INTEGRATION PLATFORM TO ENABLE OPERATIONAL INTELLIGENCE AND USER JOURNEYS FOR SMART CITIES AND THE INTERNET OF THINGS

BACKGROUND OF THE INVENTION

FIELD OF THE INVENTION

[0001] This invention relates to the field of smart infrastructure management – also known as Smart City solutions.

DESCRIPTION OF THE BACKGROUND

[0002] The field of infrastructure management has historically been limited to "silo-based" applications and operations, in which data captured through devices and human data entry stays in the applications in which it was captured or entered. Where integration across systems is required, custom solutions are made by solution integrators. These custom solutions are typically designed to serve trained operators, such as facilities managers or building engineers.

SUMMARY OF THE INVENTION

[0003] The IoT and the IoE

[0001] Since the advent of the internet, city system infrastructures have grown in sophistication and variety. During last decade, many internet-based devices have made their way to marketplace – ushering in the era of the Internet of Things (IoT), and building, building complex, neighborhood, town and city infrastructures are evolving from 'systems and devices' to the "Internet of Everything" (IoE) –which includes every device, system, network, application, cloud service, and even person, that is connected to the internet. Indeed, any person with a smartphone is now a node of the IoE. In order to be effective and efficient, system integration must no longer be merely between 'systems'-- it must include IoE.

[0002] Smart City – Smart management of infrastructure

[0003] In short, the widely growing adoption of digital technologies to manage cities, with the goal of enhancing performance of city operations, reducing cost and improving service to citizens (end-users of systems), demands a sophisticated approach to infrastructure management. These infrastructures include both physical systems such as energy and water supply, transportation, etc., and software systems such as supervisory and data acquisition (SCADA), maintenance management, lease management, etc.

[0004] The complexity of smart city implementation comes from two fundamental challenges: first, a smart city according to the invention is a 'living system' which must constantly evolve to serve city needs and to reflect constantly changing and moving citizenry/population; and secondly, people and skills involved in modeling city services are different from people and skills involved in system implementation and integration. That is, the smart city solution of the invention must be highly intuitive, as mobile users have a very short operation/attention timespan -- they expect a system to provide ONLY what they need to carry out their operation.

[0005] The present smart city solution addresses this by modeling 'user journeys'– providing a template containing the anticipated exact steps (interactions with the system of the invention) that a particular type of user will require or carry out achieve an objective.

[0006] Solution – A metadata model of the city's IoE

[0007] The present invention solves the problem by implementing a "metadata model of a city's IoE," and using the metadata model of the city's IoE to implement citizen engagement through 'user journeys' and system intelligence through automated response logic. More specifically, the present invention provides a framework that allows system integrators to build a metadata model of a city's IoE, which metadata model then enables city managers to implement user journeys and system intelligence, in incremental fashion, as the business needs of the city evolves.

[0008] The following is a summary of some of the features of the present invention:

a) The invention extracts metadata from the Internet of Everything, i.e., networks of computers, subsystems, applications, devices, people (their smartphones) and cloud services, including data and methods. Each of these items, and any person with a smartphone, is a node of the IoE.

b) The invention sets up relationships between individual metadata elements to form a metadata model for the city;

c) The invention processes events, and applies analytics to filter and qualify events – this steps is sometimes referred herein as operational intelligence;

d) The invention sets up automated sequences of actions to 'self manage' known situations;

e) The invention sets up 'user journeys'; and

f) The invention connects with end-users (citizens) through unified communication.

[0009] Example

[0010] A metadata model of a smart city having been built using the present invention, including all the devices, systems, subsystems, devices, networks, applications, data, etc., that are connected to the internet, the system receives metadata from a building control system reflecting a room temperature of 26 degrees, within normal parameters. However, the system of the present invention is monitoring metadata related to user-devices (mobile phones) of building occupants [metadata model] who are sending out Tweets indicating 'office is hot.' The system correlates the control system metadata concerning room temperature with metadata from other systems, i.e., mobile phones on the IoE [event processing & analytics], and the system qualifies the

situation as “important to respond” and sends an SMS [unified communication] to the facilities manager of the building indicating that building occupants feel uncomfortably warm. The facilities manager opens his mobile app, and it will show a single page with the details of the problem, and also what system thinks the root cause is --- for example, perhaps an “energy saving” mode has automatically kicked in due to a sudden rise of dynamic energy price. The same mobile app page can give the facilities manager the option to turn off the energy saving mode, and will show how much extra it will cost the company, without the facilities manager having to navigate to various systems to gather such knowledge. This is an example of modeling a user (in this case, a facilities manager) journey.

