

System and method for processing data

Technical Field

[0001] This disclosure relates to the field of Internet of Things, cloud computing, edge processing and telecommunications. In particular, this disclosure relates to systems and methods for selectively processing data generated by sub-devices, by selectively providing the data to one or more multicast network channels that are accessible by containerised applications.

Background

[0002] Sensors can be used to monitor a variety of parameters that range from, for example, temperature sensors for measuring a temperature of an environment to optical sensors that record images or video of a region. A large number of sensors can be included in sensing systems and optimising how data provided by the sensors is processed is technically difficult.

[0003] For example, modern surveillance systems (e.g. security systems used by private security companies or security systems used by governments for local or national security) can include a large number of surveillance cameras and other sensors. The use of these surveillance systems also extends beyond local or national security, into domains such as smart city initiatives where public infrastructure may be controlled in response to data provided by the sensors. For example, street lights may be controlled in response to changes in environmental light conditions.

[0004] Sensing systems can therefore include both a large number of sensors and a large variety in the type of sensors used. For example, sensing systems can include sensors associated with the operation of machines or facilities, surveillance cameras (e.g. Closed Circuit Television (CCTV) cameras, vibration sensors, humidity sensors, gas detectors etc. Selectively processing data from such a number of sensors and such a variety of sensors provides significant safety, security, efficiency and scalability problems.

[0005] Any discussion of documents, acts, materials, devices, articles or the like which has been included in the present specification is not to be taken as an admission that any or all of these matters form part of the prior art base or were common general knowledge in the field

relevant to the present disclosure as it existed before the priority date of each claim of this application.

[0006] Throughout this specification the word "comprise", or variations such as "comprises" or "comprising", will be understood to imply the inclusion of a stated element, integer or step, or group of elements, integers or steps, but not the exclusion of any other element, integer or step, or group of elements, integers or steps.

Summary

[0007] In some embodiments, there is provided a computer-implemented method for selectively processing data. The computer implemented method may comprise: receiving, by a service device, a request for access from a sub-device, the request for access comprising sub-device information associated with the sub-device; receiving, by the service device, authorisation to enable access, the authorisation being based at least in part on the sub-device information; receiving, by the service device, sub-device data from the sub-device; broadcasting, by the service device, the sub-device data to a corresponding multicast network channel that is accessible by a plurality of containerised applications of the service device; receiving, by a containerised application of the plurality of containerised applications, the sub-device data from the multicast network channel; processing, by the containerised application, the sub-device data to generate processed data; and transmitting, by the service device and to a data management device, the processed data.

[0008] In some embodiments, the computer implemented method further comprises: receiving, by a user interface of the data management device, authorisation credentials; generating, by the data management device, an authorisation token based at least in part on the authorisation credentials; and transmitting the authorisation token to the service device.

[0009] In some embodiments, the computer-implemented method further comprises transmitting, by the service device, the request for access to the data management device.

[0010] In some embodiments, receiving authorisation to enable access comprises receiving the authorisation token.

[0011] In some embodiments, the computer-implemented method further comprises storing, by the data management device, the processed data.

[0012] In some embodiments, the sub-device information comprises one or more of a sub-device serial number, a sub-device Media Access Control (MAC) address, a sub-device model number, a sub-device identifier and a random key.

[0013] In some embodiments, the random key is stored in a Trusted Platform Module of the service device memory.

[0014] In some embodiments, the sub-device comprises a sensor and the sub-device data comprises sensor data.

[0015] In some embodiments, the multicast network channel is a multicast Internet Protocol (IP) address or an application-layer multicast channel.

[0016] In some embodiments, processing the sub-device data comprises determining one or more sub-device data metrics based at least in part on the sub-device data.

[0017] In some embodiments, the computer-implemented method further comprises: generating a private key; generating a public key that is associated with the private key; and associating, in a sub-device record, the public key with the sub-device.

[0018] In some embodiments, the private key and the public key are generated by the service device.

[0019] In some embodiments, the computer-implemented method further comprises one or more of: providing the private key to the data management device; providing the public key to the data management device; providing the private key to the at least one containerised application; providing the public key to the at least one containerised application; and/or providing the public key to the sub-device.

[0020] In some embodiments, the private key and the public key are generated by the data management device.

[0021] In some embodiments, the computer-implemented method further comprises one or more of: providing the private key to the service device; providing the public key to the service device; providing the private key to the at least one containerised application; providing the public key to the at least one containerised application; and/or providing the public key to the sub-device.

[0022] In some embodiments, the computer-implemented method of further comprises encrypting, by the service device and using the public key, the received sub-device data prior to broadcasting the sub-device data to the multicast network channel, thereby generating encrypted sub-device data.

[0023] In some embodiments, broadcasting the sub-device data to the multicast network channel comprises broadcasting the encrypted sub-device data to the multicast network channel.

[0024] In some embodiments, receiving the sub-device data from the multicast network channel comprises receiving the encrypted sub-device data.

[0025] In some embodiments, the computer-implemented method further comprises decrypting, by the containerised application and using the private key, the encrypted sub-device data received from the multicast network channel, prior to processing the sub-device data.

[0026] In some embodiments, the computer-implemented method further comprises encrypting, by the containerised application and using the public key, the processed data to generate encrypted processed data.

[0027] In some embodiments, transmitting the processed data comprises transmitting the encrypted processed data.

[0028] In some embodiments, storing the processed data comprises storing the encrypted processed data.

[0029] In some embodiments, the computer-implemented method further comprises transmitting, by the service device and to the data management device, a request to initialise a

new containerised application; retrieving, by the service device, the new containerised application; initialising, by the service device, the new containerised application; and transmitting, by the service device, an installation confirmation indicating a successful initialisation of the containerised application to the data management device.

[0030] In some embodiments, the containerised application of the plurality of containerised applications is the new containerised application.

[0031] In some embodiments, the computer-implemented method further comprises: receiving, by the data management device, the request to initialise the new containerised application; and transmitting, by the data management device, a license key to the service device, the license key being associated with the containerised application.

[0032] In some embodiments, there is provided a data management system for selectively processing data. The data management system may comprise a service device. The service device may comprise at least one service device processor; and service device memory. The service device memory may store: a plurality of containerised applications executable by the at least one service device processor; and program instructions accessible by the at least one service device processor. The program instructions may be configured to cause the at least one service device processor to: receive a request for access from a sub-device, the request for access comprising sub-device information associated with the sub-device; receive authorisation to enable access, the authorisation being based at least in part on the sub-device information; receive sub-device data from the sub-device; broadcast the sub-device data to a corresponding multicast network channel that is accessible by the plurality of containerised applications and execute a containerised application of the plurality of containerised applications. The containerised application may be configured to: receive the sub-device data from the multicast network channel; and process the sub-device data to generate processed data. The program instructions may be configured to cause the at least one service device processor to transmit the processed data to a data management device.

[0033] In some embodiments, the data management system further comprises the data management device. The data management device may comprise: at least one data management device processor; and data management device memory. The data management device memory may store data management program instructions accessible by the at least

one data management device processor. The data management program instructions may be configured to cause the at least one data management device processor to receive the processed data.

[0034] In some embodiments, the data management program instructions are further configured to cause the at least one data management device processor to: receive authorisation credentials; generate an authorisation token based at least in part on the authorisation credentials; and transmit the authorisation token to the service device, thereby providing authorisation to enable access.

[0035] In some embodiments, the service device program instructions are further configured to cause the at least one service device processor to transmit the request for access to the data management device.

[0036] In some embodiments, the service device program instructions are further configured to cause the at least one service device processor to receive the authorisation token, thereby receiving authorisation to enable access.

[0037] In some embodiments, the data management program instructions are further configured to cause the at least one data management device processor to store the processed data.

[0038] In some embodiments, the sub-device information comprises one or more of a sub-device serial number, a sub-device Media Access Control (MAC) address, a sub-device model number, a sub-device identifier and a random key.

[0039] In some embodiments, the sub-device comprises a sensor and the sub-device data comprises sensor data.

[0040] In some embodiments, the multicast network channel is a multicast Internet Protocol (IP) address or an application-layer multicast channel.

[0041] In some embodiments, processing the sub-device data comprises determining one or more sub-device data metrics based at least in part on the sub-device data.

[0042] In some embodiments, the service device program instructions are further configured to cause the at least one service device processor to: generate a private key based at least in part on the sub-device information; generate a public key that is associated with the private key; and associate, in a sub-device record, the public key with the sub-device.

[0043] In some embodiments, the service device program instructions are further configured to cause the at least one service device processor to: provide the private key to the data management device; provide the public key to the data management device; provide the private key to the at least one containerised application; provide the public key to the at least one containerised application; and/or provide the public key to the sub-device.

[0044] In some embodiments, the data management program instructions are further configured to cause the at least one data management device processor to: generate a private key based at least in part on the sub-device information; generate a public key that is associated with the private key; and associate, in a sub-device record, the public key with the sub-device.

[0045] In some embodiments, the data management program instructions are further configured to cause the at least one data management device processor to: provide the private key to the service device; provide the public key to the service device; provide the private key to the at least one containerised application; provide the public key to the at least one containerised application; and/or provide the public key to the sub-device.

[0046] In some embodiments, the service device program instructions are further configured to cause the at least one service device processor to encrypt the received sub-device data using the public key, prior to broadcasting the sub-device data to the multicast network channel, thereby generating encrypted sub-device data.

[0047] In some embodiments, broadcasting the sub-device data to the multicast network channel comprises broadcasting the encrypted sub-device data to the multicast network channel.

[0048] In some embodiments, receiving the sub-device data from the multicast network channel comprises receiving the encrypted sub-device data.

[0049] In some embodiments, the program instructions are further configured to cause the at least one service device processor to execute the containerised application of the plurality of containerised applications.

[0050] In some embodiments, the containerised application is configured to decrypt the encrypted sub-device data received from the multicast network channel using the private key, prior to processing the sub-device data.

[0051] In some embodiments, the program instructions are further configured to cause the at least one service device processor to execute the containerised application of the plurality of containerised applications.

[0052] In some embodiments, the containerised application is configured to encrypt the processed data to generate encrypted processed data.

[0053] In some embodiments, transmitting the processed data comprises transmitting the encrypted processed data.

[0054] In some embodiments, storing the processed data comprises storing the encrypted processed data.

[0055] In some embodiments, the program instructions are further configured to cause the at least one service device processor to: transmit a request to initialise a new containerised application to the data management device; retrieve the new containerised application; initialise the new containerised application; and transmit an installation confirmation indicating successful initialisation of the containerised application to the data management device.

[0056] In some embodiments, the containerised application of the plurality of containerised applications is the new containerised application.

[0057] In some embodiments, the data management program instructions are further configured to cause the at least one data management device processor to: receive the request to initialise the new containerised application; and transmit a licence key to the service device, the license key being associated with the containerised application.

[0058] In some embodiments, there is provided a computer-implemented method for securely initialising a containerised application. The computer-implemented method may comprise: receiving, by a data management device, a request from a service device to initialise the containerised application; transmitting, by the data management device, a license key to the service device, the license key being associated with the containerised application; retrieving, by the service device, the containerised application; initialising, by the service device, the containerised application using the license key; transmitting, by the service device, an installation confirmation indicating a successful initialisation of the containerised application to the data management device.

[0059] In some embodiments, the computer-implemented method further comprises executing, by the service device, the containerised application.

