

# US Patent & Trademark Office

## Patent Public Search | Text View

---

United States Patent	12388703
Kind Code	B2
Date of Patent	August 12, 2025
Inventor(s)	Pawaskar; Sachin

---

### Systems and methods for subscribing topics and registering computer server event notifications

---

#### Abstract

Systems and methods are disclosed for secure transmission of computer server event notifications, including receiving a topic subscription request from a partner, registering the partner topic subscription request, obtaining a new computer server event to report, determining a subscribed partner subscribed to computer server events of a same type as the new computer server event, transmitting an event notification to the subscribed partner, and registering the event notification.

---

<b>Inventors:</b>	<b>Pawaskar; Sachin (Mason, OH)</b>
<b>Applicant:</b>	<b>Worldpay, LLC (Symmes Township, OH)</b>
<b>Family ID:</b>	<b>1000008749716</b>
<b>Assignee:</b>	<b>Worldpay, LLC (Symmes Township, OH)</b>
<b>Appl. No.:</b>	<b>18/520925</b>
<b>Filed:</b>	<b>November 28, 2023</b>

#### Prior Publication Data

Document Identifier	Publication Date
US 20240097964 A1	Mar. 21, 2024

#### Related U.S. Application Data

continuation parent-doc US 18303156 20230419 US 11870636 child-doc US 18520925  
continuation parent-doc US 17843249 20220617 US 11665045 20230530 child-doc US 18303156  
continuation parent-doc US 17242367 20210428 US 11398942 20220726 child-doc US 17843249  
continuation parent-doc US 16811057 20200306 US 11025479 20210601 child-doc US 17242367  
continuation parent-doc US 15368130 20161202 US 10630534 20200421 child-doc US 16811057

---

## Publication Classification

**Int. Cl.:** H04L41/06 (20220101); H04L41/00 (20220101); H04L41/0604 (20220101)

**U.S. Cl.:**

**CPC** H04L41/06 (20130101); H04L41/0604 (20130101); H04L41/24 (20130101);

## Field of Classification Search

**CPC:** G06F (9/542); H04L (51/24); H04L (9/0819); H04L (9/0861); H04L (9/321)

---

## References Cited

### U.S. PATENT DOCUMENTS

Patent No.	Issued Date	Patentee Name	U.S. Cl.	CPC
7106742	12/2005	Frisch	370/244	H04L 1/0061
8165519	12/2011	Sato et al.	N/A	N/A
8311520	12/2011	Choi et al.	N/A	N/A
8464070	12/2012	Kim	N/A	N/A
8566235	12/2012	Katz et al.	N/A	N/A
8825798	12/2013	Johnson et al.	N/A	N/A
10110536	12/2017	Vincent	N/A	N/A
10133716	12/2017	Konnola et al.	N/A	N/A
10320603	12/2018	Pawaskar	N/A	N/A
10630534	12/2019	Pawaskar	N/A	N/A
10721328	12/2019	Chen et al.	N/A	N/A
10757154	12/2019	Jacobs et al.	N/A	N/A
11398942	12/2021	Pawaskar	N/A	N/A
2002/0010803	12/2001	Oberstein et al.	N/A	N/A
2002/0126811	12/2001	Aschir et al.	N/A	N/A
2003/0212834	12/2002	Potter et al.	N/A	N/A
2004/0078440	12/2003	Potter et al.	N/A	N/A
2004/0122892	12/2003	Brittenham et al.	N/A	N/A
2006/0161915	12/2005	Barr	717/174	G06F 8/61
2007/0050630	12/2006	Kumar et al.	N/A	N/A
2007/0143473	12/2006	Wada	N/A	N/A
2007/0165615	12/2006	Shin et al.	N/A	N/A
2007/0282959	12/2006	Stern	N/A	N/A
2010/0122136	12/2009	Korndewal	714/E11.131	F16K 11/078
2010/0280909	12/2009	Zhang et al.	N/A	N/A
2011/0087732	12/2010	Lakshmanan et al.	N/A	N/A
2011/0161987	12/2010	Huang et al.	N/A	N/A
2011/0214131	12/2010	Luna	N/A	N/A
2011/0231781	12/2010	Betzler et al.	N/A	N/A
2011/0314144	12/2010	Goodman	N/A	N/A
2012/0236719	12/2011	Kikuzuki et al.	N/A	N/A
2013/0006989	12/2012	Plattner et al.	N/A	N/A
2013/0179989	12/2012	Garst, Jr.	726/27	G06Q 20/1235

2013/0212599	12/2012	Giampaolo et al.	N/A	N/A
2013/0219459	12/2012	Bradley	726/1	H04L 63/104
2013/0268994	12/2012	Cooper et al.	N/A	N/A
2014/0025410	12/2013	Churchman et al.	N/A	N/A
2014/0165200	12/2013	Singla	N/A	N/A
2014/0195476	12/2013	Sxhmidt	N/A	N/A
2014/0208394	12/2013	Goodwin	N/A	N/A
2015/0186888	12/2014	Katz et al.	N/A	N/A
2015/0248276	12/2014	Chan	N/A	N/A
2015/0304265	12/2014	Vincent	N/A	N/A
2015/0350092	12/2014	Kwon et al.	N/A	N/A
2016/0028591	12/2015	Johnson	709/225	H04L 41/145
2016/0124742	12/2015	Rangasamy et al.	N/A	N/A
2016/0255165	12/2015	Hayashi et al.	N/A	N/A
2016/0300223	12/2015	Grey et al.	N/A	N/A
2016/0346673	12/2015	Maggio	N/A	C07D 401/14
2016/0364961	12/2015	Cruz-Hernandez et al.	N/A	N/A
2017/0163752	12/2016	Kaledhonkar	N/A	H04L 67/306
2017/0242765	12/2016	Dreyer et al.	N/A	N/A
2017/0346807	12/2016	Blasi	N/A	H04L 63/0442
2019/0014076	12/2018	Huang et al.	N/A	N/A
2019/0266031	12/2018	Wu et al.	N/A	N/A