[0011] Accordingly, there is provided according to the invention the following features:

a) Devices (generating information, and acting based on received information), applications (holding information and processing information) and people (receiving information and acting based on them) are part of a single metadata model.

b) Building the model automatically (based on events) and continuously & dynamically modifying the metadata model as city elements change.

c) Use metadata relationships to extract metadata from existing nodes or from subsystems (using connectors) to build & update metadata nodes.

d) Metadata model being ‘machine readable’ due to predefined templates defining well-known metadata structures.

e) Cross reference between existing metadata nodes (enabling progressive implementation).

f) Enabling/augmenting operational intelligence using metadata model.

g) Citizen engagement through ‘user journeys’ (a sequence of operations that are specifically designed to achieve an objective) rather than merely enabling access to a bunch of applications

h) Applying such solution to ‘city wide infrastructure.’

[0012] Accordingly, there is provided according to the invention a system for integrated management of the infrastructure of a community, where the infrastructure includes a plurality of network-connected infrastructure systems, subsystems, devices, and applications, and where the systems, subsystems, devices and applications are independent and unintegrated, the system including:

- a user-defined machine-readable metadata model comprising predefined metadata node templates defining metadata structures for said infrastructure systems, subsystems, devices, and applications;
- connector modules configured to collect metadata from said infrastructure systems, subsystems, devices, and applications;
- an event engine that receives metadata from said connector modules and which populates metadata node templates using said metadata to create , populated metadata nodes;
- a dynamic metadata map/model assembled by said event engine, comprising populated metadata nodes, including relationships between nodes based on said collected metadata, and
- pre-defined operation sequences, based on the occurrence of events received from the connector modules, which pre-defined operation sequences define a set of operations according to predicted end-user requirements in response to pre-selected events.

[0013] There is further provided according to an embodiment of the invention a system wherein the community is a city.

[0014] There is further provided according to an embodiment of the invention a system wherein the community is a single building.

[0015] There is further provided according to an embodiment of the invention a system wherein the community is a collection of buildings.

[0016] There is further provided according to an embodiment of the invention a system wherein the community is a workplace or single-tenant space within a multi-tenant building where some subsystems (such as air conditioning) belong or are operated by the building and/or building management, while other subsystems (such as card access & CCTV) are owned and/or managed by the tenant.

[0017] There is further provided according to an embodiment of the invention including cross references between existing metadata nodes.

[0018] There is further provided according to an embodiment of the invention a system wherein the event engine is configured to build the dynamic metadata map automatically, continuously, and dynamically, modifying the metadata map as collected metadata changes.

[0019] There is further provided according to an embodiment of the invention a system wherein the event engine is configured to use metadata relationships to extract metadata from existing nodes or from subsystems to build and update metadata nodes.

[0020] There is further provided according to an embodiment of the invention a system wherein the dynamic metadata map/model contains metadata notes for a plurality of infrastructure systems, subsystems, devices, and/or applications that are involved in a service workflow,

wherein a service workflow comprises a set of pre-selected information that presented to an end user and said predefined operation sequences.

[0021] There is further provided according to an embodiment of the invention a system wherein the predefined operation sequences comprise an end-to-end service workflow for end-users of the system by having metadata model undertake integration with multiple subsystems involved in the service workflow.

[0022] There is further provided according to an embodiment of the invention a system further including an analytics engine configured to identify and correlate events and to update existing events or create new events based on identified correlations.

[0023] There is further provided according to an embodiment of the invention a system wherein the event engine is configured to send commands to one or more of said infrastructure systems, subsystems, devices, and applications based on the receipt of one or more event conditions received by the event engine.

[0024] There is further provided according to an embodiment of the invention a system wherein the network comprises the Internet.

[0025] There is further provided according to an embodiment of the invention a system wherein the network comprises only private networks.

BRIEF DESCRIPTION OF THE DRAWINGS

[0026] Figure 1 shows how a user (e.g., a system integrator) has defined event processing logic in the metadata node template.

[0027] Figure 2 shows a metadata model editor which a user can access to build a metadata template according to an embodiment of the invention.

[0028] Figure 3 is another view of a metadata model editor which a user can access to build a metadata template according to an embodiment of the invention.

[0029] Figure 4 is a diagram showing a metadata model according to an embodiment of the invention.

[0030] Figure 5 is a representation of the extraction of metadata from subsystems to populate a metadata model according to an embodiment of the invention.

[0031] Figure 6 is a representation of how metadata node relationships can be established using common metadata present in different metadata nodes.

[0032] Figure 7 is a representation of how an embodiment of the invention reacts to new devices entering or interacting with the "Smart City."

[0033] Figure 8 is a representation of different parts of the invention and their relationships.