[0060] In some embodiments, the computer-implemented method further comprises transmitting, by the service device, the request to initialise the containerised application to the data management device in response to the service device receiving a request for access from a sub-device that is associated with the containerised application.

[0061] In some embodiments, the computer-implemented method further comprises: transmitting, by the service device, the license key to a license validation service; and receiving, by the service device, a license validation indicating that the license key is valid.

[0062] In some embodiments, receiving the containerised application comprises: downloading the containerised application from a software repository, over a network; and installing the containerised application.

[0063] In some embodiments, there is provided a data management system for securely initialising a containerised application. The data management system may comprise a service device. The service device may comprise: at least one service device processor; and service device memory. The service device memory may store program instructions accessible by the at least one service device processor. The program instructions may be configured to cause the at least one service device processor to: transmit a request to initialise a containerised application to a data management device; retrieve the containerised application; initialise the containerised application; and transmit an installation confirmation indicating a successful

initialisation of the containerised application to the data management device. The data management system may comprise the data management device. The data management device may comprise: at least one data management device processor; and data management device memory. The data management device memory may store data management program instructions accessible by the at least one data management device processor. The data management program instructions may be configured to cause the at least one data management device processor to: receive the request to initialise the containerised application; and transmit a licence key to the service device, the license key being associated with the containerised application.

[0064] In some embodiments, the program instructions are further configured to cause the at least one service device processor to execute the containerised application.

[0065] In some embodiments, the program instructions are further configured to cause the at least one service device processor to transmit the request to initialise the containerised application to the data management device in response to the service device receiving a request for access from a sub-device that is associated with the containerised application.

[0066] In some embodiments, the program instructions are further configured to cause the at least one service device processor to: transmit the license key to a license validation service; and receive a license validation indicating that the license key is valid.

[0067] In some embodiments, receiving the containerised application comprises: downloading the containerised application from a software repository, over a network; and installing the containerised application.

[0068] Usage of a multicast network (e.g. an IP multicast network) and a plurality (which may also be referred to as a multi-number) of multicast network channels (which may also be referred to as data channels) for the distribution of sensor data among a plurality of applications for the concurrent data processing of the sensor data, while an access control mechanism allows only specific applications to receive the designated sensor data is a deployment consideration of the present disclosure that improves content privacy among dynamic multicast group memberships. Advantageously, the disclosure manages access to

sub-device data provided to multicast channels by selectively providing an access key to one or more authorised users and/or containerised applications.

[0069] In some embodiments, there is provided a method for providing access to multicast communications based on assignment of a license key, which advantageously is configured by a central management system. There is also provided a method of managing of access to multicast communications by enabling access interaction of incoming data with selected applications by remote access management.

[0070] In some embodiments, there is provided a system for managing access to multicast communications. The system comprises:

- a container of applications residing in a service device, wherein each application in the container is selectively encrypted and assigned a license key;

- a plurality of sub-devices connecting to a service device; and

- a supervisor residing in the service device configured to authenticate access to the multicast communication, wherein the supervisor is configured to perform the following:

- receiving, from the sub-device, an assigned public key that identifies the sub-device when the sub-device requests access to the containerised application;

- authenticating the identity of the sub-device by verifying the private key to identify the application;

- directing the request data to a multicast channel when identities of the sub-device and the application are verified;

- transmitting the encrypted data from the sub-device to the application for processing.

[0071] In the application side, the application will check incoming data from the multicast channel, and only the data that is intended for the application will be received and described by a call-back function. The data for other applications will be ignored by the application.

[0072] In some embodiments, there is provided a computer program product comprising a plurality of data processor executable instructions that, when executed by a data processor in a system, cause the system to perform the method as detailed above.

[0073] In some embodiments, there is provided a gateway device for securing multicast communications. The gateway device comprises an authenticating module; a reporting module; and a key generation module. Incoming data is verified by managing the access remotely.

Brief Description of Drawings

[0074] Embodiments of the present disclosure will now be described by way of non-limiting example only with reference to the accompanying drawings, in which:

[0075] Figure 1 is a block diagram of a system for selectively processing data, according to some embodiments;

[0076] Figure 2 is a block diagram of another system for selectively processing data, according to some embodiments;

[0077] Figure 3 is a block diagram of another system for selectively processing data, according to some embodiments;

[0078] Figure 4 is a block diagram of another system for selectively processing data, according to some embodiments;

[0079] Figure 5 illustrates a table for enabling selective processing of sub-device data, according to some embodiments;

[0080] Figure 6 illustrates a computer-implemented method for securely initialising a containerised application, according to some embodiments;

[0081] Figure 7 illustrates another computer-implemented method for securely initialising a containerised application, according to some embodiments;

[0082] Figure 8 illustrates a computer-implemented method for selectively processing data, according to some embodiments; and

[0083] Figure 9 illustrates a block diagram of another system for selectively processing data, according to some embodiments.

Description of Embodiments

[0084] Particular embodiments of the present disclosure will now be described with reference to the accompanying drawings. The terminology used herein is for the purpose of describing particular embodiments only and is not intended to limit the scope of the present disclosure. Additionally, unless defined otherwise, all technical and scientific terms used herein have the same meanings as commonly understood by one of ordinary skill in the art to which this disclosure belongs.

[0085] The use of the term “application” as used herein refers generally to a unit of executable software that implements a certain functionality or theme.

[0086] The use of the term “network” as used herein refers to any type of telecommunications or data network.

[0087] A commercial, residential and/or industrial facility, for example, a manufacturing facility, energy production facility, resource extraction facility, agricultural facility, public facility (such as road networks or public parks), construction sites, etc. may rely on one of more systems that comprise a variety of devices, machines or equipment to execute complex processes and process operations. When executing these processes and process operations, the systems and the operators of the systems can be required to consider a variety of operational parameters. These operational parameters can influence the design, deployment and operation of the relevant systems.

[0088] In some cases, a system may comprise a plurality of sub-devices that are each configured to generate data relating to one or more operating parameters that are associated with that system. The sub-devices may be configured to generate data relating to a common operating parameter (e.g. temperature data associated with different locations of the system). Alternatively, the sub-devices may be configured to generate data relating to different operating parameters (e.g. one sub-device may be configured to generate temperature data and another sub-device may be configured to generate pressure data) of a region of the system. In

some cases, the sub-devices may be configured to generate data that is indicative of a state of an environment of the sub-device (e.g. video data).

[0089] The data generated by a particular sub-device may be used for a plurality of different purposes. For example, sensor data from a sensor in an elevator can be processed for the purposes of considering a maintenance schedule of the elevator, in addition to considering whether one or more operational parameters are within safety limits associated with the elevator. The data generated by the particular sub-device may be processed by a plurality of processing applications. Each processing application may be associated with a particular purpose. For example, in the case of the elevator, a first processing application may process the sensor data when considering the maintenance schedule of the elevator and a second processing application may process the sensor data when considering safety parameters of the elevator. The first processing application and the second processing application may process the same sensor data simultaneously.

[0090] The data generated by a particular sub-device may be used for a plurality of different purposes. For example, sensor data from a sensor in an elevator can be processed for the purposes of considering a maintenance schedule of the elevator, in addition to considering whether one or more operational parameters are within safety limits associated with the elevator. The data generated by the particular sub-device may be processed by a plurality of processing applications. Each processing application may be associated with a particular purpose. For example, in the case of the elevator, a first processing application may process the sensor data when considering the maintenance schedule of the elevator and a second processing application may process the sensor data when considering safety parameters of the elevator. The first processing application and the second processing application may process the same sensor data simultaneously.

[0091] Various virtualisation technologies and application container technologies enable computing devices to host a large number of applications without modification to the computing device, or with reduced modification requirements. For example, application containerisation is a virtualisation method used to deploy and run applications without launching an entire virtual machine for each application. Multiple isolated applications may run on a single host computing device and access the same operating system kernel.

[0092] Enabling processing applications to simultaneously process the same data provided by a sub-device for a plurality of different purposes may be technically difficult. A common data path between the processing applications may be provided to enable the data to be provided and processed by each of the processing applications as it is received.

[0093] Privacy and security concerns also become an issue when sharing data across multiple applications. It is therefore beneficial to provide a means for enabling select processing applications to receive data provided by certain sub-devices whilst preventing, or inhibiting other processing applications ability to receive the data. This disclosure provides a way of distributing sensor data among different applications using multicast network channels (e.g. IP multicast channels) and an access control mechanism between applications and the multicast network channels.

[0094] Existing systems are known to establish communication securely with a plurality of devices by connecting via an agent. In this case, all devices receive the same data which may be processed for various purposes. However, issues of privacy and security may arise when transmitting or receiving by multiple applications by multiple users. Furthermore, enabling security for multicast applications is challenging, with additional requirements to ensure content privacy among dynamic group memberships. In this instance, it is advantageous for specific device to receive specific data, unlike a device of a general communication system. Therefore, it is advantageous to provide a way to securely transmitting data only to a limited plurality of devices or applications as needed.

[0095] Advantageously, the present disclosure enables processing of the same incoming data by a plurality of processing applications. The incoming data can therefore be simultaneously processed to provide different data outputs for a variety of purposes. Furthermore, incoming data is selectively managed so that it is securely transmitted, and interacted with by a reduced number of processing applications.

[0096] It can also be advantageous to provide a system that enables secure installation and initialisation of containerised applications.

System for selectively processing data

[0097] Embodiments of the present disclosure provide a system for managing access to multicast communications and providing connectivity securely between communication devices wirelessly.

[0098] Figure 1 is a block diagram illustrating a system 100 that is configured to selectively process data. The system 100 may be referred to as a data management system or a system for processing data. The system 100, according to some embodiments, may be considered a system for managing access to multicast communications. The system 100 comprises a service device 102. The system 100 comprises a data management device 118. The system comprises a sub-device 107. The sub-device 107 is configured to generate sub-device data, and to provide the sub-device data to the service device 102. The service device 102 is configured to broadcast the sub-device data to a multicast network channel of a multicast network 116. The multicast network channel is associated with that sub-device 107, so, where there are multiple sub-devices 107, the sub-device data from each sub-device 107 is broadcast to a respective multicast network channel that is associated with the respective sub-device 107. A containerised application 112 of the service device 102 is configured to process the sub-device data to generate processed data. The service device 102 is configured to transmit the processed data to the data management device 118.

[0099] The service device 102 comprises at least one service device processor 104 in communication with service device memory 106. The at least one service device processor 104 is configured to execute service device program instructions stored in service device memory 106 to cause the service device 102 to function as described herein. In other words, the service device program instructions are accessible by the at least one service device processor 104, and are configured to cause the at least one service device processor 104 to function as described herein.

[0100] In some embodiments, the service device program instructions are in the form of program code. The at least one service device processor 104 comprises one or more microprocessors, central processing units (CPUs), application specific instruction set processors (ASIPs), application specific integrated circuits (ASICs) or other processors capable of reading and executing program code. In the illustrated embodiment, the service device program instructions comprise a supervisor module 108 as is described in more detail

herein. The service device program instructions also comprise a sub-device data module 110 as is described in more detail herein.

