# SYSTEM FOR AND METHOD OF DISTRIBUTING DATA

## CROSS REFERENCE TO RELATED APPLICATION(S)

**[0001]** The present application claims priority from U.S. Provisional Patent Application Serial No. 63/565,977, filed March 15, 2024 and entitled SYSTEM FOR AND METHOD OF DISTIBUTING DATA, which is incorporated herein by reference in its entirety.

## RESERVATION OF COPYRIGHTS

**[0002]** Portions of the disclosure of this document contain material that is subject to copyright protection. The copyright owner has no objection to any reproduction of the document or disclosure as it appears in official records, but reserves all remaining rights under copyright.

## BACKGROUND

**[0003]** The present disclosure relates to electronic messaging, and more particularly to secure, high-volume electronic messaging suited for, but not limited to medical systems, equipment and treatment.

**[0004]** Providing patient care increasingly necessitates interactions among numerous professionals and caregivers (e.g., doctors, nurses, pharmacists, technicians, nurse practitioners, etc.) and any number of medical devices/systems needed for treatment of a given patient. Existing communications systems facilitate care processes, including secure access to electronic medical records ("EMR"), computerized provider order entry ("CPOE") and medical device/treatment management. Medical activities often include event-driven messaging, such as the raising of an alarm when an infusion pump experiences an occlusion. As medical professionals and patients increasingly rely on communications systems, the volume of electronic traffic increases substantially. This jeopardizes smooth implementation of treatment processes and the collection of data for evaluating and improving myriad aspects of treatment.

**[0005]** What is needed is a system for and method of distributing data that is secure, minimizes computing resources for originating and transmitting data, and allows for orderly processing of very large amounts of data received from diverse devices.

## SUMMARY OF THE INVENTION

**[0006]** The invention is a method of distributing data that is secure, minimizes computing resources for originating and transmitting data, and allows for orderly processing of very large amounts of data from diverse devices. An embodiment of a method configured according to

principles of the invention provides for identifying a message that corresponds to an event, encoding the message, defining an encoded message, and wrapping the encoded message in a payload. The payload is wrapped in an envelope for transmission.

**[0007]** An embodiment of a method configured according to principles of the invention provides for unwrapping a payload, revealing a message, and decoding the message. Prior to decoding, the message is routed to an appropriate dispatcher that routes the message to an appropriate event bus.

**[0008]** The invention provides improved elements and arrangements thereof, for the purposes described, which are inexpensive, dependable and effective in accomplishing intended purposes of the invention.

**[0009]** Other features and advantages of the invention will become apparent from the following description of the preferred embodiments, which refers to the accompanying drawings.

## BRIEF DESCRIPTION OF THE DRAWINGS

**[0010]** The invention is described in detail below with reference to the following figures, throughout which similar reference characters denote corresponding features consistently, wherein:

**[0011]** Fig. 1 is a schematic view of an embodiment of a system for distributing data configured according to principles of the invention;

**[0012]** Figs. 2 and 4 are graphical views of embodiments of methods of distributing data configured according to principles of the invention; and

**[0013]** Fig 3 is a graphical view of a message configured according to principles of the invention.

## DETAILED DESCRIPTION OF PREFERRED EMBODIMENTS

**[0014]** The examples shown in drawings are presented to demonstrate examples of the disclosure. The drawings are illustrative and non-limiting. In the drawings, for illustrative purposes, the size of some of the elements may be exaggerated and not drawn to a particular scale. Additionally, elements shown within the drawings that have the same numbers may be identical elements or may be similar elements, depending on the context.

**[0015]** Where the term "comprising" is used in the present description and claims, it does not exclude other elements or steps. Where an indefinite or definite article is used when referring to a singular noun, e.g., "a", "an", or "the", this includes a plural of that noun unless something otherwise is specifically stated. Hence, the term "comprising" should not be interpreted as being restricted to the items listed thereafter; it does not exclude other elements or steps, and

so the scope of the expression "a device comprising items A and B" should not be limited to devices consisting only of components A and B. Furthermore, to the extent that the terms “includes”, “has”, “possesses”, and the like are used in the present description and claims, such terms are intended to be inclusive in a manner similar to the term “comprising,” as “comprising” is interpreted when employed as a transitional word in a claim.