DETAILED DESCRIPTION OF THE INVENTION

[0034] Dynamic metadata map

[0035] The present invention may be used in any community (workplace, building, building complex, neighborhood, town, or city) that might benefit from the integrated management of infrastructure systems, the use or operation of which impacts or is impacted by devices, systems, networks, applications, and persons (smartphones) in the IoE. "Workplace" as used herein refers to a single-tenant space within a multi-tenant building where some subsystems (such as air conditioning) belong or are operated by the building and/or building management, while other subsystems (such as card access & CCTV) are owned and/or managed by the tenant. However, the invention is most useful in the case of a city wide IoE or other large collection of devices, systems, and networks, the members of which (in particular the people using smartphones) are constantly in flux, moving within the system, and moving into and out of the system. Using the

case of a Smart City as an example, the system of the invention includes a metadata map or "model" that is made up of metadata nodes that represent the IoE -- each internet-aware device/entity in the city. Since this comprehensive metadata model includes people with smartphones who are constantly moving around and going in and out of the city, also referred to herein as the solution boundary, the metadata model is adapted to be constantly, dynamically, and automatically changing as the available metadata changes. According to a preferred embodiment, there will be only one metadata map for the entire city with millions of metadata nodes connected to each other directly or indirectly.

[0036] Metadata nodes

[0037] The metadata model is made up of metadata nodes. Each node is a collection of metadata. For example, 'Meeting Room One' is a Facility-type metadata node, and it carries metadata such as 'Location', 'Seating Capacity', 'Amenities', etc. Other types of metadata nodes include: Visitor, Staff, Equipment, Work Order, and Room Booking. In the case of Work Order and Room Booking types of metadata nodes, these metadata nodes represent transactions that are dynamically introduced while system is in operation. Further, a Room Booking metadata node may contain metadata of Facility (meeting room), Staff (host) and Visitor (attendee). Visitor-type metadata nodes might be identified by a telephone number or an email address; in this way, a visitor's mobile phone constitutes a physical representation of the visitor, which in turn is represented in the invention as a metadata node on the system with, for example, the mobile phone's telephone number or email address(es). Likewise, a Staff-type metadata node might also be identified by a telephone number, email address, name, organization, department and/or work location. An Equipment-type metadata node might include Asset-ID, Asset Category, Installed Location, Serving Locations (where it serves, which may be different from installed

location), Assignee (User), Related Equipment, Make, Model and/or Serial number. A Work Order type metadata node might include Work Order ID, Service Category, Location, Work Description, Assigned Vendor/Technician, Supervisor and/or Deadline to Complete.

[0038] Events and event engine

[0039] Constructing and modifying of metadata nodes is done by an 'event engine'. It receives metadata in the form of 'events'. Events are generated by a Connector module, described below. An exemplary event might be 'Visitor V001 arrived at Gate G001'. Upon receiving new or modified metadata, the event engine either creates new metadata nodes or modifies existing nodes to absorb newly received information into metadata model. In another example, a staff member in the office may be represented by a Staff metadata node carrying two metadata (among others): Staff ID and Current Location.

[0040] Event filtering and processing is also done by the event engine, based on the logic defined in metadata model template. The metadata model is configured to allow a user to establish an 'action sequence' that defines how engine will process incoming events.

[0041] Figure 1 shows how a user (e.g., a system integrator) has defined event processing logic in the metadata node template, which will be followed by the event engine. When Staff Arrived event is received, it has two metadata: Staff ID (SID) and Gate (through which Staff member arrived, and where event was captured). Event is compared against Staff ID to determine if the event belongs to this Staff member. If not, Event is ignored. Gate parameter that is associated with the Event is used to find the Location it belongs to, then assign that Location as Current Location metadata of Staff member. This forms a new connection between Staff node and Location node. As Staff moves around the building, more such events will arise, and Staff's Current Location will be dynamically updated.

[0042] Metadata node template

[0043] A central feature of the invention is a metadata node template, which is defined by the user (e.g., system integrator) and which defines how a metadata node should represent an entity, how metadata nodes should be connected to one-another. A metadata node template is a passive element that describes the contents of a node, which contents constitute instructions to the event-engine for assembly of the nodes and the relationships between them (the metadata nodes and relationships together constituting the metadata model/map). All instructions to the event engine go into the template which is editable by system integrator to modify behavior of the entire system.

[0044] The user builds the metadata template using a metadata model editor, see Figures 2 and 3. The metadata model editor walks a user through the selection of various types of metadata nodes, the metadata they include, and actions to take as events are collected and processed by the event engine. For example, the 'Visitor in Building' node template would identify 'Arrived Gate', 'Meeting Location' and 'Host' as metadata placeholders. Upon receiving 'visitor arrived' event, new 'Visitor in Building' node would be created by event engine to represent Visitor V001.

[0045] Forming metadata model

[0046] Referring to Figure 4, the 'metadata model' is formed by connecting metadata nodes through common metadata that exists in different nodes. This is done by the event engines as part of the processing of received events. In the above example, upon receiving 'Visitor V001 arrived at gate G001' event, two embedded metadata (visitor's ID and gate's ID) are extracted and stored in metadata placeholders in the node. Further, the metadata node template (Visitor in Building) also requests the event engine to identify 'meeting location' and 'host', to complete its

structure. The event engine explores existing metadata models to find matching nodes and to extract missing metadata to complete the node as specified by node template. Once a metadata node is made (by the event engine), only ‘metadata elements’ in it can be modified in order to store information (such as Staff->Name) or to make a connection to another metadata node (such as Staff->CurrentLocation). Action sequences defined in the template are embedded in the node (like DNA), but not individually alterable, except using the metadata model editor. That is, action sequence portions of the metadata node do not change based on receipt or processing of metadata.