---

*Primary Examiner:* Duong; Oanh

*Attorney, Agent or Firm:* Bookoff McAndrews, PLLC

---

## Background/Summary

CROSS-REFERENCE TO RELATED APPLICATION(S) (1) This patent application is a continuation of and claims the benefit of priority to U.S. application Ser. No. 18/303,156, filed on Apr. 19, 2023, which is a continuation of and claims the benefit of priority to U.S. application Ser. No. 17/843,249, filed on Jun. 17, 2022, now U.S. Pat. No. 11,665,045, which is a continuation of and claims the benefit of priority to U.S. application Ser. No. 17/242,367, filed on Apr. 28, 2021, now U.S. Pat. No. 11,398,942, which is a continuation of and claims the benefit of priority to U.S. application Ser. No. 16/811,057, filed on Mar. 6, 2020, now U.S. Pat. No. 11,025,479, which is a continuation of and claims the benefit of priority to U.S. application Ser. No. 15/368,130, filed on Dec. 2, 2016, now U.S. Pat. No. 10,630,534, the entireties of which are incorporated herein by reference.

## TECHNICAL FIELD

(1) The present disclosure relates generally to the field of inter-system computer communications and, more particularly, to provide secure transmission and subscription of computer server event notifications between systems.

## BACKGROUND

(2) In distributed computing systems, such as one supporting collaborative practices including, for example, financial services and electronic payment transactions, events arising within one distributed partner's environment may be significant to other distributed partners. Events may be related to, for example, business processes, data synchronization, updating status of records, error

conditions, etc. Thus, it is important that such distributed systems provide mechanisms for notifying such events to other partners within the distributed computing system. Existing distributed computing systems rely, for example, on file transfers or on polling application programming interfaces (APIs), etc. for computer server event notifications or calling partner APIs. However, these mechanisms suffer from high computing resource costs, delays and lack of security, and may be subject to changes in underlying APIs that force changes in other parts of the distributed system.

(3) Accordingly, there is a need for methods and systems for providing transmission and registration of computer server event notifications between disparate systems that are efficient, secure and scalable.

#### SUMMARY

(4) According to certain aspects of the present disclosure, systems and methods are disclosed for providing secure transmission of computer server event notifications.

(5) In one embodiment, a computer-implemented method is disclosed for secure transmission of computer server event notifications. The method includes: receiving a topic subscription request from a partner over a computer network, registering the partner topic subscription request in a database, obtaining a new computer server event to report, determining, using a hardware processor, a subscribed partner subscribed to computer server events of a same type as the new computer server event, transmitting, over the computer network, an event notification to the subscribed partner, and registering the event notification in the database.

(6) In accordance with another embodiment, a system is disclosed for secure transmission of computer server event notifications. The system comprises: a memory having processor-readable instructions stored therein; and a processor configured to access the memory and execute the processor-readable instructions, which when executed by the processor configures the processor to perform a plurality of functions, including functions to: receive a topic subscription request from a partner over a computer network, register the partner topic subscription request in a database, obtain a new computer server event to report, determine, using the processor, a subscribed partner subscribed to computer server events of a same type as the new computer server event, transmit, over the computer network, an event notification to the subscribed partner, and register the event notification in the database.

(7) In accordance with another embodiment, a non-transitory machine-readable medium is disclosed that stores instructions that, when executed by a computer, cause the computer to perform a method for secure transmission of computer server event notifications. The method includes: receiving a topic subscription request from a partner over a computer network, registering the partner topic subscription request in a database, obtaining a new computer server event to report, determining, using a hardware processor, a subscribed partner subscribed to computer server events of a same type as the new computer server event, transmitting, over the computer network, an event notification to the subscribed partner, and registering the event notification in the database.

(8) Additional objects and advantages of the disclosed embodiments will be set forth in part in the description that follows, and in part will be apparent from the description, or may be learned by practice of the disclosed embodiments. The objects and advantages on the disclosed embodiments will be realized and attained by means of the elements and combinations particularly pointed out in the appended claims.

(9) It is to be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the detailed embodiments, as claimed.

(10) It may be understood that both the foregoing general description and the following detailed description are exemplary and explanatory only and are not restrictive of the invention, as claimed.

---

## Description

### BRIEF DESCRIPTION OF THE DRAWINGS

- (1) The accompanying drawings, which are incorporated in and constitute a part of this specification, illustrate exemplary embodiments of the present disclosure and together with the description, serve to explain the principles of the disclosure.
- (2) FIG. 1 depicts a block diagram of a system for secure transmission of computer server event notifications, according to one or more embodiments.
- (3) FIG. 2 is a flow chart depicting an example process for secure transmission of computer server event notifications, according to one or more embodiments.
- (4) FIG. 3 depicts a process flow diagram of an example method for secure transmission of computer server event notifications, according to one or more embodiments.
- (5) FIG. 4 is a flow chart depicting an example process for secure transmission of computer server event notifications, according to one or more embodiments.
- (6) FIG. 5 depicts a process flow diagram of an example method for secure transmission of computer server event notifications, according to one or more embodiments.
- (7) FIG. 6 depicts a block diagram of computer server event reporting sequences in secure transmission of computer server event notifications, according to one or more embodiments.
- (8) FIG. 7 depicts a block diagram of computer server event reporting sequences in secure transmission of computer server event notifications, according to one or more embodiments.
- (9) FIG. 8 is a flow chart depicting an example process for secure transmission of computer server event notifications, according to one or more embodiments.
- (10) FIG. 9 is a flow chart depicting an example process for secure transmission of fraud alert notifications, according to one or more embodiments.
- (11) FIG. 10 is a flow chart depicting an example process for secure transmission of underwriting notifications, according to one or more embodiments.
- (12) FIG. 11 is a flow chart depicting an example process for secure transmission of risk monitoring notifications, according to one or more embodiments.
- (13) FIG. 12 is a flow chart depicting an example process for secure transmission of payment refund notifications, according to one or more embodiments.
- (14) FIG. 13 depicts a process flow diagram of an example method for secured information exchange of computer server event notifications between notification server and subscribing partner, according to one or more embodiments.
- (15) FIG. 14 is a flow chart depicting an example process for building a signed notification payload that will be exchanged between notification server and subscribing partner.
- (16) FIG. 15 depicts a process flow diagram of an example method for secure transmission of computer server event notifications, according to one or more embodiments.
- (17) FIG. 16 is a flow chart depicting an example process for secure transmission of computer server event notifications, according to one or more embodiments.
- (18) FIG. 17 is a flow chart depicting an example process for secure transmission of computer server event notifications, according to one or more embodiments.

