the applications associated with the GUI(s). Other forms of functionally adaptable user interfaces may also be implemented, such as an audio interface utilizing speech prompts output via loudspeakers and user voice commands input via a microphone.

[128] It is to be understood that the structural context of the computer system 1500 (i.e., combined transaction processing server 108 and switching server 140) is presented merely by way of example. Therefore, in some arrangements, one or more features of the server 1500 may be omitted. Also, in some arrangements, one or more features of the server 1500 may be combined together. Additionally, in some arrangements, one or more features of the server 1500 may be split into one or more component parts.

[129] Fig. 11 shows an alternative implementation of combined transaction processing server 108 and switching server 140 (i.e., the computer system 1500). In the alternative implementation, the combined transaction processing server 108 and switching server 140 may be generally described as a physical device comprising at least one processor 1002 and at least one memory 904 including computer program codes. The at least one

memory 1004 and the computer program codes are configured to, with the at least one processor 1002, cause the combined transaction processing server 108 and switching server 140 to perform the operations described in the steps of the methods 600 and 700. The combined transaction processing server 108 and switching server 140 may also include a transaction request processing module 806, an ensemble module 906, a data module 908, a collection module 910, a retraining module 912, a lookup module 914 (e.g. comprising the plurality of lookup models 262) and a matching module 916 (e.g. comprising the plurality of models 264). The memory 1004 stores computer program code that the processor 1002 compiles to have each of the modules 806 to 912 perform their respective functions.

[130] With reference to Figs. 1 to 7, the transaction processing module 806 performs the function of communicating with the requestor device 102 and the provider device 104; and the acquirer server 106 and the issuer server 110 to respectively receive and transmit a request for insurance, as well as a transaction or travel request message. Further, the transaction processing module 806 may provide data and information associated with the request for insurance that are used for adaptively managing the request for insurance by the switching server 140, for example providing a confirmation that payment has been made for the request for insurance, providing data and information relating to the user making the request for insurance for underwriting purposes, or other similar types of data and information.

[131] Further, it will be appreciated that the SLA module 914, underwriting module 906, insurance plan module 910, calculation module 912 and data module 908 as shown in Figure 11 function in the same manner as their counterparts described in Figure 10.

[132] It will be appreciated by a person skilled in the art that numerous variations and/or modifications may be made to the present disclosure as shown in the specific embodiments without departing from the scope of the specification as broadly described. The present embodiments are, therefore, to be considered in all respects to be illustrative and not restrictive.

CLAIMS

What is claimed is:

1. A method for adaptively managing a request for insurance, comprising:

5 sending, by a switching server, the request to a first one of a plurality of service provider servers;

 determining, by the switching server, whether a response is received from the first one of the plurality of service provider servers, and

10 sending, by the switching server, the request to a second one of the plurality of service provider servers based on the determination of whether a response is received from the first one of the plurality of service provider servers.

2. The method according to claim 1, further comprising determining, by the switching server, whether the request is eligible based on an underwriting process, wherein the request is sent to the first one of the plurality of service provider servers when the request
15 is determined to be eligible.

3. The method according to claim 1, further comprising determining, by the switching server, whether to send the request to a second one of the plurality of service provider servers based on the determination of whether a response is received from the first one
20 of the plurality of service provider servers.

4. The method according to claim 1, wherein the plurality of service provider servers is determined based on data relating to an insurance product indicated in the request.

25 5. The method according to claim 4, further comprising determining, by the switching server, an identifier associated with the insurance product based on the data, wherein determining the plurality of service provider servers is based on the identifier.

30 6. The method according to claim 1, further comprising determining, by the switching server, the first one of the plurality of service provider servers based on a weight assigned to each of the plurality of service provider servers.

7. The method according to claim 6, wherein determining whether a response is received from the first one of the plurality of service provider servers is in accordance with a
35 predetermined threshold, the predetermined threshold being based on a service level

agreement, the predetermined threshold being a maximum number of times that a response is not received, or a number of times that a response is not received within a predetermined period of time.

8. The method according to claim 7, further comprising reassigning, by the switching server, weights among the plurality of service provider servers without the first one of the plurality of service provider servers if it is determined that the response is not received from the first one of the plurality of service provider servers in accordance with the predetermined threshold.

9. The method according to claim 7, further comprising:

 sending, by the switching server, one or more other requests to the first one of the plurality of service provider servers at a predetermined period of time after the determination that the response is not received, the predetermined period of time being one that is determined based on the service level agreement; and

 determining, by the switching server, whether a response is received for each of the one or more other requests from the first one of the plurality of service provider servers in accordance with the predetermined threshold, the predetermined threshold further being a maximum number of times that a response is received, or a number of times that a response is received within a predetermined period of time.

10. The method according to claim 9, further comprising reassigning, by the switching server, weights among the plurality of service provider servers with the first one of the plurality of service provider servers if it is determined that the response is received for each of the one or more other requests in accordance with the predetermined threshold.

11. A system for adaptively managing a request for insurance, comprising:

 at least one processor; and

 at least one memory including computer program code;

 the at least one memory and the computer program code configured to, with the at least one processor, cause the system at least to:

 send the request to a first one of a plurality of service provider servers when the request for insurance is determined to be eligible;

 determine whether a response is received from the first one of the plurality of service provider servers, and

send the request to a second one of a plurality of service provider servers based on the determination of whether a response is received from the first one of the plurality of service provider servers.

5 12. The system according to claim 11, further configured to determine whether the request is eligible based on an underwriting process, wherein the request is sent to the first one of the plurality of service provider servers when the request is determined to be eligible.

10 13. The system according to claim 11, further configured to determine whether to send the request to a second one of the plurality of service provider servers based on the determination of whether a response is received from the first one of the plurality of service provider servers.

15 14. The system according to claim 11, wherein the plurality of service provider servers is determined based on relating to an insurance product indicated in the request.

20 15. The system according to claim 14, further configured to determine an identifier associated with the insurance product based on the data, wherein determining the plurality of service provider servers is based on the identifier.

16. The system according to claim 11, further configured to determine the first one of the plurality of service provider servers based on a weight assigned to each of the plurality of service provider servers.

25 17. The method according to claim 16, wherein determining whether a response is received from the first one of the plurality of service provider servers is in accordance with a predetermined threshold, the predetermined threshold being based on a service level agreement, the predetermined threshold being a maximum number of times that a response is not received, or a number of times that a response is not received within a
30 predetermined period of time.

35 18. The system according to claim 17, further configured to reassign weights among the plurality of service provider servers without the first one of the plurality of service provider servers if it is determined that the response is not received from the first one of the plurality of service provider servers in accordance with the predetermined threshold.

19. The system according to claim 18, further configured to:

send one or more other requests to the first one of the plurality of service provider servers at a predetermined period of time after the determination that the response is not received, the predetermined period of time being one that is determined based on the service level agreement; and

determine whether a response is received for each of the one or more other requests from the first one of the plurality of service provider servers in accordance with the predetermined threshold, the predetermined threshold further being a maximum number of times that a response is received, or a number of times that a response is received within a predetermined period of time.

20. The system according to claim 19, further configured to reassign weights among the plurality of service provider servers with the first one of the plurality of service provider servers if it is determined that the response is received for each of the one or more other requests.