**[0016]** Furthermore, the terms "first", "second", "third", and the like, whether used in the description or in the claims, are provided to distinguish between similar elements and not necessarily to describe a sequential or chronological order. It is to be understood that the terms so used are interchangeable under appropriate circumstances (unless clearly disclosed otherwise) and that the aspects of the disclosure described herein are capable of operation in other sequences and/or arrangements than are described or illustrated herein.

**[0017]** In the following description, numerous specific details are set forth to provide a thorough understanding of various aspects and arrangements. It will be recognized, however, that the techniques described herein can be practiced without one or more of the specific details, or with other methods, components, materials, etc. In other instances, well known structures, materials, or operations may not be shown or described in detail to avoid obscuring certain aspects.

**[0018]** Reference throughout this specification to “an aspect,” “an arrangement,” “a configuration,” or “an example” indicates that a particular feature, structure, or characteristic is described. Thus, appearances of phrases such as “in one aspect,” “in one arrangement,” “in a configuration,” “in some examples,” or the like in various places throughout this specification do not necessarily each refer to the same aspect, feature, configuration, example, or arrangement. Furthermore, the particular features, structures, and/or characteristics described may be combined in any suitable manner.

**[0019]** To the extent used in the present disclosure and claims, the terms “component,” “system,” “platform,” “layer,” “selector,” “interface,” and the like are intended to refer to a computer-related entity or an entity related to an operational apparatus with one or more specific functionalities, wherein the entity may be either hardware, a combination of hardware and software, software, or software in execution. As an example, a component may be, but is not limited to being, a process running on a processor, a processor, an object, an executable, a thread of execution, a program, and/or a computer. By way of illustration and not limitation, both an application running on a server and the server itself can be a component. One or more components may reside within a process and/or thread of execution and a component may be localized on one computer and/or distributed between two or more computers. In addition, components may execute from various computer-readable media, device-readable storage devices, or machine-readable media having various data structures stored thereon. The

components may communicate via local and/or remote processes such as in accordance with a signal having one or more data packets (e.g., data from one component interacting with another component in a local system, a distributed system, and/or across a network such as the Internet with other systems via the signal). As another example, a component can be an apparatus with specific functionality provided by mechanical parts operated by electric or electronic circuitry, which may be operated by a software or firmware application executed by a processor, wherein the processor can be internal or external to the apparatus and executes at least a part of the software or firmware application. As yet another example, a component can be an apparatus that provides specific functionality through electronic components without mechanical parts; the electronic components can include a processor therein to execute software or firmware that confers at least in part the functionality of the electronic components.

**[0020]** To the extent used in the subject specification, terms such as "store," "storage," "data store," data storage," "database," and the like refer to memory components, entities embodied in a memory, or components comprising a memory. It will be appreciated that the memory components described herein can be either volatile memory or nonvolatile memory, or can include both volatile and nonvolatile memory.

**[0021]** In addition, the term "or" is intended to mean an inclusive "or" rather than an exclusive "or." That is, unless specified otherwise, or clear from context, "X employs A or B" is intended to mean any of the natural inclusive permutations. That is, if X employs A, X employs B, or X employs both A and B, then "X employs A or B" is satisfied under any of the foregoing instances. Moreover, articles "a" and "an" as used in the subject disclosure and claims should generally be construed to mean "one or more" unless specified otherwise or clear from context to be directed to a singular form.

**[0022]** The words "exemplary" and/or "demonstrative," to the extent used herein, mean serving as an example, instance, or illustration. For the avoidance of doubt, the subject matter disclosed herein is not limited by disclosed examples. In addition, any aspect or design described herein as "exemplary" and/or "demonstrative" is not necessarily to be construed as preferred or advantageous over other aspects or designs, nor is it meant to preclude equivalent exemplary structures and techniques. Furthermore, to the extent that the terms "includes," "has," "contains," and other similar words are used in either the detailed description or the claims, such terms are intended to be inclusive, in a manner similar to the term "comprising" as an open transition word, without precluding any additional or other elements.