### DETAILED DESCRIPTION

- (19) While principles of the present disclosure are described herein with reference to illustrative embodiments for particular applications, it should be understood that the disclosure is not limited thereto. Those having ordinary skill in the art and access to the teachings provided herein will recognize additional modifications, applications, embodiments, and substitution of equivalents all fall within the scope of the embodiments described herein. Accordingly, the invention is not to be considered as limited by the foregoing description.
- (20) Various non-limiting embodiments of the present disclosure will now be described to provide

an overall understanding of the principles of the structure, function, and use of systems and methods disclosed herein for installing and managing point of interaction devices within a merchant environment.

(21) As described above, existing methods for computer server event notifications in distributed computing systems may suffer from high computing resource costs, high maintenance costs, and lack of security. Thus, the embodiments of the present disclosure are directed to providing scalable and secure systems and methods for transmission of computer server event notifications.

(22) One or more examples of these non-limiting embodiments are illustrated in the selected examples disclosed and described in detail with reference to FIGS. **1-16** in the accompanying drawings. Those of ordinary skill in the art will understand that systems and methods specifically described herein and illustrated in the accompanying drawings are non-limiting embodiments. The features illustrated or described in connection with one non-limiting embodiment may be combined with the features of other non-limiting embodiments. Such modifications and variations are intended to be included within the scope of the present disclosure.

(23) Turning to FIG. **1**, in a distributed computing system, multiple computing systems may receive notifications of computer server events from other connected computing systems. For example, one or more partner computing systems **110** may receive computer server event notifications from a notification server **120**. In addition to general computing systems, partner computing systems may include specialized computing systems. For example, in financial services systems, the partner computing systems, may include, for example, underwriting services systems **140** and contractual adjustment pricing systems (CAPS) **150**, etc. Communication of the computer server event notification may be by way of a service gateway **130**. Service gateway **130** may provide secure communication between notification server **120** and the partner computing systems **110**, **140**, **150**. Interaction between service gateway **130** and the partner computing systems **110**, **140**, **150** may be according to specified APIs providing, for example, topic subscription, notification messaging, event publication by partner computing systems, and notification administration services, etc. These APIs will be discussed in further detail below. Notification server **120** may store information about, for example, partners, subscriptions, events, notifications, etc., in a database **160**. Although database **160** is depicted as a single database, it is to be appreciated that multiple databases **160** may be employed. For example, separate databases and/or tables may be provided different types of events. Separation of databases and/or tables for an event type may facilitate auditing or compliance reporting for events of a certain type. Such separation may also improve performance of notification system by isolating high-frequency event types from low-frequency event types.

(24) FIG. **2** is a flow chart depicting an example process for secure transmission of computer server event notifications, according to one or more embodiments. FIG. **3** depicts a process flow diagram of an example method for secure transmission of computer server event notifications, according to one or more embodiments. As shown in FIGS. **2** and **3**, a partner distributed computing system, such as partner computing system **110** depicted in FIG. **1**, may communicate directly with a server, such as notification server **120** depicted in FIG. **1**, without mediation by an additional gateway system, such as service gateway **130** depicted in FIG. **1**. However, routing communications between the host system and the partners via the service gateway may reduce the number of rules that have to be in place for network pathways.

(25) As shown in FIGS. **2** and **3**, in operation **210**, a server, such as notification server **120** depicted in FIG. **1**, may receive a the topic subscription registration request from a partner distributed computing system, such as partner computing system **110** depicted in FIG. **1**. As shown in FIG. **3**, the topic subscription registration request may include a license, and the server may verify a signature of the license (operation **304**) and extract an API key from the license (operation **306**). The server may then submit the registration request to an appropriate API within the server using the API key at operation **308**. In operation **230**, the server may send the registered partner distributed computing system a notification of events. In operation **240**, the server may receive a

request for notification details from the partner. As shown in FIG. 3, the notification details request may include a license, and the server may verify a signature of the license (operation **314**) and extract an API key from the license (operation **316**) to authenticate, authorize, and validate claims. The server may then submit the notification details request to an appropriate API within the server using the API key at operation **318**. In operation **250**, the server may provide notification details to a partner. In operation **260**, the server may receive a notification acknowledgment from a partner. As shown in FIG. 3, the acknowledgement may include a license, and the server may verify a signature of the license (operation **324**) and extract an API key from the license (operation **326**). In operation **270**, the server may mark the notification as acknowledged. As shown in FIG. 3, the notification may be marked as acknowledged through an appropriate API within the server using license assigned to subscriber with appropriate claims. Acknowledgement of the notification may help facilitate auditing and compliance of an enterprise. Acknowledgement of the notification may also provides the partner sufficient time to process the notification