[0101] Service device memory 106 may comprise one or more volatile or non-volatile memory types. For example, service device memory 106 may comprise one or more of random access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM) or flash memory. Service device memory 106 is configured to store program code accessible by the at least one service device processor 104. The program code may comprise executable program code modules. In other words, service device memory 106 is configured to store executable code modules configured to be executable by the at least one service device processor 104. The executable code modules, when executed by the at least one service device processor 104 cause the at least one service device 102 to perform certain functionality, as described herein. In the illustrated embodiment, the supervisor module 108 and the sub-device data module 110 are in the form of program code stored in the service device memory 106.

[0102] At least one containerised application 112 is stored in service device memory 106. In the illustrated embodiment, a plurality of containerised applications 113 are stored service device memory 106. The plurality of containerised applications 113 may comprise a first containerised application 112, a second containerised application 112 etc. There may be N containerised applications 112 in the plurality of containerised applications 113. One or more containerised application 112 may be stored in a container 114. In some embodiments, the plurality of containerised applications 113 is stored in the container 114.

[0103] The system 100 comprises the data management device 118. The service device 102 is configured to communicate with the data management device 118 using a communications network 120. The system 100 comprises a sub-device 107. In particular, the system 100 comprises a plurality of sub-devices 109. The service device 102 is configured to communicate with each sub-device 107 of the plurality of sub-devices 109 using the communications network 120. Throughout the disclosure, a sub-device may also be referred to as data collection device.

[0104] In some embodiments, the service device 102 comprises a service device network interface (not shown). The service device network interface is configured to enable the

service device 102 to communicate with other computing devices using one or more communications networks (e.g. the communications network 120). For example, the service device network interface is configured to enable the service device 102 to communicate with the data management device 118 and/or each sub-device 107 of the plurality of sub-devices 109. The service device network interface may comprise a combination of network interface hardware and network interface software suitable for establishing, maintaining and facilitating communication over a relevant communication channel. Examples of a suitable communications network include a cloud server network, wired or wireless internet connection, BluetoothTM or other near field radio communication, and/or physical media such as a Universal Serial Bus (USB) connection.

[0105] In some embodiments, the service device 102 comprises a service device user interface (not shown), whereby one or more user(s) can submit requests to the service device 102, and whereby the service device 102 can provide outputs to the user(s). The service device user interface may comprise one or more user interface components, such as one or more of a display device, a touch screen display, a keyboard, a mouse, a camera, a microphone, buttons, switches and lights. Alternatively, the user(s) may connect to, and control the service device 102 remotely using the service device user interface. For example, the service device user interface may be in the form of a portal accessible over a network (e.g. the communications network 120) from another computing device.

[0106] The service device 102 comprises a multicast network 116. The multicast network 116 comprises a plurality of multicast network channels. The multicast network 116 may also be referred to as a multicast bus. The service device 102 is configured to provide sub-device data that is received from a sub-device 107 to a multicast network channel. In some embodiments, each sub-device 107 of the plurality of sub-devices 109 is associated with its own multicast network channel. That is, the sub-device data provided by a first sub-device 107 of the plurality of sub-devices 109 may be provided to a first multicast network channel. Similarly, the sub-device data provided by a second sub-device 107 of the plurality of sub-devices 109 may be provided to a second multicast network channel. The multicast network 116 may comprise the same number of multicast network channels as there are sub-devices 107 that are in communication with the service device 102. Each sub-device 107 of the plurality of sub-devices 109 may be associated with its own multicast network channel

to which its sub-device data is provided. Providing sub-device data to a multicast network channel may comprise streaming the sub-device data to the network channel.

[0107] Each containerised application 112, when executed by the at least one service device processor 104, is configured to receive data from the multicast network 116. In particular, each containerised application 112 is configured to receive data from one or more of the multicast network channels. In other words, the at least one service device processor 104 is configured to provide data to one or more of the containerised applications 112 of the plurality of containerised applications 113 using the multicast network 116.

[0108] The service device 102 manages the access each sub-device 107 of the plurality of sub-devices 109 has to each containerised application 112 of the plurality of containerised applications 113. The service device 102 may control or provide access to authorised sub-devices 107 and applications 112 installed on the service device 102. Access control may be provided when users (i.e. an operator of the service device 102 and/or the data management device 118) permit access of their service device 102 and the installed application 112 on each service device 102. Where there are multiple different applications 112 installed on each service device 102 under different users, access control may be provided by the specific owner of the service device 102 and the application 112. Taken as a whole, there is provided a method of efficient management of access to allow selective interaction of containerised applications 112 with incoming data from the sub-devices.

[0109] The service device 102 (also known as an IGX) connects to one or more of the plurality of sub-devices 109, and is configured to facilitate access by propagating sub-device data from the plurality of sub-devices 109 to a multiple number of containerised applications (which may also be referred to as virtual appliances). The service device 102 comprises a supervisor unit 108 configured to manage the access or rights to a group or identity. The supervisor unit 108 may also be referred to as supervisor module. The supervisor unit 108 further provides a secure medium for information exchange. As such, the supervisor unit 108 performs various actions associated with security control. These can include managing the receipt, storage, sending and interpretation of entitlements. The service device 102 also comprises the multicast network 116, which is configured to enable segregating logically the containerised applications 112 by interacting with the sub-device data from authorised

sub-devices 107. Advantageously, there is selective processing of sub-device data from the authorised sub-devices 107 by the containerised applications 112.

[0110] Embodiments of the present disclosure provide a method for managing the access to multicast communications for providing connectivity securely between communication devices wirelessly.

[0111] The data management device 118 comprises at least one data management device processor 122 in communication with data management device memory 124. The at least one data management device processor 122 is configured to execute data management device program instructions stored in data management device memory 124 to cause the data management device 118 to function as described herein. In other words, the data management device program instructions are accessible by the at least one data management device processor 122, and are configured to cause the at least one data management device processor 122 to function as described herein. In some embodiments, the data management device program instructions are in the form of program code. The at least one data management device processor 122 comprises one or more microprocessors, central processing units (CPUs), application specific instruction set processors (ASIPs), application specific integrated circuits (ASICs) or other processors capable of reading and executing program code.

[0112] At least one available containerised application 126 is stored in data management device memory 124. In particular, a plurality of available containerised applications 127 are stored in data management device memory 124. The available containerised applications 127 comprises containerised applications 126 that are available for the service device 102 to retrieve, when the service device 102 has the correct authorisation. The plurality of available containerised applications 127 may be made available within a marketplace 128. The marketplace 128 is hosted by the data management device 118.

[0113] The data management device 118 is configured to communicate with the service device 102 using the communications network 120. The data management device 118 is configured to communicate with the plurality of sub-devices 109 using the communications network 120.

[0114] In some embodiments, the data management device 118 comprises a data management device network interface (not shown). The data management device network interface is configured to enable the data management device 118 to communicate with other computing devices using one or more communications networks (e.g. the communications network 120). For example, the data management device network interface is configured to enable the data management device 118 to communicate with the service device 102 and/or each sub-device 107 of the plurality of sub-devices 109. The data management device network interface may comprise a combination of network interface hardware and network interface software suitable for establishing, maintaining and facilitating communication over a relevant communication channel.

[0115] The data management device 118 comprises a data management service (or software). The data management device 118 may also be referred to in this disclosure as a DLC. The data management device 118 is configured to connect to the service device 102 via a communication means. The communication means may be the communications network 120. Throughout this disclosure, the service device 102 may also be referred to as an IGX.

[0116] The data management device 118 is configured to receive processed sub-device data from the service device 102. The data management device 118 is configured to manage the processed sub-device data centrally. In some embodiments, the data management device 118 is configured to receive processed sub-device data from a plurality of services devices 102. In these cases, the data management device 118 is configured to integrate data from one or more service device 102.

[0117] In some embodiments, the data management device 118 comprises a data management device user interface (not shown), whereby one or more user(s) can submit requests to the data management device 118, and whereby the data management device 118 can provide outputs to the user(s). The data management device user interface may comprise one or more user interface components, such as one or more of a display device, a touch screen display, a keyboard, a mouse, a camera, a microphone, buttons, switches and lights. Alternatively, the user(s) may connect to, and control the data management device 118 remotely using the data management device user interface. For example, the data

management device user interface may be in the form of a portal accessible over a network from another computing device.

[0118] The data management device 118 hosts the marketplace 128 of available containerised applications 127. The marketplace 128 may provide a source of software applications and services, including native cloud applications. The marketplace 118 of applications may provide a number of marketplace functions and services for the various types of data processing applications, for multiple purposes. An example of such marketplace is a cloud marketplace.

[0119] In addition, the data management device 118 manages access control and may be configured to enable access to user, the service device 102, sub-devices 107, multicast channels, containerised applications 112 and licenses and subscription of applications. The data management device 118 manages the configured access mechanisms for remote resource services and secure connections to remote networks.

[0120] The data management device 118 also comprises an application deployment module (not shown), in which the hosted applications of the marketplace 128 may be provided. The deployment module manages the deployment of the available containerised applications 126 and applications in the virtual machine (the service device 102), such as downloading, installation or the removal of the containerised applications 112 and virtual appliances. The data management device 118 may be configured to communicate with the service device 102 to enable selective deployment of available containerised applications 127 and control accessibility of the containerised applications 112 with respect to the multicast network channels.

[0121] As previously described, the system 100 comprises a plurality of sub-devices 109. Each sub-device 107 of the plurality of sub-devices 109 comprises at least one sub-device processor 130 in communication with sub-device memory 132. The at least one sub-device processor 130 is configured to execute sub-device program instructions stored in sub-device memory 132 to cause the sub-device 107 to function as described herein. In other words, the sub-device program instructions are accessible by the at least one sub-device processor 130, and are configured to cause the at least one sub-device processor 130 to function as described herein. In some embodiments, the sub-device program instructions are in the form of

program code. The at least one sub-device processor 130 comprises one or more microprocessors, central processing units (CPUs), application specific instruction set processors (ASIPs), application specific integrated circuits (ASICs) or other processors capable of reading and executing program code.

[0122] Sub-device memory 132 may comprise one or more volatile or non-volatile memory types. For example, sub-device memory 132 may comprise one or more of random access memory (RAM), read-only memory (ROM), electrically erasable programmable read-only memory (EEPROM) or flash memory. Sub-device memory 132 is configured to store program code accessible by the at least one sub-device processor 130. The program code may comprise executable program code modules. In other words, sub-device memory 132 is configured to store executable code modules configured to be executable by the at least one sub-device processor 130. The executable code modules, when executed by the at least one sub-device processor 130 cause the sub-device 107 to perform certain functionality, as described herein.

[0123] Sub-device information is stored in sub-device memory 132. The sub-device information is related to the particular sub-device 107. The sub-device information may comprise one or more of a sub-device serial number, a sub-device Media Access Control (MAC) address and a sub-device model number, a sub-device identifier and a random key. The sub-device information is stored in one or more memory blocks of the sub-device memory 132 of the relevant sub-device 107. The one or more memory blocks may be predetermined. That is, the one or more memory blocks may be known and specifically designated to store the sub-device information.

[0124] The sub-device information may be stored in service device memory 106. For example, the random key may be stored in a Trusted Platform Module (TPM) of service device memory 106. The sub-device 107 may retrieve the random key from the TPM of the service device.