**[0023]** As used herein, the term "infer" or "inference" refers generally to the process of reasoning about, or inferring states of, the system, environment, user, and/or intent from a set of observations as captured via events and/or data. Captured data and events can include user data, device data, environment data, data from sensors, application data, implicit data,

explicit data, etc. Inference can be employed to identify a specific context or action or can generate a probability distribution over states of interest based on a consideration of data and events, for example.

**[0024]** The disclosed subject matter can be implemented as a method, apparatus, or article of manufacture using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof to control a computer to implement the disclosed subject matter. The term "article of manufacture," to the extent used herein, is intended to encompass a computer program accessible from any computer-readable device, machine-readable device, computer-readable carrier, computer-readable media, or machine-readable media. For example, computer-readable media can include, but are not limited to, a magnetic storage device, e.g., hard disk; floppy disk; magnetic strip(s); an optical disk (e.g., compact disk (CD), digital video disc (DVD), Blu-ray Disc (BD)); a smart card; a flash memory device (e.g., card, stick, key drive); a virtual device that emulates a storage device; and/or any combination of the above computer-readable media.

**[0025]** Generally, program modules include routines, programs, components, data structures, etc., that perform particular tasks or implement particular abstract data types. The illustrated aspects of the subject disclosure may be practiced in distributed computing environments where certain tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules can be located in both local and remote memory storage devices.

**[0026]** Computing devices can include at least computer-readable storage media, machine-readable storage media, and/or communications media. Computer-readable storage media or machine-readable storage media can be any available storage media that can be accessed by the computer and includes both volatile and nonvolatile media, removable and non-removable media. By way of example, and not limitation, computer-readable storage media or machine-readable storage media can be implemented in connection with any method or technology for storage of information such as computer-readable or machine-readable instructions, program modules, structured data, or unstructured data.

**[0027]** Computer-readable storage media can include, but are not limited to, random access memory (RAM), read only memory (ROM), electrically erasable programmable read only memory (EEPROM), flash memory or other memory technology, compact disk read only memory (CD ROM), digital versatile disk (DVD), Blu-ray disc (BD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, solid state drives or other solid state storage devices, or other tangible and/or non-transitory media that can be used to store desired information. In this regard, the terms “tangible” or “non-transitory” herein as applied to storage, memory, or computer-readable media, are to be

understood to exclude only propagating transitory signals per se as modifiers, and do not exclude any standard storage, memory, or computer-readable media that are not only propagating transitory signals per se.

**[0028]** Computer-readable storage media can be accessed by one or more local or remote computing devices, e.g., via access requests, queries, or other data retrieval protocols, for a variety of operations with respect to the information stored by the medium.

**[0029]** A system bus, as may be used herein, can be any of several types of bus structure that can further interconnect to a memory bus (with or without a memory controller), a peripheral bus, and a local bus using any of a variety of commercially available bus architectures. A database, as may be used herein, can include basic input/output system (BIOS) that can be stored in a non-volatile memory such as ROM, EPROM, or EEPROM, with BIOS containing the basic routines that help to transfer information between elements within a computer, such as during startup. RAM can also include a high-speed RAM such as static RAM for caching data.

**[0030]** As used herein, a computer can operate in a networked environment using logical connections via wired and/or wireless communications to one or more remote computers. The remote computer(s) can be a workstation, server, router, personal computer, portable computer, microprocessor-based entertainment appliance, peer device, or other common network node. Logical connections depicted herein may include wired/wireless connectivity to a local area network (LAN) and/or larger networks, e.g., a wide area network (WAN). Such LAN and WAN networking environments are commonplace in offices and companies, and facilitate enterprise-wide computer networks, such as intranets, any of which can connect to a global communications network, e.g., the Internet.

**[0031]** When used in a LAN networking environment, a computer can be connected to the LAN through a wired and/or wireless communication network interface or adapter. The adapter can facilitate wired or wireless communication to the LAN, which can also include a wireless access point (AP) disposed thereon for communicating with the adapter in a wireless mode.

**[0032]** When used in a WAN networking environment, a computer can include a modem or can be connected to a communications server on the WAN via other means for establishing communications over the WAN, such as by way of the Internet. The modem, which can be internal or external, and a wired or wireless device, can be connected to a system bus via an input device interface. In a networked environment, program modules depicted herein relative to a computer or portions thereof can be stored in a remote memory/storage device.