(26) FIG. 4 is a flow chart depicting an example process for secure transmission of computer server event notifications, according to one or more embodiments. FIG. 5 depicts a process flow diagram of an example method for secure transmission of computer server event notifications, according to one or more embodiments. As shown in FIGS. 4 and 5, communication between the server, such as notification server **120** depicted in FIG. 1, and a partner distributed computing system, such as partner computing system **110** depicted in FIG. 1, may be mediated by a service gateway, such as the service gateway **130** depicted in FIG. 1. The service gateway may provide additional advantages, such as, for example, security for transmitted notifications, acknowledgements, etc., or abstraction for APIs published by the notification server, the partner systems, or other components within a distributed computing system. That is, the service gateway may publish an API that is equivalent to an API published by the notification server. The partner system, thus, may interface to the service gateway API, as opposed to the notification server API. Interfacing with the service gateway API may allow the notification server API to be modified without disturbing the implementation of the partner system. Similar API abstractions may be published for APIs published by partner systems, such as, for example, partners **110**, underwriting service **140** or contractual adjustment pricing system (CAPS) **150** depicted in FIG. 1. Security protocols provided by the service gateway and notification server may include, for example, message authentication codes (MAC), JavaScript Object Notation (JSON) Web Tokens (JWT), or secure Hypertext Transfer Protocol Secure (HTTPS), etc. Additional security aspects of the service gateway will be discussed in greater detail below with respect to FIGS. 13-16.

(27) Such secure transmission of computer server event notifications, may provide benefits for partners. For example, topic notification to partners can be turned off temporarily or permanently depending on business requirements over time.

(28) As shown in FIGS. 4 and 5, in operation **405**, a server, such as notification server **120** depicted in FIG. 1, may receive a subscription request from a partner distributed computing system, such as partner computing system **110** depicted in FIG. 1, the service gateway may validate if subscribing partner is authorized to receive notification of requested event type and in operation **410**, the server may register the partner topic subscription in a database, such as database **160** depicted in FIG. 1. The registration request from the partner distributed computing system may be submitted by way of a service gateway, such as the service gateway **130** depicted in FIG. 1. In operation **415**, the server may wait for an event to report from the database **160**. For example, as shown in FIG. 5, in operation **502**, notification server **120** may periodically poll database **160** for new events. If one or more new events are found, database **160** may return them to notification server **120** in operation **504**. Notification server **120** may then, in operation **506**, create a notification ID for the new event and store the notification ID in database **160**. In operation **420**, the server may query the database for partners subscribed to events of the same type as the new event. Notification server **120** may assign a notification identification to events. In operation **425**, the server may transmit an event

notification to each subscribed partner. The event notification may be transmitted via a service gateway, such as the service gateway **130** depicted in FIG. **1**. As shown in FIG. **5**, the partner may send receipt of the event notification in operation **514**. The receipt may be sent via a service gateway in operation **516**. Upon receiving the receipt from the partner, the server may mark the notification in the database as sent in operation **430**. For example, the event notification may be registered with a status of “sent,” as shown in FIG. **5**. If a receipt is not received from the partner, as in operation **520**, the server may increment a counter of the number of attempts to deliver the notification to the partner in operation **522**. The number of attempts to deliver the notification may be configured in the database for each topic subscription. This may be used for later reporting or auditing of the notification server. If the number of attempts to deliver the notification exceeds a predetermined threshold, then the server may determine that the notification has permanently failed. The server may further attempt to inform the affected partner system of the failure. This may be done, for example, by accessing an API published by the partner system or by transmitting a message to an administrator of the affected partner system or by providing APIs by host system to give details of notifications that failed. The server may hold subsequent notifications until the affected partner can be verified as available to receive notifications. Alternatively, the server may proceed with subsequent notifications and attempt to re-send the failed notification upon the successful completion of a subsequent notification. In operation **435**, the server may receive a request for notification details from the subscribed partner via the service gateway, and in operation **440**, the server may transmit notification details to requesting subscribed partner. The notification details may be transmitted to the partner via the service gateway. In operation **445**, the server may receive an acknowledgment of the event report from subscribed partner via the service gateway. Finally, in operation **450**, the server may update the event notification status to reflect the acknowledgment by the partner. For example, as shown in FIG. **5**, the server may set the notification status as “complete.”

(29) FIGS. **6** and **7** depict block diagrams of computer server event reporting sequences in secure transmission of computer server event notifications, according to one or more embodiments.

(30) As shown in FIGS. **6** and **7**, a server, such as notification server **120** depicted in FIG. **1** may transmit multiple events with each notification to a partner, such as partner **110** depicted in FIG. **1**. Grouping multiple events in a notification may reduce chattiness between the partners and the host system. The number of events grouped within each notification may be determined, for example, according to event type, partner preference settings, or notification server settings. In one or more embodiments, the number of events to be grouped in a notification may be based on sending a notification when a predetermined number of events have been received for notification. For example, a notification may be sent when five events have been received for a given event type. However, the number of events may be determined separately for each event type and for each partner. Further, the number of events may be determined differently depending on a time of day or day of the week. For example, a smaller or larger number of events may be reported in periods of low activity, such as overnight. In one or more embodiments, the number of events to be transmitted in a notification may be based on sending a notification per each predetermined period of time. For example, a notification may be sent every five minutes when at least one event has been received for notification. However, the length of the period of time may be determined separately for each event type and for each partner. Further, the length of the period of time may be determined differently depending on a time of day or day of the week. For example, a shorter or longer period of time may be used in periods of low activity, such as overnight. In addition, a predetermined maximum threshold may be set for the number of events transmitted in a single notification. If a maximum threshold is reached before a notification is to be transmitted based on a time period, then the notification may be transmitted early. For example, if a time period is set for five minutes and a threshold is set at 1,000 events, a notification may be transmitted early if, for example, 1,001 events are received for reporting within three minutes of the five-minute period.



Once a notification has been transmitted, the period of time may be restarted or a next notification may be transmitted at the end of the original time period.