[0125] Upon an initial boot, the service device 102 checks for the presence of a TPM module in service device memory 106. If a TPM module is visible to the at least one service device processor 104, the at least one service device processor 104 creates a random key. The at least one service device processor 104 stores the random key in service device

memory 106. The random key may be stored in a database. The random key may be 16 bytes. The at least one service device processor 104 creates a hardware signature. The at least one service device processor 104 creates the hardware signature based on a Media Access Control (MAC) address of a Local Area Network (LAN) port of the service device 102. The hardware signature may be generated using a SHA256 checksum generator. For example, the hardware signature may be generated by running the MAC address through the SHA256 checksum generator. The at least one service device processor 104 stores the hardware signature in service device memory 106. The at least one service device processor 104 combines the hardware signature with the random key to form a combined key. The combined key may be 32 bytes. The at least one service device processor 104 stores the combined key in service device memory 106. The at least one service device processor 104 may appropriately format and mount a partition of the service device memory 106 that is associated with the random key. The at least one service device processor 104 may then execute a software module (e.g. the supervisor unit 108) to perform functionality described herein.

[0126] Each sub-device 107 of the plurality of sub-devices 109 is configured to generate sub-device data. The sub-device data is indicative of one or more sub-device parameters. As illustrated in Figure 1, a sub-device 107 may comprise a sub-device sensor 134 that is configured to respond to a sensor parameter. The one or more sub-device parameters of that sub-device 107 comprises the sensor parameter. A sensor characteristic associated with the sub-device sensor 134 may change in response to a change in the sensor parameter. The sub-device 107 is configured to measure the sensor characteristic to determine a value that is indicative of the sensor characteristic. The sub-device 107 is configured to store the value that is indicative of the sensor characteristic. The sub-device 107 may store the value in sub-device memory 132. In some embodiments, the sub-device 107 is configured to store a time-series of values of the sensor characteristic. The sub-device 107 may store the time-series in sub-device memory 132. The sub-device data comprises the value that is indicative of the sensor characteristic, or the time-series of values of the sensor characteristic.

[0127] For example, the sub-device sensor 134 may be in the form of a temperature sensor. A resistance of the temperature sensor may change in response to a change in temperature of the temperature sensor. In such a case, the sensor characteristic may be the resistance of the temperature sensor. Furthermore, the value that is indicative of the sensor characteristic may

be the value of the resistance of the temperature sensor at a given time, which can be correlated to a temperature of the temperature sensor.

[0128] In some embodiments, one or more of the sub-devices 107 may comprise one or more of a temperature sensor, a pressure sensor, an acceleration sensor, a vibration sensor, a position sensor (e.g. a Global Positioning System module configured to determine a position estimate of the sub-device), a chemical sensor, an acoustic sensor, a flow sensor, a light sensor (e.g. an optical camera), a humidity sensor, a motion sensor, a force sensor, a magnetic sensor, an antenna and a Near Field Communication sensor. In other words, one or more of the sub-device sensors 134 may comprise, or be in the form of at least one of the aforementioned sensors.

[0129] In some embodiments, the system 100 may be considered an “Internet of Things” (IoT) system. Therefore, the service device 102 may be considered an IoT gateway device. Furthermore, each of the sub-devices 102 may be referred to as an IoT device.

[0130] Each sub-device 107 is configured to communicate with the service device 102 using the communications network 120. In some embodiments, one or more of the sub-devices 107 are configured to communicate with the data management device 118 using the communications network 120.

[0131] In some embodiments, each sub-device 107 of the plurality of sub-devices 109 comprise a respective sub-device network interface (not shown). The sub-device network interface is configured to enable the respective sub-device 107 to communicate with other computing devices using one or more communications networks (e.g. the communications network 120). For example, the sub-device network interface is configured to enable the sub-device 107 to communicate with the service device 102, another sub-device 107 and/or the data management device 118. The sub-device network interface may comprise a combination of network interface hardware and network interface software suitable for establishing, maintaining and facilitating communication over a relevant communication channel.

[0132] In some embodiments, each sub-device 107 comprises a respective sub-device user interface (not shown), whereby one or more user(s) can submit requests to the respective

sub-device 107, and whereby the sub-device 107 can provide outputs to the user(s). The sub-device user interface may comprise one or more user interface components, such as one or more of a display device, a touch screen display, a keyboard, a mouse, a camera, a microphone, buttons, switches and lights. Alternatively, the user(s) may connect to, and control the sub-device 107 remotely using the sub-device user interface. For example, the sub-device device user interface may be in the form of a portal accessible over a network from another computing device.

[0133] The service device 102 is configured to connect to one or more of the plurality of sub-devices 109 via the communication means. In some embodiments, the data management device 118, the service device 102 and the plurality of sub-devices 109 are configured to connect for the communication of data from one device to another device.

[0134] In some embodiments, the communications network 120 can comprise a plurality of communications networks. For example, the service device 102 may be configured to communicate with the data management device 118 using a first communications network. The service device 102 may be configured to communicate with one or more of the sub-devices 107 using another communications network. In this case, the communications network 120 may be said to comprise a first communications network and a second communications network.

[0135] In some embodiments, the service device 102 is configured to communicate with each sub-device 107 using a different communications network. In these embodiments, the service device 102 may be configured to communicate with the data management device 118 using a data management device communications network. Furthermore, the service device 102 may be configured to communicate with each sub-device 107 using a respective sub-device communications network. In these embodiments, the communications network 120 may be said to comprise the data management device communications network (a first communications network) and the plurality of sub-device communications networks (second, third etc. communications networks).

[0136] Figure 2 is a block diagram illustrating another embodiment of the system 100. In the embodiment of Figure 2, the service device 102 is configured to communicate with the data management device 118 using a first communications network 120A. In the embodiment

of Figure 2, the system 100 comprises Q sub-devices 107. Each sub-device 107 is configured to communicate with the service device 102 using a respective sub-device communications network 120B(A-Q). In some embodiments, the previously described communications network 120 can be said to comprise the first communications network 120A and each of the sub-device communications networks 120V(A-Q). The service device network interface and the respective sub-device network interfaces may facilitate this communication. Examples of a suitable first communications network 120A include a cloud server network, wired or wireless internet connection, Bluetooth™ or other near field radio communication, and/or physical media such as a Universal Serial Bus (USB) connection. Examples of suitable sub-device communications networks 120B(A-Q) include cloud server networks, wired or wireless internet connections, Bluetooth™ or other near field radio communications, and/or physical media such as a Universal Serial Bus (USB) connections.

[0137] Figure 3 is a block diagram illustrating another embodiment of the system 100. In the embodiment of Figure 3, the service device 102 is configured to communicate with the data management device 118 using the first communications network 120A. Each sub-device 107 is configured to communicate with the service device 102 using a respective sub-device communications network 120B(A-Q). In the embodiment of Figure 3, the previously described multicast network 116 is in the form of a multicast bus 116A. Each sub-device 107 of the plurality of sub-devices 109 is configured to communicate with the multicast bus 116A. In particular, the sub-device data that each sub-device 107 provides to the service device 102 is provided to the multicast bus 116A. This may be as described with reference to the multicast system 116. For example, the at least one service device processor 104 may be configured to provide the sub-device data to the multicast bus 116A as is described herein (e.g. with reference to the multicast network 116).

[0138] As illustrated in Figure 3, one or more of the containerised applications 112 is configured to receive data from the multicast bus 116A. In some embodiments, sub-device data provided by the sub-devices 107 is provided directly to the multicast bus 116A as illustrated in Figure 3. In some embodiments, the at least one service device processor 104 receives sub-device data provided by the sub-devices 107 (e.g. via the sub-device network interface) and provides the sub-device data to the multicast bus 116A as is described herein.

[0139] Figure 4 is a block diagram illustrating another embodiment of the system 100. In some embodiments, such as that shown in Figure 4, one or more of the sub-devices 107 is configured to generate a respective data stream 142A-Q. The data stream 142A-Q may be a data stream of sub-device data. Each sub-device 107 is configured to provide the data stream 142A-Q to a forwarder 138 via a machine-to-machine communication system 136. In some embodiments, the machine-to-machine communication system 136 and the forwarder 138 may be the at least one service device processor 104. In some embodiments, the machine-to-machine communication system 136 may comprise the service device network interface and/or the sub-device network interfaces.

[0140] In the embodiment illustrated in Figure 4, the machine-to-machine communication system 136 is configured to aggregate the data streams 142A-Q provided by each of the sub-devices 107 into an aggregated data stream 144. The machine-to-machine communication system 136 is configured to provide the aggregated data stream 144 to the forwarder 138. The forwarder 138 is configured to provide (or forward) the aggregated data stream 144 to the multicast bus 116A of the service device 102. The multicast bus 116A is configured to provide one data stream of the plurality of data streams 142A-Q to each multicast network channel of the multicast bus 116A.

[0141] The system 100 illustrated in Figure 4 comprises a plurality of containers 114A-Q. Container A 114A may be considered a first container. Container B may be considered a second container. Container Q may be considered a *Qth* container. Each of the containerised applications 112 are configured to operate within a respective container of the plurality of containers 114A-Q.

[0142] The system 100 comprises a plurality of Software Development Kits (SDKs) 140A-Q. Each containerised application 112 is associated with a respective SDK 140A-Q. In some embodiments, each containerised application 112 comprises a respective SDK 140A-Q. Each SDK 140A-Q checks for sub-device data on a multicast network channel. The SDK 140A-Q receives sub-device data intended for the corresponding containerised application 112 for processing. The sub-device data from other sources (i.e. the sub-device data of other multicast network channels) is not received by that SDK 140A-Q or processed by the corresponding containerised application 112.

[0143] As illustrated in Figure 4, SDK A 140A, which may be referred to as a first SDK, receives sub-device data from the data stream 142A of a first sub-device 107 and the data stream 142B of a second sub-device 107 and provides these data streams to a first containerised application 112. SDK B 140B, which may be referred to as a second SDK, receives sub-device data from each of the data streams 140A-Q and provides these data streams to a second containerised application 112. Thus, in some embodiments, the SDKs enable selective provision of data streams to the containerised applications 112. SDKs 140A-Q enables secure communications between the containerised applications 112 and the plurality of sub-devices 109. The SDKs 140A-Q enable authentication, authorisation and encryption/decryption of data packets, and are thereby employed to enable secure communication.

[0144] Although the system 100 of Figure 4 comprises the plurality of containers 114A-Q, it will be understood that each of the containerised applications 112 may operate within one container 114 (i.e. a common container).

[0145] Figure 9 is a block diagram illustrating another embodiment of the system 100. In some embodiments, such as that shown in Figure 9, one or more containerised applications 112 are configured to be in communication with the forwarder 138. The forwarder 138 may be as previously described. The one or more containerised applications 112 are configured to be in communication with the forwarder over a UDP data communication network 139. The forwarder 138 is also configured to be in communication with other devices via a Machine-to-Machine MODBUS module 141. The forwarder 138 is also configured to be in communication with other devices via a Machine-to-Machine EIP module 143. The forwarder 138 is also configured to be in communication with other devices via a Machine-to-Machine MQTT module 147. For example, the forwarder 138 is configured to communicate with the data management device 118 using the Machine-to-Machine MQTT module 147. The forwarder 138 may also be configured to communicate with the service device memory 106 via the Machine-to-Machine MQTT module 147. The forwarder 138 is configured to communicate with other devices via a Machine-to-Cloud DDS module 149. The forwarder 138 may be configured to communicate with the containerised applications 112 via the UDP data communication network 139. The forwarder 138 may be configured to communicate with another device using the Machine-to-Machine MODBUS module 141, the Machine-to-Machine EIP module 143, the Machine-to-Machine MQTT

module 147, the data management device 118, the device memory 106 and the Machine-to- Cloud DDS module 149 via the UDP data communication network 139.