**[0033]** When used in either a LAN or WAN networking environment, a computer can access cloud storage systems or other network-based storage systems in addition to, or in place of, external storage devices. Generally, a connection between a computer and a cloud storage

system can be established over a LAN or a WAN, e.g., via an adapter or a modem, respectively. Upon connecting a computer to an associated cloud storage system, an external storage interface can, with the aid of the adapter and/or modem, manage storage provided by the cloud storage system as it would other types of external storage. For instance, the external storage interface can be configured to provide access to cloud storage sources as if those sources were physically connected to the computer.

**[0034]** As employed in the subject specification, the term "processor" can refer to substantially any computing processing unit or device comprising, but not limited to comprising, single-core processors; single-core processors with software multithread execution capability; multi-core processors; multi-core processors with software multithread execution capability; multi-core processors with hardware multithread technology; vector processors; pipeline processors; parallel platforms; and parallel platforms with distributed shared memory. Additionally, a processor can refer to an integrated circuit, an application specific integrated circuit (ASIC), a digital signal processor (DSP), a field programmable gate array (FPGA), a programmable logic controller (PLC), a complex programmable logic device (CPLD), a state machine, a discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. Processors can exploit nano-scale architectures such as, but not limited to, molecular and quantum-dot based transistors, switches, and gates, in order to optimize space usage or enhance performance of user equipment. A processor may also be implemented as a combination of computing processing units. For example, a processor may be implemented as one or more processors together, tightly coupled, loosely coupled, or remotely located from each other. Multiple processing chips or multiple devices may share the performance of one or more functions described herein, and similarly, storage may be effected across a plurality of devices. A processor may be implemented to reside in a cloud-based network such as, e.g., the Internet.

**[0035]** The actions of a method or algorithm described in connection with the arrangements disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, hard disk, a removable disk, a CD-ROM, or any other known form of storage medium. A storage medium may be coupled to the processor such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in functional equipment such as, e.g., a computer, a robot, a user terminal, a mobile telephone or tablet, a car, or an IP camera. In the alternative, the processor and the storage medium may reside as discrete components in such functional equipment. Additionally, or

alternatively, at least one of the processor and/or the storage medium may reside in a cloud-based network such as, e.g., the Internet.

**[0036]** Configurations of the present teachings are directed to computer systems for accomplishing the methods discussed in the description herein, and to computer readable media containing programs for accomplishing these methods. The raw data and results can be stored for future retrieval and processing, printed, displayed, transferred to another computer, and/or transferred elsewhere. Communications links can be wired or wireless, for example, using cellular communication systems, military communications systems, and satellite communications systems. Parts of the system can operate on a computer having a variable number of CPUs. Other alternative computer platforms can be used.

**[0037]** The present configuration is also directed to software/firmware/hardware for accomplishing the methods discussed herein, and computer readable media storing software for accomplishing these methods. The various modules described herein can be accomplished on the same CPU, or can be accomplished on different CPUs. In compliance with the statute, the present configuration has been described in language more or less specific as to structural and methodical features. It is to be understood, however, that the present configuration is not limited to the specific features shown and described, since the means herein disclosed comprise preferred forms of putting the present configuration into effect.

**[0038]** Methods can be, in whole or in part, implemented electronically. Signals representing actions taken by elements of the system and other disclosed configurations can travel over at least one live communications network. Control and data information can be electronically executed and stored on at least one computer-readable medium. The system can be implemented to execute on at least one computer node in at least one live communications network. Common forms of at least one computer-readable medium can include, for example, but not be limited to, a floppy disk, a flexible disk, a hard disk, magnetic tape, or any other magnetic medium, a compact disk read only memory or any other optical medium, punched cards, paper tape, or any other physical medium with patterns of holes, a random access memory, a programmable read only memory, and erasable programmable read only memory (EPROM), a Flash EPROM, or any other memory chip or cartridge, or any other medium from which a computer can read. Further, the at least one computer readable medium can contain graphs in any form, subject to appropriate licenses where necessary, including, but not limited to, Graphic Interchange Format (GIF), Joint Photographic Experts Group (JPEG), Portable Network Graphics (PNG), Scalable Vector Graphics (SVG), and Tagged Image File Format (TIFF).