(31) For example, as shown in FIG. 6, Events **1-6** may be transmitted to the partner in three notifications. For example, in operation **602**, Event **1** may be transmitted to the partner in Notification **1** by operation **602**, Events **2-4** may be transmitted to the partner in Notification **2** by operation **608**, and Events **5** and **6** may be transmitted to the partner in Notification **3** by operation **614**. As shown in FIG. 6, each notification received by the partner may, in turn, be fetched and acknowledged by the partner. For example, Notification **1** may be fetched by the partner in operation **604** and acknowledged by the partner in operation **606**. Notification **2** may be fetched by the partner in operation **610** and acknowledged by the partner in operation **612**. Notification **3** may be fetched by the partner in operation **616** and acknowledged by the partner in operation **622**. However, an acknowledgment of a notification from the partner may not be received by the server. In this case, the server may attempt to resend the unacknowledged notification. For example, as shown in FIG. 6, if the acknowledgment of Notification **2** is not received, then the server may resend Notification **2** to the partner in operation **618**. The partner may then acknowledge Notification **2** in operation **620**.

(32) Alternatively, if transmission of a notification to a partner fails, then the server may detect a failure of the partner to acknowledge the notification and may resend the notification. For example, as shown in FIG. 7, if the transmission of Notification **2** in operation **708** fails to be delivered to the partner, then the server may resend Notification **2** combined with Notification **3** in operation **710**. The partner may then fetch the combined Notifications **3** and **2** at operation **712** and acknowledge the combined Notifications **3** and **2** at operation **714**.

(33) FIG. 8 is a flow chart depicting an example process for secure transmission of computer server event notifications, according to one or more embodiments. In operation **820**, a server, such as notification server **120** depicted in FIG. 1, may determine if there one or more new events to report in a database, such as database **160** depicted in FIG. 1. If there are none, then the server may wait for a predetermined period of time before repeating the determination. The predetermined period of time may vary based on the type of event or other settings for the server. For example, the predetermined time period may be shorter for event types that occur frequently or may be longer for event types that occur infrequently. The predetermined period of time may also vary based on the time of day or the day of the week, etc. For example, the predetermined time period may be shorter at times that events occur more frequently or may be longer at times that events occur less frequently. If there are one or more new events to be reported, then in operation **830** the server may query the database for one or more partners, such as partners **110** depicted in FIG. 1, subscribed to events of same types as new events. In operation **840**, the server may generate, for each subscribed partner, an event report of all events for the subscribed partner, and in operation **850**, the server may transmit each event report to the subscribed partner. The event report may be transmitted via service gateway, such as service gateway **130**. In operation **860**, the server may register the event notification for each reported event in the database.

(34) The process of reporting event notifications from a server to a partner may vary depending on the type of event notification to be reported. For example, events may be related to fraud alerts, underwriting events, risk monitoring, or payment refunds, etc. FIGS. 9-12 depict example processes for transmitting different types of event notifications.

(35) FIG. 9 is a flow chart depicting an example process for secure transmission of fraud alert notifications, according to one or more embodiments. In operation **920**, a server, such as notification server **120** depicted in FIG. 1, may determine if one or more new fraud alert events to report are stored in a database, such as database **160** depicted in FIG. 1. If no fraud alert events are stored in the database, then the server may wait for a predetermined period of time before repeating the determination. If fraud alert events to report are stored in the database, then in operation **930** the server may query the database for one or more partners, such as partners **110** depicted in FIG. 1,

subscribed to events related to fraud alerts. In operation **940**, the server may generate, for each subscribed partner, notification related to fraud alerts for the subscribed partner, and in operation **960**, the server may transmit notification to the subscribed partner. The event notification may be transmitted via service gateway, such as service gateway **130**. In operation **950**, the server may register the notification for events related to a specific type of fraud alerts in the database. For some event types, including events related to fraud alerts, the server may track the notification and acknowledgement of the reported events. Accordingly, in operation **970**, following the reporting of an event related to fraud alerts, the server may increment a number of attempted notifications for the fraud alert events. If receipt of the notification related to fraud alerts is not received at operation **980**, then the server may return to operation **960** to re-transmit the unacknowledged event report related to fraud alerts to the subscribed partner. Otherwise, if the reporting of an event related to fraud alerts is acknowledged at operation **980**, then at operation **990**, the server may register the event report related to fraud alerts as completed in the database.

(36) In addition to events stored in a database, such as database **160** depicted in FIG. **1**, events may be published by an internal system such as, for example, an underwriting service **140** or a contractual adjustment pricing system (CAPS) **150** depicted in FIG. **1**. For example, underwriting service **140** may publish events related to underwriting events or risk monitoring events, etc. FIG. **10** is a flow chart depicting an example process for secure transmission of underwriting notifications, according to one or more embodiments. In operation **1005**, a server, such as notification server **120** depicted in FIG. **1**, may receive a registration request from a partner distributed computing system, such as partner computing system **110** depicted in FIG. **1**, and in operation **1010**, the server may register the partner topic subscription in a database, such as database **160** depicted in FIG. **1**. The registration request from the partner distributed computing system may be submitted by way of a service gateway, such as the service gateway **130** depicted in FIG. **1**. In operation **1015**, the server may determine if one or more underwriting events to report are stored in a database, such as database **160** depicted in FIG. **1**. If no underwriting events are stored in the database, then the server may wait for a predetermined period of time before repeating the determination. If underwriting events to report are stored in the database, then in operation **1020** the server may query the database for one or more partners, such as partners **110** depicted in FIG. **1**, subscribed to underwriting events. In operation **1025**, the server may generate, for each subscribed partner, a notification identification of all underwriting events for the subscribed partner, and in operation **1035**, the server may transmit each underwriting notification identification to the subscribed partner. The notification identification may be transmitted via service gateway, such as service gateway **130**. In operation **1030**, the server may create the event notification in the database. In operation **1040**, the server may receive a request for notification details from the subscribed partner via the service gateway, and in operation **1045**, the server may transmit notification details to requesting subscribed partner. The notification details may be transmitted to the partner via the service gateway. In operation **1050**, the server may receive an acknowledgment of the notification report from subscribed partner via the service gateway. In operation **1055**, the server may update the notification status to reflect the acknowledgment by the partner. In operation **1060**, the server may receive underwriting events from an external underwriting service, such as underwriting service **140** depicted in FIG. **1**. The event details may be transmitted to the partner via the service gateway. In operation **1065**, the server may create the new underwriting event in the database.