[0047] Thus, the Metadata map or model is the mesh or network of metadata nodes (built using metadata organized according to metadata note templates), which nodes are “connected” using the relationships that are reflected in the metadata. For example, Staff-JohnSmith node forms a metadata map (on a microscale) with Building-125IndustrialDrive node when CurrentLocation metadata (of Staff-JohnSmith) carries the reference to Building-125IndustrialDrive. According to a preferred embodiment of the invention, the metadata map encompasses an entire city, with millions of metadata nodes constructed using collected metadata and assembled using metadata node templates, and the relationships between which are are also defined by common metadata elements between metadata nodes.

[0048] Extraction of metadata from subsystems

[0049] Referring to Figure 5, the collection and/or extraction of metadata from systems, subsystems, networks, devices, applications and other elements of the IofE requires identifying data elements and methods of the silo-system. A “Connector” or “a software executable, identifies metadata from the various internet-aware elements of the IofE and sends them to the ‘event engine’ as ‘events’. According to an embodiment of the invention, there is a specific

"Connector" for each subsystem/application that the invention collects information from so with Connector can effectively communicate with the target/corresponding/associated subsystem/application. The Connectors then transform the events into the format that event-engine can understand and send it to the event engine.

[0050] Alternatively, certain subsystems may be programmed to 'push' their metadata to the event engine directly, for example using their own Connectors configured to communicate with the system of the invention.

[0051] Examples of subsystems according to the invention include: air conditioning and heating systems, elevator systems, card access systems, CCTV systems and BMS (building management systems). Applications are addressed in the same way as any other subsystem. Mobile devices (such as a temperature sensor inside transportable refrigerator) report their status to a central server, and a Connector specific to that device's system will pick it up from the central server. With respect to IoT-devices, they carry an IP-address that allows a Connector to directly address them. Smartphones and Networks are not considered a subsystem, although, a smartphone could represent a device or system through a connector (app) installed in the smartphone – for example, a connector placed in the smartphone could detect the proximity of iBeacon thereby determine current location of the Visitor and notify event engine.

[0052] Correlation of events

[0053] In the above example, the Connector may receive information reflecting that Visitor V001 is leaving the building, i.e., Visitor V001's entry/exit card was presented at a security device-monitored exit point. At that point, Connector will raise 'Visitor V001 left' event. The event engine will correlate this event to existing 'Visitor In Building' node for V001 and make the necessary updates to reflect the current status.

[0054] Navigation through metadata model

[0055] Referring to Figure 6, the metadata model (collection of nodes) behaves similar to an 'object collection' in object oriented programming. That is, it forms a hierarchy (such as parent--child relationship in Locations), and it carries object attributes (such as node's attributes representing metadata) and methods (actions represented by metadata).

[0056] For example, Location metadata ('Room 001') of equipment ('A001') gets attached to Location ('Room 001') attribute (metadata). Location hierarchy makes it immediately inherit parent location ('Floor 001') of Room 001. In this way, for example, any service that is looking for 'photocopy machines' (equipment of category 'photocopy') at Location 'Floor 001' will now find A001. The metadata model is a machine readable structure that does not require a human to browse through it.

[0057] Cross references between metadata nodes

[0058] The invention allows system implementation to take place progressively. When the system (the metadata map) is expanded by adding new metadata node templates and connectors, the system of the invention generates/builds cross references between new nodes and existing nodes, without having to modify existing nodes. Referring, for example, to Figure 7, originally implemented 'people count' map node (representing floor F001) maintains a list of employees in F001, collected from Card Access system, which is required in case of emergency evacuation. However, people count node is initially not aware of newly introduced event 'Visitor001 arrived', collected from QR Code scanner. Therefore, when a new visitor arrives, it doesn't reflect on people count. The invention addresses this by having newly introduced Visitor node raise a 'secondary' event, 'Person V001 entered F001', which is recognized by original people count node (without having to modify it).

[0059] Consolidated actions – orchestrations

[0060] As explained in the example above, 'people count' node has a list of people (both employees and visitors) who are currently in floor F001. The present invention allows a system integrator to define a single action for an event associated with a metadata node, e.g., "notify emergency evacuation" to everyone in F001 in the event that metadata is received reflecting an emergency situation, for example, a terrorist threat, fire or smoke, regardless of the silo that captured their presence on floor F001, whether it was Room Booking systems managing visitors, Work Order system managing contractors, or Space Management system managing seating locations for internal staff, and/or regardless of what silo captured the existence of an event, whether it was twitter feed from personal devices, email applications, a building/workplace fire detection/suppression system, and/or any other system or application.