[0146] The system 100 of Figure 9 comprises a supervisor 108. The supervisor 108 may be in the form of the supervisor module 108 described herein. The supervisor 108 is configured to communicate with other devices via one or more of the Machine-to-Machine MODBUS module 141, Machine-to-Machine EIP module 143, Machine-to-Machine MQTT module 147 and Machine-to-Cloud DDS module 149 and a TCP/IP communication network 151. The supervisor 118 may be configured to communicate with the containerised applications 112 via a UDP control communication network 153.

[0147] In some embodiments, the service device 102 comprises one or more of the containerised applications 112, forwarder 138, supervisor 108, the Machine-to-Machine MODBUS module 141, the Machine-to-Machine EIP module 143, the Machine-to-Machine MQTT module 147 and the Machine-to- Cloud DDS module 149.

[0148] Figure 5 illustrates a table 145 for enabling selective processing of sub-device data by specific containerised applications 114. The mapping table 145 identifies which containerised applications 112 are authorised to process sub-device data provided by a particular sub-device 107. In some embodiments, the mapping table 145 defines the mapping of a sub-device 107, a deployed containerised application 112, as well as the destination of the processed sub-device data. The mapping table 145, or a portion thereof, may be stored in the service device memory 106. The mapping table 145, or a portion thereof, may also be stored in the data management device memory 124. The mapping table 145, or a portion thereof, may be stored in the sub-device memory 132. In other words, the identities are known for both the sub-device 107, from which sub-device data is received and the containerised application 112, to which the sub-device data is to be transmitted or provided for processing. Both the sub-device 107 and the containerised application 112 are authenticated, by verifying an identity of the sub-device 107, to ensure authenticity prior to deploying the containerised application 112. The sub-device 107 and the containerised application 112 are authenticated as is described herein.

Method for securely initialising a containerised application

[0149] Referring to Figures 6 and 7, in accordance with embodiments of the present disclosure, there is provided a method of establishing an exchange of information between the service device 102 and the containerised applications 112 by remote access management. This may be by way of a supervisor application, which may also be referred to as a supervisor module 108.

[0150] The method comprises the following steps:

Step 1: Installing a containerised application 112.

- a) Initiating request to install an application selected from the marketplace 128 of the data management device 118.
- b) Obtaining a license key that corresponds to the selected application.
- c) Installing the selected application.

Step 2: Initiating and running the selected application by providing the license key corresponding to the application.

Step 3: Limiting access to the selected application.

- a) Assigning private keys to sub-devices to identify authorised users.
- b) Authenticating, by requesting the private key, when a sub-device sends request data to request access to the selected application.
- c) Directing the request data to a multicast channel if the sub-device is authorised, and rejecting any unauthorised requests.

Step 4: Selectively manage the interaction of data flow by transmitting the multicast data to the selected application for processing. All other data from other sources is not processed.

[0151] In some embodiments, the service device 102 is configured to selectively process data by way of data control. This data control may be defined by selecting the source data to be processed. The source data can be in the form of sensor data, or sub-device data. Data control may also be defined by selecting the containerised applications 112 to process the source data. Containerised applications 112 may be selected to be deployed to permit processing the data from the source. Data control may be further configured to selectively process the source data by transmitting the data to an allowed destination. This is advantageous to control the interaction of incoming data with selected containerised applications 112, by giving access to particular data feed to a particular containerised application 112, thereby providing a secure environment.

[0152] Figure 6 illustrates a computer-implemented method 200 as performed by the system 100 for securely initialising a containerised application 152. Figure 6 is to be understood as a blueprint for a software program and may be implemented step-by-step, such that each step in Figure 6 is represented by a function in a programming language, such as C++ or Java. The resulting source code is then compiled and stored as computer executable instructions on one or more of service device memory 106 and data management device memory 124, or implemented in a different computer environment. In some embodiments, the containerised application 152 is one of the plurality of available containerised applications 127 described previously.

[0153] At 202, the service device 102 transmits a request to install a containerised application 152 to the data management device 118. In particular, the at least one service device processor 104 generates the request and transmits the request to the data management device 118. The containerised application 152 may be a new containerised application 152. That is, the containerised application 152 may not be installed on the service device 102 at the time the request is transmitted. The at least one service device processor 104 may transmit the request to the data management device using the service device network interface and the communications network 120 (or the first communications network described previously, where applicable). The request comprises a containerised application identifier that identifies the containerised application 152. The containerised application 152 may be one of a plurality of available containerised applications 127 that are available on the marketplace 128.

[0154] The service device 102 (e.g. the at least one service device processor 104) may transmit the request to initialise the containerised application 152 to the data management device 118 in response to the service device 102 receiving a request for access from a sub-device 107 that is associated with the containerised application 152. For example, a particular sub-device 107 may be configured to generate sub-device data for processing by the containerised application 152. A user of the system 100 may attempt to connect this sub-device 107 to the service device 102 and/or the data management device 118. In the case where the service device 102 does not already have the containerised application 152 necessary to process the sub-device data produced by that sub-device 107 installed, the sub-device data generated by the sub-device will not be able to be processed. In response to this, the service device 102 may transmit the request to initialise the containerised application 152 to the data management device 118.

[0155] At 204, the data management device 118 receives the request to initialise the containerised application 152 from the service device 102. In particular, the at least one data management device processor 122 receives the request to initialise the containerised application 152 from the service device 102. The at least one data management device processor 122 may receive the request using the data management device network interface and the communications network 120 (or the first communications network 120A described previously, where applicable).

[0156] At 206, the data management device 118 transmits a license key that is associated with the containerised application 152 to the service device 102. In particular, the at least one data management device processor 122 transmits the license key that is associated with the containerised application 152 to the service device 102. The license key may be stored in data management device memory 124. The at least one data management device processor 122 may transmit the license key using the data management device network interface and the communications network 120 (or the first communications network 120A described previously, where applicable).

[0157] At 208, the service device 102 receives the license key from the data management device 118. In particular, the at least one service device processor 104 receives the license key from the data management device 118. The at least one service device processor 104 may receive the license key using the service device network interface and the communications network 120 (or the first communications network 120A described previously, where applicable).

[0158] At 210, the service device 102 transmits the license key to a license validation service 146. In particular, the at least one service device processor 104 transmits the license key to a license validation service 146. The at least one service device processor 104 may transmit the license key using the service device network interface and the communications network 120 (or the first communications network 120A described previously, where applicable).

[0159] The license validation service 146 may be a third party that is authorised to validate license keys associated with containerised applications 112 of the plurality of available containerised applications 127. In some embodiments, the owner and/or operator of the data

management device 118 and/or service device 102 may also be an owner and/or operator of the license validation service 146.

[0160] At 212, the license validation service 146 receives the license key from the service device 102. The license validation service 146 may receive the license key using the communications network 120 (or the first communications network 120A described previously, where applicable).

[0161] At 214, the license validation service 146 validates the license key. The license validation service 146 may validate the license key by comparing the license key to a pre-existing license key register. Alternatively, the license validation service 146 may validate the license key by using the license key as an input to a license key validation algorithm, and validating the output of the license key validation algorithm.

[0162] At 216, the license validation service 146 transmits a license validation to the service device 102. The license validation is indicative of the validity of the license key. For example, the license validation may indicate that the license key is valid where this has been determined by the license validation service 146. Similarly, the license validation may indicate that the license key is not valid where this has been determined by the license validation service 146. The license validation service 146 may transmit the license validation using the communications network 120 (or the first communications network 120A described previously, where applicable).

[0163] At 218, the service device 102 receives the license validation. In particular, the at least one service device processor 104 receives the license validation from the license validation service 146. The license validation may indicate that the license key is valid. The license validation may indicate that the license key is invalid. The service device 102 may receive the license validation using the service device network interface and/or the communications network 120 (or the first communications network 120A described previously, where applicable).

[0164] At 220, the service device 102 retrieves the containerised application 152. In particular, the at least one service device processor 104 retrieves the containerised application 152 and stores the containerised application 152 in service device memory 106.

In some embodiments, retrieving the containerised application 152 comprises downloading the containerised application 152 from a software repository. In some embodiments, retrieving the containerised application 152 comprises downloading the containerised application 152 from the data management device 118. In particular, retrieving the containerised application 152 comprises downloading the containerised application 152 from the marketplace 128. In some embodiments, the service device 102 may retrieve the containerised application 152 using the service device network interface and/or the communications network 120 (or the first communications network 120A described previously, where applicable).

[0165] Retrieving the containerised application 152 may also comprise installing the containerised application 152 on the service device 102.

[0166] At 222, the service device 102 initialises the containerised application 152. The service device 102 may initialise the containerised application 152 using the license key. In particular, the at least one service device processor 104 initialises the containerised application 152. The at least one service device processor 104 may initialise the containerised application 152 by executing the containerised application 152. After initialisation, the at least one service device processor 104 is configured to execute the containerised application 152 to process sub-device data.

[0167] At 224, the service device 102 transmits an installation confirmation to the data management device 118. In particular, the at least one service device processor 104 transmits the installation confirmation to the data management device 118. The installation confirmation may indicate a successful installation of the containerised application 152. Alternatively, if the containerised application 152 fails to install, or fails to initiate correctly, the installation confirmation may indicate an unsuccessful installation.

[0168] Transmitting an installation confirmation to the data management device 118 may comprise generating the installation confirmation. In some embodiments, the service device 102 may transmit the installation confirmation using the service device network interface and/or the communications network 120 (or the first communications network 120A described previously, where applicable).

[0169] Figure 7 illustrates an alternative computer-implemented method 300 as performed by the system 100 for securely initialising the containerised application 152. Figure 7 is to be understood as a blueprint for a software program and may be implemented step-by-step, such that each step in Figure 7 is represented by a function in a programming language, such as C++ or Java. The resulting source code is then compiled and stored as computer executable instructions on one or more of service device memory 106 and data management device memory 124, or implemented in a different computer environment. In some embodiments, the containerised application 152 is one of the plurality of available containerised applications 127 described previously.

[0170] In the method 300, the license key is validated after initialisation of the containerised application 152. That is, steps 202, 204, 206, 208 and 220 described with reference to the method 200 are applicable to the method 300. In some embodiments, steps 202, 204, 206, 208 and 220 described with reference to the method 200 precede step 302 of the method 300.

[0171] At 302, the service device 102 initialises the containerised application 152. In particular, the at least one service device processor 104 initialises the containerised application 152. The at least one service device processor 104 may initialise the containerised application 152 by executing the containerised application 152.

[0172] At 304, the service device 102, by executing the containerised application 152, generates a license key validation request. In particular, at least one service device processor 104, by executing the containerised application 152, generates the license key validation request. The license key validation request may comprise the licence key.

[0173] At 306, the service device 102 transmits the license key validation request to the license validation service 146. In particular, the at least one service device processor 104 transmits the license key validation request to the license validation service 146. The at least one service device processor 104 may transmit the license key validation request using the service device network interface and/or the communications network 120 (or the first communications network 120A described previously, where applicable).