(37) FIG. **11** is a flow chart depicting an example process for secure transmission of risk monitoring notifications, according to one or more embodiments. In operation **1105**, a server, such as notification server **120** depicted in FIG. **1**, may receive a topic subscription request from a partner distributed computing system, such as partner computing system **110** depicted in FIG. **1**, and in operation **1110**, the server may register the partner topic subscription in a database, such as database **160** depicted in FIG. **1**. The registration request from the partner distributed computing

system may be submitted by way of a service gateway, such as the service gateway **130** depicted in FIG. **1**. In operation **1115**, the server may determine if one or more new events related to risk monitoring to report are stored in a database, such as database **160** depicted in FIG. **1**. If no events related to risk monitoring are stored in the database, then the server may wait for a predetermined period of time before repeating the determination. If events related to risk monitoring to report are stored in the database, then in operation **1120** the server may query the database for one or more partners, such as partners **110** depicted in FIG. **1**, subscribed to events related to risk monitoring. In operation **1125**, the server may generate, for each subscribed partner, an event report of all events related to risk monitoring for the subscribed partner, and in operation **1135**, the server may transmit notification related to risk monitoring to the subscribed partner. The event notification may be transmitted via service gateway, such as service gateway **130**. In operation **1130**, the server may register the event notification for each reported event related to risk monitoring in the database. In operation **1140**, the server may receive a request for notification details from the subscribed partner via the service gateway, and in operation **1145**, the server may transmit notification details to requesting subscribed partner. The notification details may be transmitted to the partner via the service gateway. In operation **1150**, the server may receive an acknowledgment of the notification from subscribed partner via the service gateway. In operation **1155**, the server may update the event notification status to reflect the acknowledgment by the partner. In operation **1160**, the server may receive a request to create a new risk monitoring event from an internal system, such as underwriting service **140** depicted in FIG. **1**. The event details may be transmitted to the partner via the service gateway. In operation **1165**, the server may register the new risk monitoring event in the database. An event identification number for registered event may be transmitted to partner via service gateway.

(38) For some event types, such as, for example, payment refund events published by an external service, a server, such as notification server **120** depicted in FIG. **1**, may wish track the reporting and acknowledgement of the reported events. FIG. **12** is a flow chart depicting an example process for secure transmission of payment refund notifications, according to one or more embodiments. In operation **1205**, a server, such as notification server **120** depicted in FIG. **1**, may receive a registration request from a partner distributed computing system, such as partner computing system **110** depicted in FIG. **1**, and in operation **1210**, the server may register the partner topic subscription in a database, such as database **160** depicted in FIG. **1**. The registration request from the partner distributed computing system may be submitted by way of a service gateway, such as the service gateway **130** depicted in FIG. **1**. In operation **1215**, the server may determine if one or more new events related to payment refund to report are stored in a database, such as database **160** depicted in FIG. **1**. If no events related to payment refund are stored in the database, then the server may wait for a predetermined period of time before repeating the determination. If events related to payment refund to report are stored in the database, then in operation **1220** the server may query the database for one or more partners, such as partners **110** depicted in FIG. **1**, subscribed to events related to payment refund. In operation **1225**, the server may generate, for each subscribed partner, an event report of all events related to payment refund for the subscribed partner, and in operation **1235**, the server may transmit each event report related to payment refund to the subscribed partner. The event report may be transmitted via service gateway, such as service gateway **130**. In operation **1230**, the server may register the event notification for each reported event related to payment refund in the database. For some event types, including events related to payment refund, the server may track the reporting and acknowledgement of the reported events. Accordingly, in operation **1240**, following the reporting of an event related to payment refund, the server may increment a number of attempted notifications for the payment refund event. If the reporting of an event related to payment refund is not acknowledged at operation **1245**, then the server may return to operation **1235** to re-transmit the unacknowledged event report related to payment refund to the subscribed partner. Otherwise, if the reporting of an event related to payment refund is acknowledged at

operation **1245**, then at operation **12500** the server may register the event report related to payment refund as completed in the database. In operation **1255**, the server may receive a request for event details from the subscribed partner via the service gateway, and in operation **1260**, the server may transmit event details to requesting subscribed partner. The event details may be transmitted to the partner via the service gateway. In operation **1265**, the server may receive new payment refund event from an external refund system, such as the contractual adjustment payment system (CAPS) **150** depicted in FIG. **1**. The event details may be transmitted to the partner via the service gateway. In operation **1270**, the server may register the new payment refund event in the database.

(39) FIGS. **13** and **14** depict a process flow diagram and a flow chart, respectively, of an example method for secure transmission of computer server event notifications, according to one or more embodiments. In operation **1410**, the server may prepare a notification workload including accessing encryption/decryption key information from a database, such as key store **1310** depicted in FIG. **13**. In operation **1420**, the server may obtain an encryption key for a subscribing partner, such as partner **110** depicted in FIG. **1**, from a database, such as database **1320** depicted in FIG. **13**. In operation **1430**, the server may sign the notification using a subscribing partner key. For example, the server may sign the notification using a JSON Web Token (JWT) such as JWT **1350** depicted in FIG. **13** and the subscribing partner encryption key obtained in operation **1420**.