**[0039]** Various arrangements are described herein. For simplicity of explanation, the methods or algorithms are depicted and described as a series of steps or actions. It is to be understood

and appreciated that the various arrangements are not limited by the actions illustrated and/or by the order of actions. For example, actions can occur in various orders and/or concurrently, and with other actions not presented or described herein. Furthermore, not all illustrated actions may be required to implement the methods. In addition, the methods could alternatively be represented as a series of interrelated states via a state diagram or events. Additionally, the methods described hereafter are capable of being stored on an article of manufacture, as defined herein, to facilitate transporting and transferring such methodologies to computers.

**[0040]** Referring to Fig. 1, the invention is a system for and method of distributing data that is secure and minimizes computing resources. Fig. 1 depicts an exemplary implementation of or system 300 for treating a patient (not shown) with a medical device 301. Medical device 301 may be selected from, for example: a patient-condition device, such as a pulse oximeter device, a heart monitor device, a blood pressure device, and a temperature device; a patient treatment device, such as a drip counter, an infusion pump, a pill dispenser, or any internet-enabled device or device interface that is connected with a message-generation activity.

**[0041]** Device 301, preferably, is configured to communicate with a gateway 305 via an internet connection 302, for example, a hard-wired connection, such as Ethernet. Preferably device 301 communicates wirelessly, such as with Wi-Fi, Bluetooth and the like. Cellular communications between device 301 and gateway 305 also are within the scope of the invention.

**[0042]** While operating, device 301 experiences events, such as, but not limited to those listed in column 2 of Table 1 below, based on signals from components and sensors in device 301 commensurate with normal and abnormal operations. Each event is associated with a unique ID, such as those listed in column 1 of Table 1 below.

| ID | Event Name |
|---|---|
| 2 | Infusion Events (Alarms, Alerts, Notifications) |
| 2.1 | High priority technical Alarm signaled |
| 2.2 | High priority Operational Alarm signaled |
| 2.3 | Occlusion Alarm signaled |
| 2.4 | Side clamp not installed when loading administartion set |
| 2.5 | Peristaltic pump not sealed |
| 2.6 | Administration set removed during infusion |
| 2.7 | Under infusion Alarm |
| 2.8 | Air limit reached |
| 2.9 | Air single bubble exceeds allowable |
| 2.1 | Alarm condition cleared by operator |
| 2.11 | Internal Software Error |
| 2.12 | Medium priority Alert signaled |
| 2.13 | Medium priority Alert escalated signaled |
| 2.14 | Operator inactivity during programming |
| 2.15 | Low priority Alert signaled |
| 2.16 | Infusion near end Alert |
| 2.17 | Callback alert signaled |
| 2.18 | Notification signaled |
| 2.19 | Alarm silenced |

Table 1

**[0043]** Referring also to Fig. 2, a method configured according to principles of the invention includes device 301 performing a step 405 of identifying an event. For example, responsive to detecting a high pressure in device 301, a pressure sensor may transmit an alarm signal to a processor (not shown) in device 301 that correlates with or is identifiable as an occlusion, for example in a cannula dedicated for supplying an infusate to a patient. Consistent with Table 1 above, the occlusion event would be identified by the event ID 2.3.

**[0044]** Once identified, preferably, method 400 includes device 301 performing a step 410 of encoding the event associated ID as a message. Step 410 is included and configured as required to respect formatting and/or protocols of existing decoding functions, as described below with respect to step 545. Commensurate with an envisioned decoder configured to receive and process hexadecimal inputs, one embodiment of step 410 includes converting the event ID into a hexadecimal string. For example, for an "Occlusion Alarm signaled" event having an ID of 2.3, step 410 may include converting the ID from $2.3_{10}$ to $2.4CCCCCCCCC_{16}$.

Alternatively, the invention could take advantage of a decoder configured to receive and process base ten inputs. In which case, step 410 would not be necessary.

**[0045]** While the message may be inherently small and/or compressed as part of step 410, method 400 does not exclude an additional step 415 of compressing the message to achieve as small a message as possible.