[0174] At 308, the license validation service 146 receives the license key from the service device 102. The license validation service 146 may receive the license key using the

communications network 120 (or the first communications network 120A described previously, where applicable).

[0175] At 310, the license validation service 146 validates the license key. The license validation service 146 may validate the license key by comparing the license key to a pre-existing license key register. Alternatively, the license validation service 146 may validate the license key by using the license key as an input to a license key validation algorithm, and validating the output of the license key validation algorithm.

[0176] At 312, the license validation service 146 transmits a license validation to the service device 102. The license validation is indicative of the validity of the license key. For example, the license validation may indicate that the license key is valid where this has been determined by the license validation service 146. Similarly, the license validation may indicate that the license key is not valid where this has been determined by the license validation service 146. The license validation service 146 may transmit the license validation using the communications network 120 (or the first communications network 120A described previously, where applicable).

[0177] At 314, the service device 102 provides the license validation to the containerised application 152 being executed by the service device 102. The service device 102 also provides a private key to the containerised application 152. The private key enables the containerised application to decrypt encrypted sub-device data that is received from the multicast network 116 as is described herein. After receiving the license validation, the at least one service device processor 104 is configured to execute the containerised application 152 to process sub-device data.

Method for selectively processing data

[0178] Figure 8 illustrates a computer-implemented method 400 as performed by the system 100 for selectively processing data. Figure 8 is to be understood as a blueprint for a software program and may be implemented step-by-step, such that each step in Figure 8 is represented by a function in a programming language, such as C++ or Java. The resulting source code is then compiled and stored as computer executable instructions on one or more

of service device memory 106, data management device memory 124 and sub-device memory 132, or implemented in a different computer environment.

[0179] When a user requests to deploy a new containerised application (e.g. one of the plurality of available containerised applications 127) on the service device 102, the data management device 118 transmits commands and configuration data (such as license key, application ID and mapping table) to the service device 102. The service device 102 checks and verifies the configuration data. The configuration data will be stored in a database if the data is correct and correctly formatted.

[0180] At 402, the service device 102 receives a request for access from a sub-device 107. In particular, the at least one service device processor 104 receives the request for access from the sub-device 107. In some embodiments, the service device 102 may receive the request for access using the service device network interface, via the communications network 120 or the relevant sub-device communications network 120B(A-Q). The request for access may be associated with one or more containerised applications 112 of the plurality of containerised applications 113. That is, the sub-device 107 may, with the request for access, request access to one or more containerised applications 112 of the plurality of containerised applications 113. The relevant containerised applications 112 may be configured to process sub-device data generated by the sub-device 107.

[0181] The request for access comprises sub-device information that is associated with the sub-device 107. For example, the sub-device information may comprise one or more of a sub-device serial number, a sub-device Media Access Control (MAC) address, a sub-device model number, a sub-device identifier and a random key. The random key may be as previously described.

[0182] At 404, the request for access is authorised based at least in part on the sub-device information. In some embodiments, the service device 102 receives authorisation to enable access. In particular, the at least one service device processor 104 receives authorisation the request for access. For example, the at least one service device processor 104 may authorise the request for access by comparing the sub-device information to a pre-existing database of authorised sub-devices. By this comparison, the at least one service device processor 104 may confirm that the sub-device 107 that is associated with the received request for access is

authorised to communicate with the service device 102 and/or one or more containerised application 112 of the plurality of containerised applications 113. Confirming that the sub-device 107 that is associated with the received request for access is authorised to communicate with the service device 102 and/or one or more containerised application 112 of the plurality of containerised applications 113, the at least one service device processor 104 may be considered receiving authorisation to enable access. In some embodiments, the at least one service device processor 104 may generate an authorisation token in response to confirming that the sub-device 107 that is associated with the received request for access is authorised to communicate with the service device 102 and/or one or more containerised application 112 of the plurality of containerised applications 113. The authorisation token may be provided to another component or software module of the service device 102, which may also be considered receiving authorisation to enable access. In some embodiments, the at least one service device processor 104 executes the supervisor module 108 to authorise the request for access.

[0183] Alternatively, the service device 102 may be configured to receive authorisation credentials, receipt of which comprises authorisation of the sub-device 107. The service device 102 may, for example, receive authorisation credentials via the service device user interface (i.e. login details of the user). Alternatively, the service device 102 may receive the authorisation credentials from another computing device such as the data management device 118 or the sub-device 107. The authorisation credentials may be associated with the sub-device information, for example, in a database. Receipt of authorisation credentials may be considered receiving authorisation to enable access.

[0184] In some embodiments, the service device 102 transmits the request for access or the sub-device information to another computing device which authorises the request for access. For example, at least one service device processor 104 may transmit the request for access or the sub-device information to the data management device 118. The data management device 118 may then authorise the request for access. For example, the data management device 118 may authorise the request for access by comparing the sub-device information to a pre-existing database of authorised sub-devices to confirm that the sub-device 107 that is associated with the received request for access is authorised to communicate with the service device 102. This may be executed, for example, by the at least one data management device processor 122.

[0185] Alternatively, the data management device 118 and in particular, the at least one data management device processor 122 may be configured to receive authorisation credentials. In these embodiments, authorising the request for access may therefore comprise receiving the authorisation credentials. The at least one data management device processor 122 may, for example, receive the authorisation credentials via the data management device user interface (i.e. login details of the user). For example, the user may use a keyboard to provide the authorisation credentials to the data management device 118. Alternatively, the data management device 118, or an associated computing device may comprise a camera. The user may present one or more machine-readable indicia (such as a barcode, a QR code etc.), a representation of which may be captured by the camera. The sub-device 107 may comprise the machine-readable indicia (e.g. displayed on a display of the sub-device 107, engraved on a body of the sub-device 107, attached as a sticker etc.). The machine-readable indicia may comprise the authorisation credentials. Alternatively, the at least one data management device processor 122 may receive the authorisation credentials from another computing device.

[0186] The at least one data management device processor 122 generates an authorisation token based at least in part on the authorisation credentials. For example, the at least one data management device processor 122 may generate the authorisation token when the authorisation credentials are valid. The at least one data management device processor 122 may transmit the authorisation token to the service device 102. The service device 102 may be configured to authorise the sub-device 107 upon receipt of the authorisation token. In other words, by receiving the authorisation token, the service device 102 may be said to receive authorisation to enable access.

[0187] In some embodiments, the at least one data management device processor 122 may receive the authorisation credentials using the data management device network interface, via the communications network 120 (or the first communications network 120A, where relevant). The at least one data management device processor 122 may transmit the authorisation token to the service device 102 using the data management device network interface, via the communications network 120 (or the first communications network 120A, where relevant).

[0188] At 406, a private key and a corresponding public key are generated. The private key and/or the public key may be generated based at least in part on the sub-device information. In other words, a private/public key pair is generated based at least in part on the sub-device information. In some embodiments, the private key and/or the public key may be generated using another method. For example, the private key may be generated using a random number generator or a pseudo-random number generator.

[0189] In some embodiments, the at least one service device processor 104 generates the private key. The at least one service device processor 104 may generate the private key based at least in part on the sub-device information. The at least one service device processor 104 also generates the public key that is associated with the private key. In some embodiments, the at least one service device processor 104 executes the supervisor module 108 to generate the private key and/or the public key. The at least one service device processor 104 stores the private key and the public key in service device memory 106.

[0190] In some embodiments, the at least one data management device processor 122 generates the private key. The at least one data management device processor 122 may generate the private key based at least in part on the sub-device information. The at least one data management device processor 122 also generates the public key that is associated with the private key. The at least one data management device processor 122 stores the private key and the public key in data management device memory 124.

[0191] At 408, the public key is associated with the sub-device 107 in a sub-device record. In some embodiments, the at least one service device processor 104 associates the public key with the sub-device 107 in the sub-device record. The sub-device record may be stored in service device memory 106. In some embodiments, the at least one data management device processor 122 associates the public key with the sub-device 107 in the sub-device record. The sub-device record may be stored in data management device memory 124. In some embodiments, the at least one service device processor 104 executes the supervisor module 108 to associate the public key with the sub-device 107.

[0192] At 410, the public key and the private key are distributed between a number of the devices of the system 100. As previously described, in some embodiments, the at least one service device processor 104 may generate the private key and the public key. In these

embodiments, at 410, the at least one service device processor 104 transmits the public key to the relevant sub-device 107. In other words, the at least one service device processor 104 transmits the public key to the sub-device 107 that is associated with that public key (in the sub-device record). Each sub-device 107 is associated with its own respective public/private key pair. In some embodiments, the at least one service device processor 104 executes the supervisor module 108 to transmit the public key to the sub-device 107 that is associated with that public key.

[0193] The at least one service device processor 104 also transmits the public key to the data management device 118. Furthermore, the at least one service device processor 104 may transmit the private key to the data management device 118. Thus, the public key and the private key may be stored in the data management device memory 124. The at least one service device processor 104 may transmit the public key and/or the private key using the communications network 120, the first communications network and/or the relevant sub-device communications network 130B. In some embodiments, the at least one service device processor 104 executes the supervisor module 108 to transmit the public key and/or the private key.

[0194] The service device 102 may provide the private key to a containerised application 112 of the plurality of containerised applications 113. In particular, the at least one service device processor 104 may provide the private key to the containerised application 112. In some embodiments, at least one service device processor 104 provides the private key to containerised applications 112 of the plurality of containerised applications 113 that are authorised to process the sub-device data provided by the sub-device 107 that is associated with that private key. In some embodiments, the at least one service device processor 104 executes the supervisor module 108 to provide the private key to the containerised application 112.

[0195] This may comprise, for example, making the private key, which may be stored in service device memory 106, available to the containerised application 112. The at least one service device processor 104 selectively provides the private key to the containerised application 112. In particular, the at least one service device processor 104 may provide the private key only to containerised applications 112 of the plurality of containerised application 113 that are configured to process sub-device data provided by the sub-device 107 that is

associated with that private key. For example, where the sub-device data comprises temperature data, the at least one service device processor 104 may provide the private key associated with the relevant sub-device 107 only to containerised applications 112 that are configured to process temperature data.

[0196] The service device 102 may provide the public key to a containerised application 112 of the plurality of containerised applications 113. In particular, the at least one service device processor 104 may provide the public key to the containerised application 112. In some embodiments, at least one service device processor 104 provides the public key to containerised applications 112 of the plurality of containerised applications 113 that are authorised to process the sub-device data provided by the sub-device 107 that is associated with that public key. In some embodiments, the at least one service device processor 104 executes the supervisor module 108 to provide the public key to the containerised application 112.

[0197] This may comprise, for example, making the public key, which may be stored in service device memory 106, available to the containerised application 112. The at least one service device processor 104 selectively provides the public key to the containerised application 112. In particular, the at least one service device processor 104 may provide the public key only to containerised applications 112 of the plurality of containerised applications 113 that are configured to process sub-device data provided by the sub-device that is associated with that public key. For example, where the sub-device data comprises temperature data, the at least one service device processor 104 may provide the public key associated with the relevant sub-device 107 only to containerised applications 112 that are configured to process temperature data.

[0198] Therefore, at 410, the private key may be provided to the data management device 118, the public key may be provided to the data management device 118, the private key may be provided to a containerised application 112, the public key may be provided to a containerised application 112 and/or the public key may be provided to the relevant sub-device 107 of the plurality of sub-devices 109.