However, other means for securely signing the notification may be employed. In operation **1440**, the server may append headers to the signed notification. For example, a header such as the HTTP header **1340** depicted in FIG. **13** may be appended to the signed notification. However, other types of headers may be appended to the notification. In operation **1450**, the server may deliver the notification to the subscribing partner via a service gateway, such as service gateway **130** depicted in FIG. **1**.

(40) FIGS. **15** and **16** depict a process flow diagram and a flow chart, respectively, of an example method for authentication in a system for secure transmission of computer server event notifications, according to one or more embodiments.

(41) In operation **1610**, the server may receive an updated shared key from the subscribing partner, and in operation **1620**, the server may store the received shared key in a database in an encrypted form, such as key-store **1310** depicted in FIG. **15**. A key stored in system key store is used encrypt shared key received. Operations **1610** and **1620** may be performed by a subscription manager within the server, such as subscription manager **122** depicted in FIG. **15**. In operation **1630**, the server may prepare a notification payload including accessing encryption/decryption key information from the database. In operation **1640**, the server may obtain MAC key for subscribing partner. In order to employ the obtained partner MAC key, the server may obtain a MAC key decryption key from the database in operation **1514**, and may decrypt the partner MAC key using the obtained MAC key decryption key in operation **1516**. In operation **1650**, the server may sign the notification using the subscribing partner MAC key. For example, the server may sign the notification using a JSON Web Token (JWT), such as JWT **1350** depicted in FIG. **15** and the subscribing partner MAC key obtained in operation **1610**. However, other means for securely signing the notification may be employed. In operation **1660**, the server may append headers to the signed notification. For example, a header such as the HTTP header **1340** depicted in FIG. **15** may be appended to the signed notification. However, other types of headers may be appended to the notification. In operation **1670**, the server may deliver the notification to the subscribing partner via a service gateway, such as service gateway **130** depicted in FIG. **1**. The preparation, signing, and transmission of the notification may be performed by a scheduler within the server, such as the scheduler **124** depicted in FIG. **15**.

(42) FIG. **17** depicts a flow chart of an example method for secure transmission of computer server event notifications, according to one or more embodiments. In operation **1710**, the server may get a header string containing a designation of a type of a token and a designation of a hashing algorithm used as well as a payload string that may contain claims information including epoch time. For



clarity, certain components, modules, or methods may be described solely in connection with a specific figure. In this disclosure, any identification of specific techniques, arrangements, etc. are either related to a specific example presented or are merely a general description of such a technique, arrangement, etc. Identifications of specific details or examples are not intended to be, and should not be, construed as mandatory or limiting unless specifically designated as such. Any failure to specifically describe a combination or sub-combination of components should not be understood as an indication that any combination or sub-combination is not possible. It will be appreciated that modifications to disclosed and described examples, arrangements, configurations, components, elements, apparatuses, devices, systems, methods, etc. can be made and may be desired for a specific application. Also, for any methods described, regardless of whether the method is described in conjunction with a flow diagram, it should be understood that unless otherwise specified or required by context, any explicit or implicit ordering of steps performed in the execution of a method does not imply that those steps must be performed in the order presented but instead may be performed in a different order or in parallel.

(49) Reference throughout the specification to “various embodiments,” “some embodiments,” “one embodiment,” “some example embodiments,” “one example embodiment,” or “an embodiment” means that a particular feature, structure, or characteristic described in connection with any embodiment is included in at least one embodiment. Thus, appearances of the phrases “in various embodiments,” “in some embodiments,” “in one embodiment,” “some example embodiments,” “one example embodiment, or “in an embodiment” in places throughout the specification are not necessarily all referring to the same embodiment. Furthermore, the particular features, structures or characteristics may be combined in any suitable manner in one or more embodiments.

(50) Throughout this disclosure, references to components or modules generally refer to items that logically can be grouped together to perform a function or group of related functions. Like reference numerals are generally intended to refer to the same or similar components. Components and modules can be implemented in software, hardware, or a combination of software and hardware. The term “software” is used expansively to include not only executable code, for example machine-executable or machine-interpretable instructions, but also data structures, data stores and computing instructions stored in any suitable electronic format, including firmware, and embedded software. The terms “information” and “data” are used expansively and includes a wide variety of electronic information, including executable code; content such as text, video data, and audio data, among others; and various codes or flags. The terms “information,” “data,” and “content” are sometimes used interchangeably when permitted by context. It should be noted that although for clarity and to aid in understanding some examples discussed herein might describe specific features or functions as part of a specific component or module, or as occurring at a specific layer of a computing device (for example, a hardware layer, operating system layer, or application layer), those features or functions may be implemented as part of a different component or module or operated at a different layer of a communication protocol stack. Those of ordinary skill in the art will recognize that the systems, apparatuses, devices, and methods described herein can be applied to, or easily modified for use with, other types of equipment, can use other arrangements of computing systems such as client-server distributed systems, and can use other protocols, or operate at other layers in communication protocol stacks, than are described.

(51) It is intended that the specification and examples be considered as exemplary only, with a true scope and spirit of the invention being indicated by the following claims.

## Claims

1. A method comprising: receiving, by a notification server, a subscription request from a partner computing system, wherein the subscription request identifies a license; registering, by the notification server, the subscription request in a database; querying, by the notification server, the

database for one or more events associated with the partner computing system; processing, by the notification server, the one or more events for inclusion in one or more notifications based on a time-dependent parameter, wherein a smaller or a larger number of events are included in the one or more notifications during periods of low activity; transmitting, by the notification server, the one or more notifications regarding the one or more events to the partner computing system, wherein an attempt count for transmitting the one or more notifications is incremented upon determining a confirmation has not been received from the partner computing system; receiving, by the notification server, a request for the one or more events from the partner computing system; transmitting, by the notification server, the one or more events to the partner computing system; and receiving, by the notification server, an acknowledgement from the partner computing system.