**[0046]** Referring also to Fig. 3, once the message is encoded, compressed or otherwise readied for packaging, method 400 progresses to initializing and building a package or envelope 600, preferably, consistent with an open standard file format and data interchange format that uses human-readable text to store and transmit data objects consisting of attribute–value pairs and arrays, such as JSON, for communicating the event. Preferably, envelope 600 is configured so as to enable its routing through external or uncontrolled communications networks without having to unpack or understand the content of the message.

**[0047]** Envelope 600 includes a payload 605 and a header 610. Payload 605 includes a number of prescribed fields or keys that are associated with values. For example, one key may be tenantId 620, for identifying an associated dedicated instance reserved for a specific customer or intended internal message bus 375, as described below.

**[0048]** Preferably, payload 605 includes a key, such as route 655, for describing or identifying the path that envelope 600 took before arriving at a final decoding location. For example, route 655 may describe package 600 as having coursed from device to router to gateway to event stream to internal message bus to dispatcher to event bus to decoder to storage. Route 655 may or may not include the unique identifier associated with each stop along the way.

**[0049]** Preferably, payload 605 includes a key, such as Sourcetype 660, for identifying from where an event originated, such as medical device 301.

Preferably, payload 605 includes a key, such as signModeKeyId 630, for identifying which asymmetrical key to use for validating or verifying the authenticity of the package.

**[0050]** Preferably, payload 605 includes a key, such as encModeKeyId 635, which identifies the symmetrical key used for decrypting payload 605.

**[0051]** Preferably, payload 605 includes a timestamp key 640 corresponding to the date and time of the occurrence of the event.

**[0052]** Preferably, payload 605 includes a key, such as reporterId 665, for identifying the device or function that last transmitted the message.

**[0053]** Preferably, payload 605 includes keys, such as eventId key 670 and eventType 675, for defining a status of the device, such as device is online (ev_device_online).

**[0054]** Preferably, payload 605 includes a key, such as sourceId 680, for identifying the originator of the message, such as medical device 301.

**[0055]** Preferably, payload 605 includes a key, such as userId 685 for identifying a patient receiving treatment from or associated with medical device 301.

**[0056]** Preferably, payload 605 includes a key, such as eventData 690, for the encoded/compressed/encrypted message described above with respect to table 1.

**[0057]** Payload 605 may include a key, such as metadata 695, for any additional data about the message that is not otherwise included in payload 605, such as a client application version, in an unstructured manner.

**[0058]** Preferably, payload 605 includes keys, such as internalEventId 705 and internalEventType 700, which essentially are extended or enhanced versions respectively of eventId key 675 and eventType 680, for additional info or more trustworthy internal tracking.

**[0059]** Preferably, internalEventId 705 and internalEventType 700 are not added to payload 605 by device, but rather by a dispatcher 315, as described below.

**[0060]** Preferably, header 610 includes a key, such as envelopeVersion 615, that identifies the version of the architecture, that is, the specific keys and arrangements thereof in the for the package or envelope 600.

**[0061]** Preferably, header 610 also includes key 620 as described above.

**[0062]** Preferably, header 610 includes a key, such as signature 625, for containing an asymmetric private key for authenticating the package 600.

**[0063]** Preferably, header 610 also includes keys 630, 635 and 640 as described above.

**[0064]** Preferably, header 610 includes a key, such as requestId 645 for tracking and associating every action done on the event.

**[0065]** Continuing to refer to Figs. 2 and 3, with the architecture of envelope 600 detailed, method 400 continues and includes device 301 performing a step 420 wrapping the message, or associating the message with keys, consistent with envelopeVersion key 615.

**[0066]** Method 400 then continues with device 301 performing a step 425 of encrypting the payload, preferably with the symmetric key, such as AES/CBC/PKCS5Padding, of device 301. Consistent with principles of the invention, the encryption protocol should convert the message into a cypher having a minimized size.

**[0067]** Once payload 605 is encrypted, method 400 includes device 301 performing a step 430 of digitally signing the payload, preferably with an asymmetric private key, such as RSA 2048 / SHA256withRSA with PKCS1v15 padding. Step 430 aids in validating payload 610 before it is passed through gateway 305. The value of step 430 defines key 625.

**[0068]** Once payload 605 is signed, method 400 includes device 301 performing a step 435 of encoding payload 605, preferably in base 64. Like step 410, step 435 is included to respect formatting and/or protocols that may be incompatible with JSON data objects.