[0061] Automated Orchestrations

[0062] In some cases, it is required to automatically execute a series of actions, upon receiving an event. For example, People Count node may also be programmed to send SMS notification to all 'persons' in the particular floor when evacuation is ordered. More importantly, additional metadata that would identify a person as 'handicapped' could enable people count node to send a notification to facilities managers about locations of handicap persons in the building with instructions to provide special handling and assistance during the evacuation.

[0063] Complex event processing (analytics)

[0064] The invention includes an analytic engine that is able to process events and event---data to identify patterns and correlations. Examples are as follows:

a) Generate extended event based on repeated basic events within a given timespan (repeated "card rejected" event may generate "card reader faulty" event)

b) Correlate repeated events into a single event (repeated smoke---detected events indicates the single fire---incident event)

c) Transform data by consolidation of events, and include them as a new event; for example power consumption data collected/received in a series of events can be consolidated into "energy usage" data)

d) Predict secondary events by analyzing heuristics of primary events. For example, event history may indicate that a sustained 'car park full' event is generally followed by 'dirty toilet' feedback event for a particular restroom located next to car park. According to this example, the correlation between events is not configured by the user (system integrator or facilities manager); rather it is a correlation that is automatically identified by the analytic engine through analysis of event and event history data.

[0065] Enabling Operational Intelligence

[0066] The present invention also processes events through analytics, allowing identifying meaningful events – both threats and opportunities, therefore enabling "operational intelligence." For example, when one smoke detector reports a fire in a particular location, a new metadata node may be initiated to build necessary connections between the city's metadata elements and to orchestrate certain actions such as notification to first responders. A few minutes later, more smoke detectors (in that Location) will likely report a fire. The system's response to 2nd and 3rd smoke detected events should be different from the first event. They act as "confirmation" to first event, rather than initiating new fire---incident scenarios.

[0067] Similarly, additional metadata can be appended to an event to add "more sense." In the same example above, if the first smoke detected alarm came from a detector that has produced a

‘false alarm’ a week ago, it would be wise to look for a 2nd confirmation from another detector, or a human confirmation, before the orchestration is carried out to respond to a fire.

[0068] Enabling Predictive Operations

[0069] According to a preferred embodiment of the invention, operational intelligence produced through this invention is ‘machine readable’. As a result, it allows predetermined orchestrations to be executed based on complex event conditions. This allows operators to manage the infrastructure predictively. For example, in a large space such as the lobby of a shopping mall, it takes a long time for the air-conditioning system to react when more shoppers come into the mall. This is due to the natural delay in warmed air reaching air-conditioning ducts on the roof, where sensors are placed to recognize the rising temperature. The present invention allows system integrators to setup an orchestration that acts on ‘people count’ that is monitored through an independent system to influence air- conditioning system to produce more air-conditioning, predictively.

Abstract

A metadata model of a city's Internet of Everything and use thereof to implement citizen engagement through 'user journeys' and system intelligence through automated response logic. A framework that allows system integrators to build a metadata model of a city's IoE, which metadata model then enables city managers to implement user journeys and system intelligence, in incremental fashion, as the business needs of the city evolves.

Please use Figure 1 for publication.