2. The method of claim 1, further comprising: publishing, by the notification server, an application programming interface (API) for secure transmission of the one or more notifications, wherein the API published by the notification server corresponds to an API published by a service gateway; and transmitting, by the notification server, the one or more notifications to the partner computing system via the API published by the notification server and the service gateway.

3. The method of claim 2, wherein receiving the subscription request comprises: verifying, by the notification server, a signature of the license; extracting, by the notification server, an API key from the license; and transmitting, by the notification server, the subscription request to the API published by the notification server based on the API key.

4. The method of claim 1, wherein querying the database for the one or more events comprises: determining, by the notification server, whether the one or more events are stored in the database, wherein the one or more events are new events, and wherein the notification server waits for a predetermined time for the one or more events before querying the database.

5. The method of claim 1, wherein transmitting the one or more notifications to the partner computing system is terminated upon determining the attempt count has exceeded an attempt count threshold.

6. The method of claim 5, wherein transmitting the one or more notifications comprises: determining, by the notification server, a notification failure when the attempt count exceeds the attempt count threshold; and accessing, by the notification server, an application programming interface (API) published by the partner computing system to report the notification failure.

7. The method of claim 1, wherein transmitting the one or more notifications comprises: pausing, by the notification server, transmission of the one or more notifications until the partner computing system is verified as available to receive the one or more notifications.

8. The method of claim 1, wherein transmitting the one or more events comprises: collecting, by the notification server, the one or more events associated with the partner computing system; determining, by the notification server, a collection of a pre-determined number of events; and grouping, by the notification server, the one or more events into a single event notification based on an event type, a preference setting of the partner computing system, or a setting of the notification server.

9. The method of claim 8, wherein grouping the one or more events into the single event notification comprises: setting, by the notification server, a pre-determined threshold for the one or more events that are grouped in the single event notification, wherein the pre-determined threshold is based on pre-determined time threshold.

10. The method of claim 1, further comprising: marking, by the notification server, transmission of the one or more events in the database as sent upon receiving the acknowledgement from the partner computing system.

11. A system comprising: one or more processors of a computing system; and at least one non-transitory computer readable medium storing instructions which, when executed by the one or more processors, cause the one or more processors to perform operations comprising: receiving, by a notification server, a subscription request from a partner computing system, wherein the

subscription request identifies a license; registering, by the notification server, the subscription request in a database; querying, by the notification server, the database for one or more events associated with the partner computing system; processing, by the notification server, the one or more events for inclusion in one or more notifications based on a time-dependent parameter, wherein a smaller or a larger number of events are included in the one or more notifications during periods of low activity; transmitting, by the notification server, the one or more notifications regarding the one or more events to the partner computing system, wherein an attempt count for transmitting the one or more notifications is incremented upon determining a confirmation has not been received from the partner computing system; receiving, by the notification server, a request for the one or more events from the partner computing system; transmitting, by the notification server, the one or more events to the partner computing system; and receiving, by the notification server, an acknowledgement from the partner computing system.

12. The system of claim 11, further comprising: publishing, by the notification server, an application programming interface (API) for secure transmission of the one or more notifications, wherein the API published by the notification server corresponds to an API published by a service gateway; and transmitting, by the notification server, the one or more notifications to the partner computing system via the API published by the notification server and the service gateway.

13. The system of claim 12, wherein receiving the subscription request comprises: verifying, by the notification server, a signature of the license; extracting, by the notification server, an API key from the license; and transmitting, by the notification server, the subscription request to the API published by the notification server based on the API key.

14. The system of claim 11, wherein querying the database for the one or more events comprises: determining, by the notification server, whether the one or more events are stored in the database, wherein the one or more events are new events, and wherein the notification server waits for a predetermined time for the one or more events before querying the database.

15. The system of claim 11, wherein transmitting the one or more notifications to the partner computing system is terminated upon determining the attempt count has exceeded an attempt count threshold.

16. The system of claim 15, wherein transmitting the one or more notifications comprises: determining, by the notification server, a notification failure when the attempt count exceeds the attempt count threshold; and accessing, by the notification server, an application programming interface (API) published by the partner computing system to report the notification failure.

17. The system of claim 11, wherein transmitting the one or more notifications comprises: pausing, by the notification server, transmission of the one or more notifications until the partner computing system is verified as available to receive the one or more notifications.

18. A non-transitory computer readable medium, the non-transitory computer readable medium storing instructions which, when executed by one or more processors of a computing system, cause the one or more processors to perform operations comprising: receiving, by a notification server, a subscription request from a partner computing system, wherein the subscription request identifies a license; registering, by the notification server, the subscription request in a database; querying, by the notification server, the database for one or more events associated with the partner computing system; processing, by the notification server, the one or more events for inclusion in one or more notifications based on a time-dependent parameter, wherein a smaller or a larger number of events are included in the one or more notifications during periods of low activity; transmitting, by the notification server, the one or more notifications regarding the one or more events to the partner computing system, wherein an attempt count for transmitting the one or more notifications is incremented upon determining a confirmation has not been received from the partner computing system; receiving, by the notification server, a request for the one or more events from the partner computing system; transmitting, by the notification server, the one or more events to the partner computing system; and receiving, by the notification server, an acknowledgement from the partner



computing system.

19. The non-transitory computer readable medium of claim 18, further comprising: publishing, by the notification server, an application programming interface (API) for secure transmission of the one or more notifications, wherein the API published by the notification server corresponds to an API published by a service gateway; and transmitting, by the notification server, the one or more notifications to the partner computing system via the API published by the notification server and the service gateway.

20. The non-transitory computer readable medium of claim 19, wherein receiving the subscription request comprises: verifying, by the notification server, a signature of the license; extracting, by the notification server, an API key from the license; and transmitting, by the notification server, the subscription request to the API published by the notification server based on the API key.

---