**[0069]** Once payload 605 is encoded, method 400 includes device 301 performing a step 440 or wrapping the payload in an envelope or associating identifying information, such as header 610, with payload 605. Step 440 aids in routing envelope 600 to the appropriate gateway 305.

**[0070]** Finally, method 400 includes device 301 performing a step 445 of publishing the event or envelope 600 to a messaging broker that services gateway 305 for receiving, routing and queuing messages appropriately.

**[0071]** Preferably, step 445 employs a lightweight, publish-subscribe, machine-to-machine network protocol for message queue/message queuing service, such as MQTT. A lightweight protocol conserves processing resources otherwise consumed from processing a great number of messages from a great number of devices, which may have resource constraints or limited network bandwidth typical to IoT applications.

**[0072]** Preferably, step 445 is not encumbered with transmission confirmations or receipts. This greatly reduces the amount of processing resources consumed as well as the potential for operational hysteresis. That is, particularly when many devices 301 are operating and generating many messages, relieving the system from generating and processing message confirmations greatly reduces the potential for losing or corrupting contemporaneous messages.

**[0073]** In sum, method 400 is configured to produce very small (size-wise) packages or envelopes 600 that contain or represent a lot of detailed data. Step 410 encoding (and step 415 compressing, if performed) renders payload 605 very small so that step 425 encrypting performed on that small payload 605 further minimizes processing demands. Since payload 605 is not expanded until needed, package 600 can travel far in its miniaturized form without losing any fidelity.

**[0074]** Referring again to Fig. 1 and to Fig. 4, having departed device 301 at the end of method 400, package 600 is processed according to a method 500, preferably executed in the cloud 365. Method 500 includes a step 505 of gateway 305 receiving an envelope or package. The MQTT broker that services gateway 305 then performs a step 510 of placing envelope 600 into an event stream 310.

**[0075]** Method 500 includes event stream 310 performing a step 515 of decoding payload 605 commensurate with step 435 above.

**[0076]** Once payload 605 is decoded, method 500 includes event stream 310 performing a step 515 of verifying the digital signature thereof commensurate with step 430 above.

**[0077]** Once the digital signature of payload 605 is verified, method 500 includes event stream 310 performing a step 520 of decrypting payload 605 commensurate with step 425 above.

**[0078]** Once payload 605 is decrypted, preferably, method 500 includes event stream 310 performing a step 525 of routing payload 605 to an internal message bus 375 and queuing the message.

**[0079]** Preferably, step 525 employs a simple publish-subscribe system, preferably, but not limited to Amazon® AWS SNS. Through the SNS, messages are queued asynchronously to a "topic" to which a subscriber, like dispatcher 315, may subscribe. A "topic" may be created for each device 301. Queuing ensures that messages are retained until processing by dispatcher 315 is desired or possible. Preferably, step 525 also employs a simple queuing service, such as but not limited to Amazon® AWS SQS. SQS enables receiving and storing messages at any volume without losing messages or requiring other services to be available. SQS prevents the invention from becoming bogged down or halted from a superabundance of data and permit routing and processing of data according to the processing capabilities of system components.

**[0080]** When feasible or desired, method 500 includes dispatcher 315 performing a step 530 of examining payload 605, in plaintext, to determine where to route the message, that is, to which event bus 320 the message pertains, each of diverse device types or groups having a designated event bus 320. For example, devices that provide for drip infusion may define one group and devices that provide for monitoring parameters, like blood pressure, may define another group, each group having its own event bus 320. As mentioned above, dispatcher 315 may add to or supplement payload 605 with keys 700 and 705 based on this information.

**[0081]** Based on step 530, method 500 includes dispatcher 315 performing a step 535 of publishing the envelope 600 to an appropriate event bus 320, e.g. corresponding to an SNS Topic associated with an appropriate tag (internalEventType 700). Device-specific buses promote rapid processing of payloads 605 that are specific to any of a wide variety of medical devices that perform different functions associated with different parameter sets.

**[0082]** When envelope 600 is received, method 500 includes event bus 320 performing a step 540 of queuing the envelope in a decoder 325.