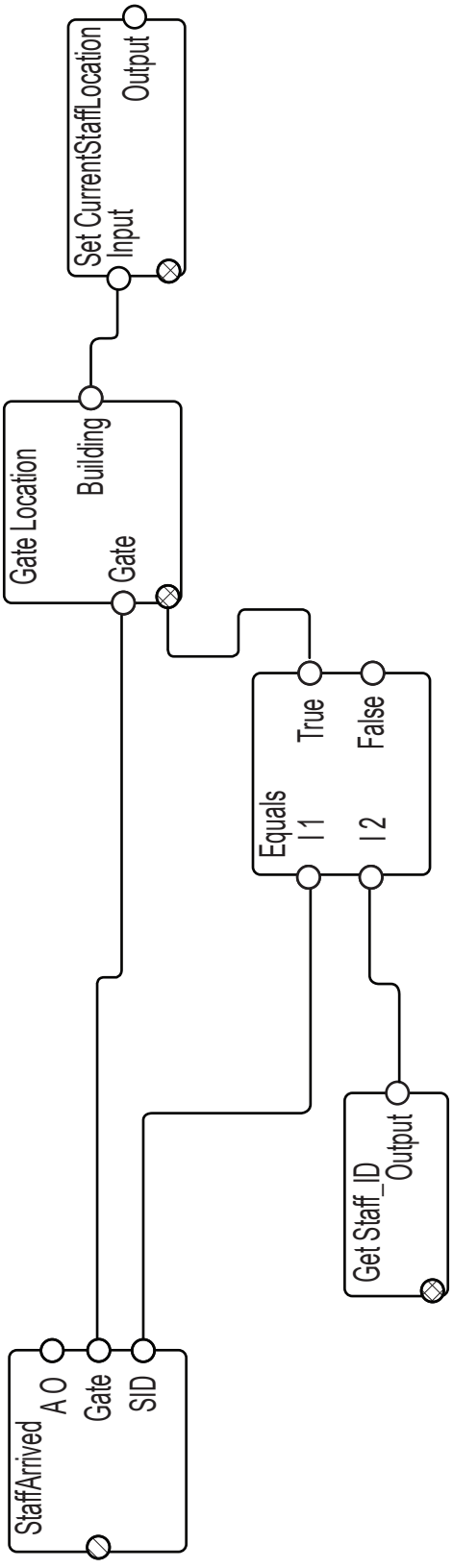

StaffArrived
A O
Gate
SID
Gate Location
Building
Gate
Set CurrentStaffLocation
Input
Output
Equals
I 1
I 2
True
False
Get Staff_ID
Output

FIG. 1

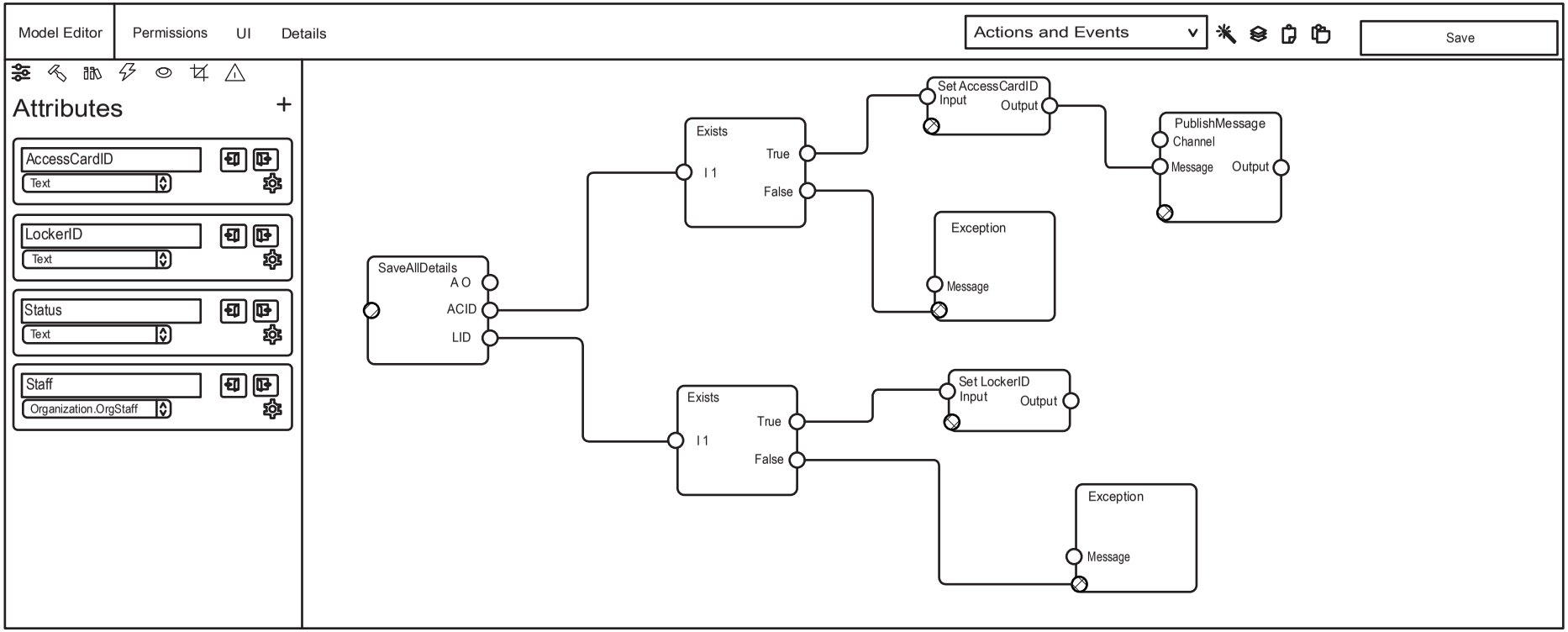

Model Editor
Permissions
UI
Details
Actions and Events
Save
Attributes
AccessCardID
Text
LockerID
Text
Status
Text
Staff
Organization.OrgStaff
SaveAllDetails
A O
ACID
LID
Exists
I 1
True
False
Set AccessCardID
Input
Output
PublishMessage
Channel
Message
Output
Exception
Message
Exists
I 1
True
False
Set LockerID
Input
Output
Exception
Message