[0199] As previously described, the at least one data management device processor 122 may generate the private key and the public key. In these embodiments, the at least one data

management device processor 122 transmits the public key to the service device 102. Furthermore, the at least one data management device processor 122 may transmit the private key to the service device 102. Thus, the public key and the private key may be stored in the service device memory 106. In these embodiments, at 410, the at least one data management device processor 122 transmits the public key to the relevant sub-device 107 via the service device 102. In other words, the at least one data management device processor 122 transmits the public key to the sub-device 107 that is associated with that public key (in the sub-device record) via the service device 102. The at least one data management device processor 122 may transmit the public key and/or the private key using the communications network 120 or the first communications network 120A.

[0200] Therefore, at 410, the private key may be provided to the service device 102, the public key may be provided to the service device 102, the private key may be provided to a containerised application 112 as described previously (via the service device 102), the public key may be provided a containerised application 112 as describe previously (via the service device 102) and/or the public key may be provided to the relevant sub-device 107 of the plurality of sub-devices 109 as described previously (via the service device 102).

[0201] The at least one service device processor 104 may execute the sub-device data module 110 to perform one or more of steps 402, 404, 406, 408 and 410.

[0202] As previously described, each sub-device 107 is configured to generate sub-device data and to provide the sub-device data to the service device 102. At 412, the service device 102 receives sub-device data from a sub-device 107 of the plurality of sub-devices 109. In particular, the at least one service device processor 104 receives the sub-device data from the sub-device 107. The sub-device 107 may provide the sub-device data to the service device 102 using the communications network 120 or the relevant sub-device communications network 120B. As previously described, each sub-device 107 of the plurality of sub-devices 109 may comprise a sensor. Thus, the sub-device data may comprise sensor data.

[0203] At 414, the service device 102 encrypts the sub-device data, thereby generating encrypted sub-device data. That is, after receiving the sub-device data from the sub-device 107, the service device 102 encrypts the sub-device data. In particular, the at least one sub-device processor 104 encrypts the sub-device data using the public key. The private key

may be used to decrypt the encrypted sub-device data. In some embodiments, the at least one service device processor 104 executes the supervisor module 108 to encrypt the sub-device data.

[0204] In some embodiments, the sub-device 107 is configured to encrypt the sub-device data before providing (i.e. transmitting) the sub-device data to the service device 102. The sub-device 107 may encrypt the sub-device data using the public key. Again, the private key may be used to decrypt the encrypted sub-device data. In these embodiments, the service device 102 may not encrypt the sub-device data after receiving it, as it is already encrypted.

[0205] At 416, the service device 102 broadcasts the sub-device data to a multicast network channel of the multicast network 116 (or multicast bus 116A). In particular, the at least one service device processor 104 broadcasts the sub-device data to the multicast network channel. Broadcasting the sub-device data to the multicast network channel may comprise broadcasts the encrypted sub-device data. Broadcasting the sub-device data to the multicast network channel may comprise making the sub-device data (or the encrypted sub-device data) available on the network channel.

[0206] In some embodiments, broadcasting the sub-device data to the multicast network channel may be referred to as providing the sub-device data to the multicast network channel. In some embodiments, broadcasting the sub-device data to the multicast network channel may be referred to as transmitting the sub-device data to the multicast network channel. In some embodiments, broadcasting the sub-device data to the multicast network channel may be referred to as streaming the sub-device data to the multicast network channel. In some embodiments, broadcasting the sub-device data to the multicast network channel may be referred to as making the sub-device data available on the multicast network channel.

[0207] The multicast network channel is associated with the sub-device 107 that provided the sub-device data. In other words, the multicast channel may be considered to correspond to the sub-device 107 that provided the sub-device data. The multicast network channel is accessible by each containerised application 112 of the plurality of containerised applications 113. In some embodiments, the multicast network channel is a multicast Internet Protocol (IP) address or an application-layer multicast channel. In some embodiments, the at least one

service device processor 104 executes the supervisor module 108 to provide the sub-device data to the multicast network channel.

[0208] In a machine-to-machine (M2M) communication system such as that described herein, data may be obtained from any sub-device 107. The service device 102 loops through all installed containerised applications 112 to verify the sub-device data to determine if the sub-device data may be sent to an appropriate multicast channel. For a containerised application 112 with rights to receive the sub-device data, the sub-device data will be sent to the multicast channel corresponding to that containerised application 113.

[0209] As previously described, the private key may be used to decrypt the encrypted sub-device data. As the private key is only provided to the containerised applications 112 that are configured, or authorised to process the sub-device data provided by the sub-device 107 that is associated with the private key, only these containerised applications 112 can decrypt, and therefore process the sub-device data. In this way, the system 100 provides for improved security of the sub-device data before processing.

[0210] At 418, a containerised application 112 of the plurality of containerised applications 113 receives the sub-device data from the multicast network channel. In some embodiments, multiple containerised applications 112 of the plurality of containerised applications receive the sub-device data from the multicast network channel. Receiving the sub-device data from the multicast network channel comprises receiving the encrypted sub-device data from the multicast network channel. In particular, the at least one service device processor 104 executes the relevant containerised application(s) 112 to receive the sub-device data from the multicast network channel. In some embodiments, the at least one service device processor 104 executes the sub-device data module 110 to receive the sub-device data from the multicast network channel.

[0211] At 420, the containerised application(s) 112 that receive the sub-device data from the multicast network channel and that have been provided the private key, decrypt the sub-device data. In particular, the at least one service device processor 104 executes the relevant containerised application(s) 112 to decrypt the encrypted sub-device data using the private key. The at least one service device processor 104 executes the relevant containerised application(s) 112 to decrypt the encrypted sub-device data prior to processing the sub-device

data. In some embodiments, the at least one service device processor 104 executes the sub-device data module 110 to decrypt the sub-device data.

[0212] At 422, the containerised application(s) 112 process the sub-device data. In particular, the at least one service device processor 104 executes the relevant containerised application(s) 112 to process the sub-device data. In this way, each of the containerised application(s) 112 that are associated with the sub-device 107 can process the sub-device data for different purposes simultaneously. By executing the relevant containerised application(s) 112 to process the sub-device data, the at least one service device processor 104 generates processed data. Processing the sub-device data may comprise determining one or more sub-device data metrics based at least in part on the sub-device data. For example, where the sub-device data comprises a time-series of temperature values that are indicative of a temperature of an environment of the relevant sub-device 107, processing the sub-device data may comprise determining one or more of a maximum temperature, a minimum temperature or an average temperature over a time window of the time-series. The maximum temperature, minimum temperature and/or average temperature may be considered sub-device data metrics, for example. In some embodiments, the at least one service device processor 104 executes the sub-device data module 110 to process the sub-device data.

[0213] A particular containerised application 112 of the plurality of the containerised applications 113 may be configured to receive sub-device data from a plurality of the multicast network channels simultaneously. In such a case, the at least one service device processor 104 executes the relevant containerised application(s) 112 to process the sub-device data. Thus, the at least one service device processor 104 executes the relevant containerised application(s) 112 to process sub-device data that has been provided by a plurality of sub-devices 107.

[0214] In some embodiments, the at least one service device processor 104 is configured to store the sub-device data or the encrypted sub-device data in service device memory 106. In some embodiments, the at least one service device processor 104 is configured to store the processed data in service device memory 106. In some embodiments, the at least one service device processor 104 is configured to store the one or more sub-device data metrics in service device memory 106. In some embodiments, the at least one service device processor 104

executes the sub-device data module 110 to store the sub-device data, the encrypted sub-device data and/or the one or more sub-device data metrics in service device memory 106.

[0215] At 424, the at least one service device processor 104 encrypts the processed data. In particular, the at least one service device processor 104 encrypts the processed data using a public key, thereby generating encrypted processed data. The public key may be the previously described public key. Alternatively, the public key may be another public key. For example, the at least one service device processor 104 may execute the relevant containerised application 112 to encrypt the processed data.

[0216] At 426, the service device 102 transmits the processed data to the data management device 118. This may comprise transmitting the one or more sub-device data metrics to the data management device 118. In particular, the at least one service device processor 104 transmits the processed data to the data management device 118. The at least one service device processor 104 may transmit the processed data to the data management device 118 using the service device network interface and/or the communications network 120 (or first communications network 120A, where applicable). In some embodiments, the at least one service device processor 104 executes the sub-device data module 110 to transmit the processed data to the data management device 118. Transmitting the processed data may comprise transmitting encrypted processed data.

[0217] The at least one service device processor 104 may execute the supervisor module 108 to perform one or more of steps 412, 414, 416, 418, 420, 422, 424 and 426.

[0218] At 428, the data management device 118 receives the processed data and stores the processed data. In particular, the at least one data management processor 122 stores the processed data in data management memory 124. As the processed data may be encrypted before being transmitted by the service device 102, storing the processed data may comprise storing the encrypted processed data. Alternatively, the at least one data management processor 122 may decrypt the encrypted processed data. The at least one data management processor 122 may decrypt the encrypted processed data using a private key. The private key may be the previously described private key. Alternatively, the private key may be another public key. The at least one data management processor 122 may then store the decrypted processed data in data management memory 124.

[0219] The present disclosure provides a number of advantages.

[0220] The present disclosure provides an application marketplace for containerised applications which enables selective upload, purchase and deployment of applications, enabling selected applications to be processed and interact with received data.

[0221] The present disclosure provides a central access management control system, which enables selective exchange of data between a containerised data centre and multiple sub-devices. This provides control over the access for data to be processed by user devices, thereby enabling the selective processing of data to be processed over secure communications.

[0222] The present disclosure provides secure and reliable transmission for multicasting services, which advantageously enhances information security in Internet of Things (IoT) systems when multiple physical sub-devices (e.g. sensors, machines, facilities, etc.) are employed and synchronised.

[0223] Advantageously, the system 100 provides for central management of access to containerised applications provided by an operator of the system by selectively providing an access key (e.g. the private key) to one or more authorised users.

[0224] It will be appreciated by persons skilled in the art that numerous variations and/or modifications may be made to the above-described embodiments, without departing from the broad general scope of the present disclosure. The present embodiments are, therefore, to be considered in all respects as illustrative and not restrictive.

[0225] It is to be understood that the above embodiments have been provided only by way of exemplification of this disclosure, and that further modifications and improvements thereto, as would be apparent to persons skilled in the relevant art, are deemed to fall within the broad scope and ambit of the present disclosure described herein. It is further to be understood that features from one or more of the described embodiments may be combined to form further embodiments.

CLAIMS:

1. A computer-implemented method for selectively processing data, the computer-implemented method comprising:
 - receiving, by a service device, a request for access from a sub-device, the request for access comprising sub-device information associated with the sub-device;
 - receiving, by the service device, authorisation to enable access, the authorisation being based at least in part on the sub-device information;
 - receiving, by the service device, sub-device data from the sub-device;
 - broadcasting, by the service device, the sub-device data to a corresponding multicast network channel that is accessible by a plurality of containerised applications of the service device;
 - receiving, by a containerised application of the plurality of containerised applications, the sub-device data from the multicast network channel;
 - processing, by the containerised application, the sub-device data to generate processed data; and
 - transmitting, by the service device and to a data management device, the processed data.
2. The computer-implemented method of claim 1, further comprising:
 - receiving, by a user interface of the data management device, authorisation credentials;
 - generating, by the data management device, an authorisation token based at least in part on the authorisation credentials; and
 - transmitting the authorisation token to the service device.
3. The computer-implemented method of claim 1 or claim 2, further comprising transmitting, by the service device, the request for access to the data management device.
4. The computer-implemented method of claim 3 when dependent on claim 2, wherein receiving authorisation to enable access comprises receiving the authorisation token.
5. The computer-implemented method of any one of claims 1 to 4, further comprising storing, by the data management device, the processed data.