**[0083]** When decoder 325 is available, method 500 includes decoder 325 performing a step 545 of decoding the message of envelope 600. An embodiment of the invention embraces employing decoders that are appropriate for each message version. However, package 600 may be configured so that step 545 decoding can service any payload 610 regardless of to what device 301 package 600 pertains.

**[0084]** Once the message is decoded, method 500 includes decoder 325 performing a step 550 of storing in an event database 370. Step 550 is facilitated by returning envelope 600 to event bus 320, which performs a step 555 of routing envelope 600 to a storage function 335.

**[0085]** The foregoing pertains to basic message generation, transmission, reception, recognition and storage. The individual elements and combinations of (1) creating small-sized, data rich, encrypted message packages (2) transmitted via a lightweight network protocol enables (1) transmission in a miniaturized form of significant amounts of data (2) with high fidelity. Since the package is not expanded or decoded until needed, it can travel far in miniaturized form without losing any fidelity.

**[0086]** Once the payload is at its destination, the decoder turns it into human-readable text that can be used by any service. A single decoder service relieves consumers from having to know how to decode packages on their own and reduces redundant package processing; the package is processed once and the resulting data can be used by many consumers. The minimized package size enabled by the invention also reduces the amount of resources that a device, like a medical device, uses to store and transmit the event, thus saving storage space on the device and ensuring that more events can be stored in the limited space available until they are all transmitted to a server.

**[0087]** The minimized packaging, light-weight queuing and use of micro functions for managing received messages enables high availability and horizontal scalability, that is, the ability to handle messaging from high numbers of medical devices employed simultaneously across the globe.

**[0088]** The invention extends beyond basic information communication and storage to analytics. To this end, preferably, method 500 includes storage function 335 performing a step 560 of storing the decoded message in an event database 370, such as Opensearch or similar NoSQL database.

**[0089]** The invention includes an event query API service 385 for querying database 370 for messages based on a variety of criteria, such as time, device of origin, type of message, etc.

**[0090]** The invention includes user interface (UI) services 350, preferably accessible via a browser, utilizing API service 385.

**[0091]** The invention includes a data analysis and research service 340 for servicing the API queries.

**[0092]** The invention includes a medical device manager 380 for sending message responses back to device 301 as needed. For example, manager 380 may send an event to device 301 that: verifies the authenticity of an attached component, such as an insulin pump; informs device 301 of the availability of a firmware software update; or instructs device 301 to modify its settings.

**[0093]** The invention is not limited to the particular embodiments described and depicted herein, rather only to the following claims.

CLAIMS

WE CLAIM:

1. Method of distributing data comprising:
identifying a message that corresponds to an event;
encoding the message and defining an encoded message; and
wrapping the encoded message in a payload.

2. Method of claim 1 wherein the message is based on a status of a medical device.

3. Method of claim 1 wherein said encoding comprises hexadecimal conversion.

4. Method of claim 1 further comprising encrypting the message, the encoded message and/or the payload.

5. Method of claim 1 further comprising compressing the message, the encoded message and/or the payload.

6. Method of claim 1 further comprising signing the payload.

7. Method of claim 1 further comprising wrapping the payload in an envelope.

8. Method of claim 7 further comprising transmitting the envelope comprising a lightweight, publish-subscribe, machine-to-machine network protocol.

9. Method of distributing data comprising:
unwrapping a payload and revealing a message; and
decoding the message and defining a decoded message.

10. Method of claim 9 further comprising placing an envelope comprising the payload in an event stream.

11. Method of claim 9, wherein an envelope wraps the payload, further comprising routing the envelope to an event bus.

12. Method of claim 11 wherein said routing is based on the payload.

13. Method of claim 9, further comprising supplementing the payload.

14. Method of claim 9 wherein the message is based on a status of a medical device.

15. Method of claim 9 wherein said decoding comprises hexadecimal conversion.

16. Method of claim 9 further comprising decompressing the message, the decoded message and/or the payload.

17. Method of claim 9 further comprising decrypting the message, the decoded message and/or the payload.

18. Method of claim 9 further comprising verifying a signature of the payload.

19. Method of claim 9 further comprising:
storing the decoded message in a database; and
providing an application protocol interface configured for accessing the database.

20. Method of claim 9 further comprising transmitting a second message based on the message.