FIG. 2

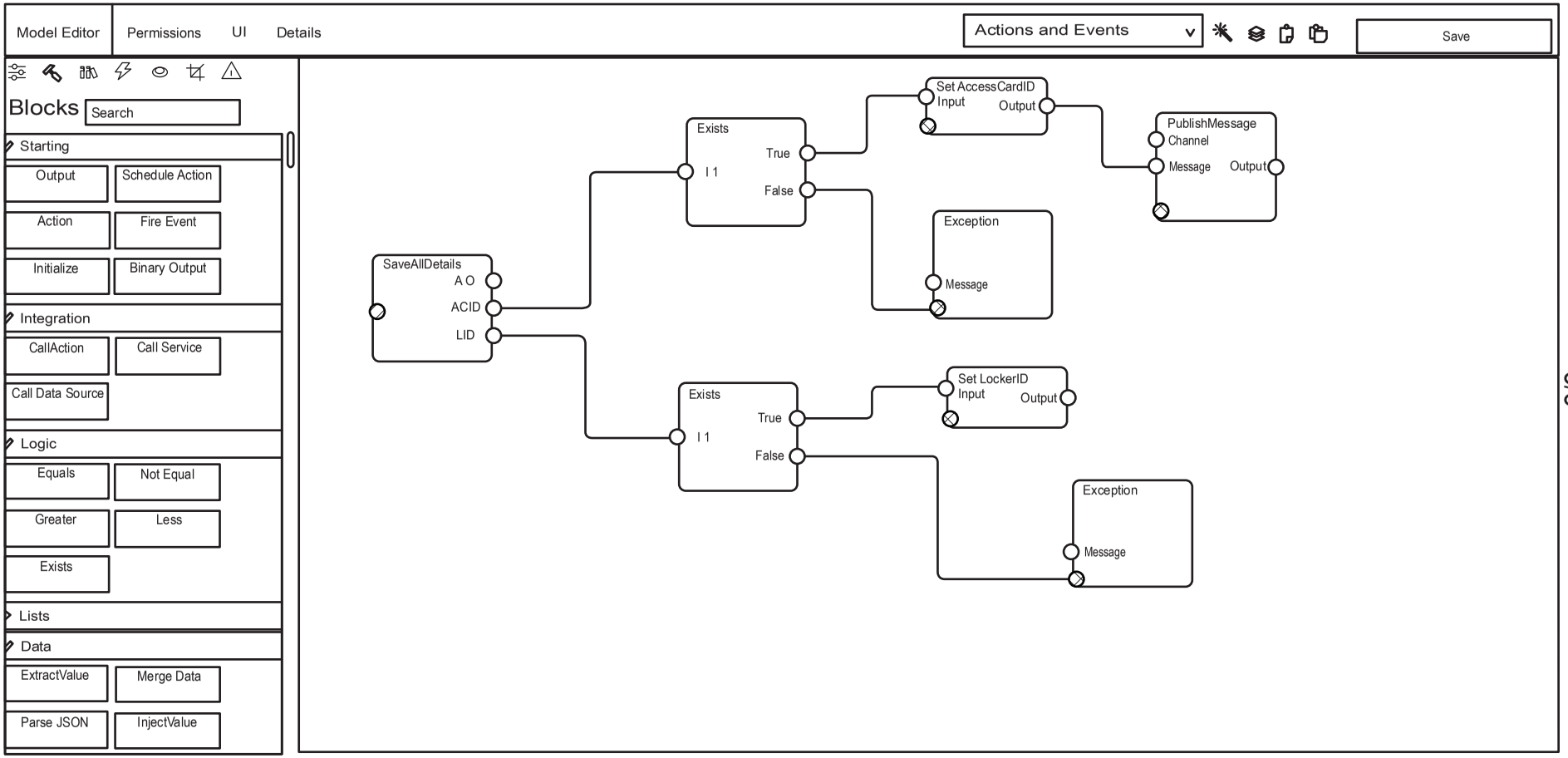

Model Editor
Permissions
UI
Details
Actions and Events
Save
Blocks
Search
Starting
Output
Schedule Action
Action
Fire Event
Initialize
Binary Output
Integration
CallAction
Call Service
Call Data Source
Logic
Equals
Not Equal
Greater
Less
Exists
Lists
Data
ExtractValue
Merge Data
Parse JSON
InjectValue
SaveAllDetails
A O
ACID
LID
Exists
I 1
True
False
Set AccessCardID
Input
Output
PublishMessage
Channel
Message
Output
Exception
Message
Exists
I 1
True
False
Set LockerID
Input
Output
Exception
Message