6. The computer-implemented method of any one of claims 1 to 5, wherein the sub-device information comprises one or more of a sub-device serial number, a sub-device Media Access Control (MAC) address, a sub-device model number, a sub-device identifier and a random key.
7. The computer-implemented method of any one of claims 1 to 6, wherein the sub-device comprises a sensor and the sub-device data comprises sensor data.
8. The computer-implemented method of any one of claims 1 to 7, wherein the multicast network channel is a multicast Internet Protocol (IP) address or an application-layer multicast channel.
9. The computer-implemented method of any one of claims 1 to 8, wherein processing the sub-device data comprises determining one or more sub-device data metrics based at least in part on the sub-device data.
10. The computer-implemented method of any one of claims 1 to 9, further comprising:
 - generating a private key;
 - generating a public key that is associated with the private key; and
 - associating, in a sub-device record, the public key with the sub-device.
11. The computer-implemented method of claim 10, wherein the private key and the public key are generated by the service device; and
 - wherein the computer-implemented method further comprises one or more of:
 - providing the private key to the data management device;
 - providing the public key to the data management device;
 - providing the private key to the at least one containerised application;
 - providing the public key to the at least one containerised application; and/or
 - providing the public key to the sub-device.
12. The computer-implemented method of claim 10, wherein the private key and the public key are generated by the data management device; and
 - wherein the computer-implemented method further comprises one or more of:

providing the private key to the service device;
providing the public key to the service device;
providing the private key to the at least one containerised application;
providing the public key to the at least one containerised application; and/or
providing the public key to the sub-device.

13. The computer-implemented method of any one of claims 10 to 12, further comprising encrypting, by the service device and using the public key, the received sub-device data prior to broadcasting the sub-device data to the multicast network channel, thereby generating encrypted sub-device data.

14. The computer-implemented method of claim 13, wherein:
broadcasting the sub-device data to the multicast network channel comprises broadcasting the encrypted sub-device data to the multicast network channel; and
receiving the sub-device data from the multicast network channel comprises receiving the encrypted sub-device data.

15. The computer-implemented method of claim 14, further comprising decrypting, by the containerised application and using the private key, the encrypted sub-device data received from the multicast network channel, prior to processing the sub-device data.

16. The computer-implemented method of any one of claims 10 to 15, further comprising encrypting, by the containerised application and using the public key, the processed data to generate encrypted processed data.

17. The computer-implemented method of claim 16 when dependent on claim 5, wherein:
transmitting the processed data comprises transmitting the encrypted processed data;
and
storing the processed data comprises storing the encrypted processed data.

18. The computer-implemented method of any one of claims 1 to 17, further comprising:
transmitting, by the service device and to the data management device, a request to initialise a new containerised application;

retrieving, by the service device, the new containerised application;
initialising, by the service device, the new containerised application; and
transmitting, by the service device, an installation confirmation indicating a
successful initialisation of the containerised application to the data management device;
wherein the containerised application of the plurality of containerised applications is
the new containerised application.

19. The computer-implemented method of claim 18, further comprising:
receiving, by the data management device, the request to initialise the new
containerised application; and
transmitting, by the data management device, a license key to the service device, the
license key being associated with the containerised application.

20. A data management system for selectively processing data, the data management
system comprising:
a service device comprising:
at least one service device processor; and
service device memory storing:
a plurality of containerised applications executable by the at least one
service device processor; and
program instructions accessible by the at least one service device
processor, and configured to cause the at least one service device processor to:
receive a request for access from a sub-device, the request for
access comprising sub-device information associated with the sub-device;
receive authorisation to enable access, the authorisation being
based at least in part on the sub-device information;
receive sub-device data from the sub-device;
broadcast the sub-device data to a corresponding multicast network
channel that is accessible by the plurality of containerised applications;
execute a containerised application of the plurality of containerised
applications, the containerised application being configured to:
receive the sub-device data from the multicast network
channel; and
process the sub-device data to generate processed data; and

transmit the processed data to a data management device.

21. The data management system of claim 20, further comprising the data management device, the data management device comprising:

at least one data management device processor; and

data management device memory storing data management program

instructions accessible by the at least one data management device processor, and configured to cause the at least one data management device processor to:

receive the processed data.

22. The data management system of claim 21, wherein the data management program instructions are further configured to cause the at least one data management device processor to:

receive authorisation credentials;

generate an authorisation token based at least in part on the authorisation credentials;

and

transmit the authorisation token to the service device, thereby providing authorisation to enable access.

23. The data management system of any one of claims 20 to 22, wherein the service device program instructions are further configured to cause the at least one service device processor to transmit the request for access to the data management device.

24. The data management device of claim 23 when dependent on claim 22, wherein the service device program instructions are further configured to cause the at least one service device processor to receive the authorisation token, thereby receiving authorisation to enable access.

25. The data management system of claim 21 or any one of claims 22 to 24 when dependent on claim 21, wherein the data management program instructions are further configured to cause the at least one data management device processor to store the processed data.

26. The data management system of any one of claims 20 to 25, wherein the sub-device information comprises one or more of a sub-device serial number, a sub-device Media Access

Control (MAC) address, a sub-device model number, a sub-device identifier and a random key.

27. The data management system of any one of claims 20 to 26, wherein the sub-device comprises a sensor and the sub-device data comprises sensor data.

28. The data management system of any one of claims 20 to 27, wherein the multicast network channel is a multicast Internet Protocol (IP) address or an application-layer multicast channel.

29. The data management system of any one of claims 20 to 28, wherein processing the sub-device data comprises determining one or more sub-device data metrics based at least in part on the sub-device data.

30. The data management system of any one of claims 20 to 29, wherein the service device program instructions are further configured to cause the at least one service device processor to:

- generate a private key based at least in part on the sub-device information;
- generate a public key that is associated with the private key; and
- associate, in a sub-device record, the public key with the sub-device.

31. The data management system of claim 30, wherein the service device program instructions are further configured to cause the at least one service device processor to:

- provide the private key to the data management device;
- provide the public key to the data management device;
- provide the private key to the at least one containerised application;
- provide the public key to the at least one containerised application; and/or
- provide the public key to the sub-device.

32. The data management system of claim 21 or any one of claims 22 to 31 when dependent on claim 21, wherein the data management program instructions are further configured to cause the at least one data management device processor to:

- generate a private key based at least in part on the sub-device information;
- generate a public key that is associated with the private key; and

associate, in a sub-device record, the public key with the sub-device.

33. The data management system of claim 32, wherein the data management program instructions are further configured to cause the at least one data management device processor to:

- provide the private key to the service device;
- provide the public key to the service device;
- provide the private key to the at least one containerised application;
- provide the public key to the at least one containerised application; and/or
- provide the public key to the sub-device.

34. The data management system of any one of claims 30 to 33, wherein the service device program instructions are further configured to cause the at least one service device processor to encrypt the received sub-device data using the public key, prior to broadcasting the sub-device data to the multicast network channel, thereby generating encrypted sub-device data.

35. The data management system of claim 33, wherein:

- broadcasting the sub-device data to the multicast network channel comprises broadcasting the encrypted sub-device data to the multicast network channel; and
- receiving the sub-device data from the multicast network channel comprises receiving the encrypted sub-device data.

36. The data management system of claim 35, wherein the program instructions are further configured to cause the at least one service device processor to execute the containerised application of the plurality of containerised applications, the containerised application being configured to decrypt the encrypted sub-device data received from the multicast network channel using the private key, prior to processing the sub-device data.

37. The data management system of any one of claims 30 to 36, wherein the program instructions are further configured to cause the at least one service device processor to execute the containerised application of the plurality of containerised applications, the containerised application being configured to encrypt the processed data to generate encrypted processed data.

38. The data management system of claim 37 when dependent on claim 25, wherein transmitting the processed data comprises transmitting the encrypted processed data; and storing the processed data comprises storing the encrypted processed data.

39. The data management system of any one of claims 20 to 38, wherein the program instructions are further configured to cause the at least one service device processor to:

- transmit a request to initialise a new containerised application to the data management device;

- retrieve the new containerised application;

- initialise the new containerised application; and

- transmit an installation confirmation indicating successful initialisation of the containerised application to the data management device;

wherein the containerised application of the plurality of containerised applications is the new containerised application.

40. The data management system of claim 39 when dependent on claim 21, wherein the data management program instructions are further configured to cause the at least one data management device processor to:

- receive the request to initialise the new containerised application; and

- transmit a licence key to the service device, the license key being associated with the containerised application.

41. A computer-implemented method for securely initialising a containerised application, the computer-implemented method comprising:

- receiving, by a data management device, a request from a service device to initialise the containerised application;

- transmitting, by the data management device, a license key to the service device, the license key being associated with the containerised application;

- retrieving, by the service device, the containerised application;

- initialising, by the service device, the containerised application using the license key;

and

- transmitting, by the service device, an installation confirmation indicating a successful initialisation of the containerised application to the data management device.

42. The computer-implemented method of claim 41, further comprising executing, by the service device, the containerised application.

43. The computer-implemented method of claim 41 or claim 42, further comprising transmitting, by the service device, the request to initialise the containerised application to the data management device in response to the service device receiving a request for access from a sub-device that is associated with the containerised application.

44. The computer-implemented method of any one of claims 41 to 43, further comprising:
transmitting, by the service device, the license key to a license validation service; and
receiving, by the service device, a license validation indicating that the license key is valid.

45. The computer-implemented method of any one of claims 41 to 44, wherein receiving the containerised application comprises:
downloading the containerised application from a software repository, over a network; and
installing the containerised application.

46. A data management system for securely initialising a containerised application, the data management system comprising:
a service device comprising:
at least one service device processor; and
service device memory storing program instructions accessible by the at least one service device processor, and configured to cause the at least one service device processor to:
transmit a request to initialise a containerised application to a data management device;
retrieve the containerised application;
initialise the containerised application; and
transmit an installation confirmation indicating a successful initialisation of the containerised application to the data management device; and

the data management device, wherein the data management device comprises:
at least one data management device processor; and
data management device memory storing data management program instructions accessible by the at least one data management device processor, and configured to cause the at least one data management device processor to:
receive the request to initialise the containerised application; and
transmit a licence key to the service device, the license key being associated with the containerised application.

47. The data management system of claim 46, wherein the program instructions are further configured to cause the at least one service device processor to execute the containerised application.

48. The data management system of claim 46 or claim 47, wherein the program instructions are further configured to cause the at least one service device processor to transmit the request to initialise the containerised application to the data management device in response to the service device receiving a request for access from a sub-device that is associated with the containerised application.

49. The data management device of any one of claims 46 to 48, wherein the program instructions are further configured to cause the at least one service device processor to:
transmit the license key to a license validation service; and
receive a license validation indicating that the license key is valid.

50. The data management device of any one of claims 46 to 49, wherein receiving the containerised application comprises:
downloading the containerised application from a software repository, over a network; and
installing the containerised application.