FIG. 3

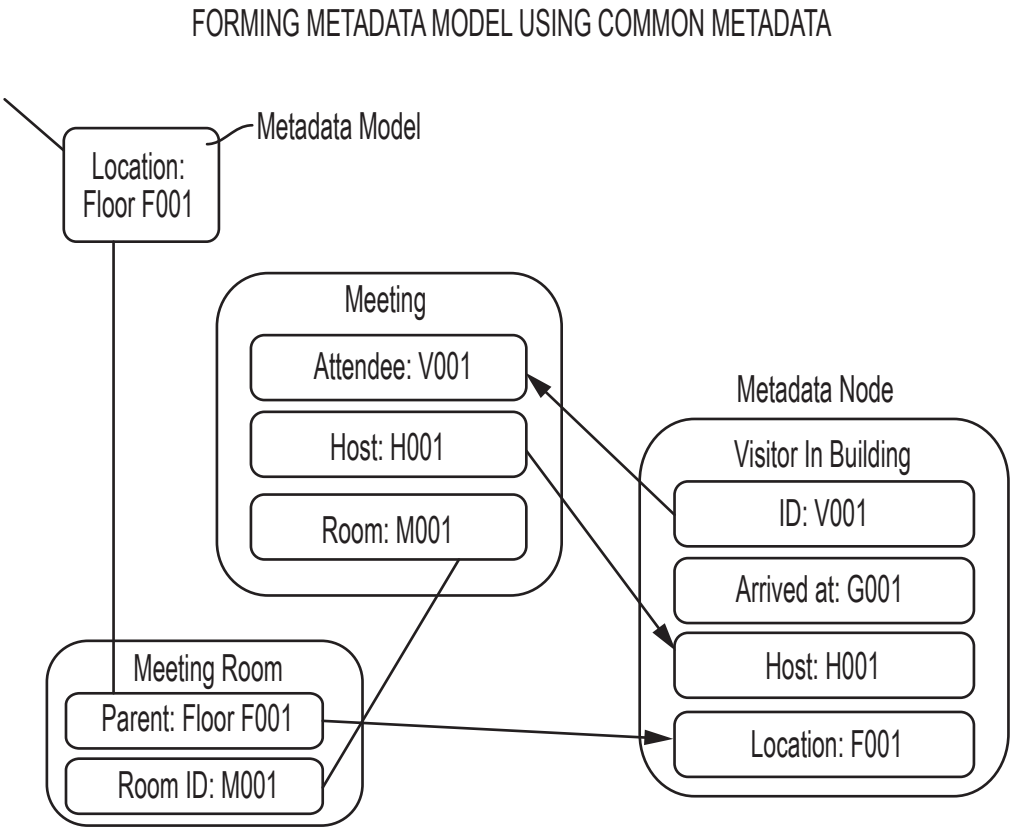

FORMING METADATA MODEL USING COMMON METADATA
Metadata Model
Location:
Floor F001
Meeting
Attendee: V001
Host: H001
Room: M001
Metadata Node
Visitor In Building
ID: V001
Arrived at: G001
Host: H001
Location: F001
Meeting Room
Parent: Floor F001
Room ID: M001

FIG. 4

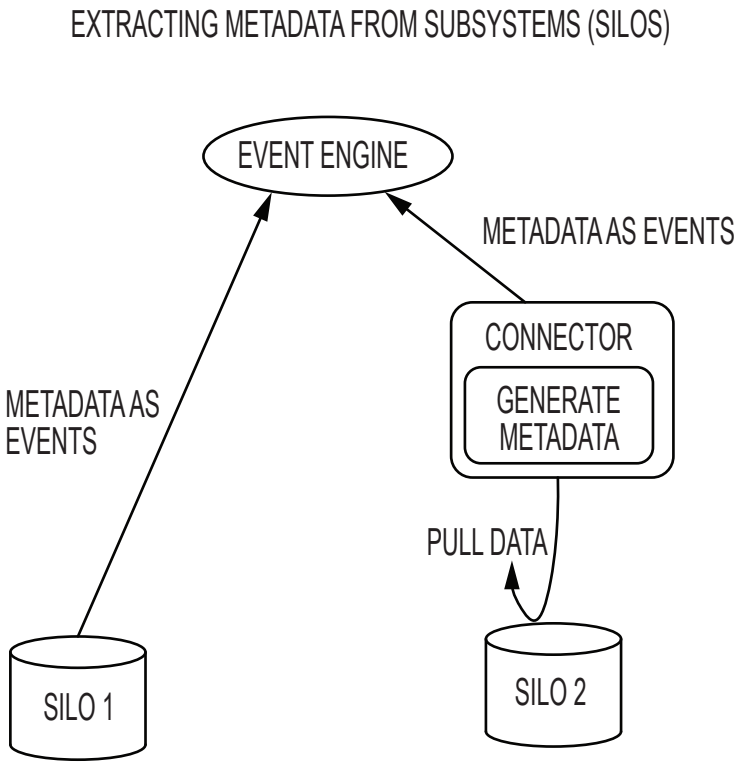

EXTRACTING METADATA FROM SUBSYSTEMS (SILOS)
EVENT ENGINE
METADATA AS EVENTS
CONNECTOR
GENERATE METADATA
METADATA AS EVENTS
PULL DATA
SILO 1
SILO 2

FIG. 5

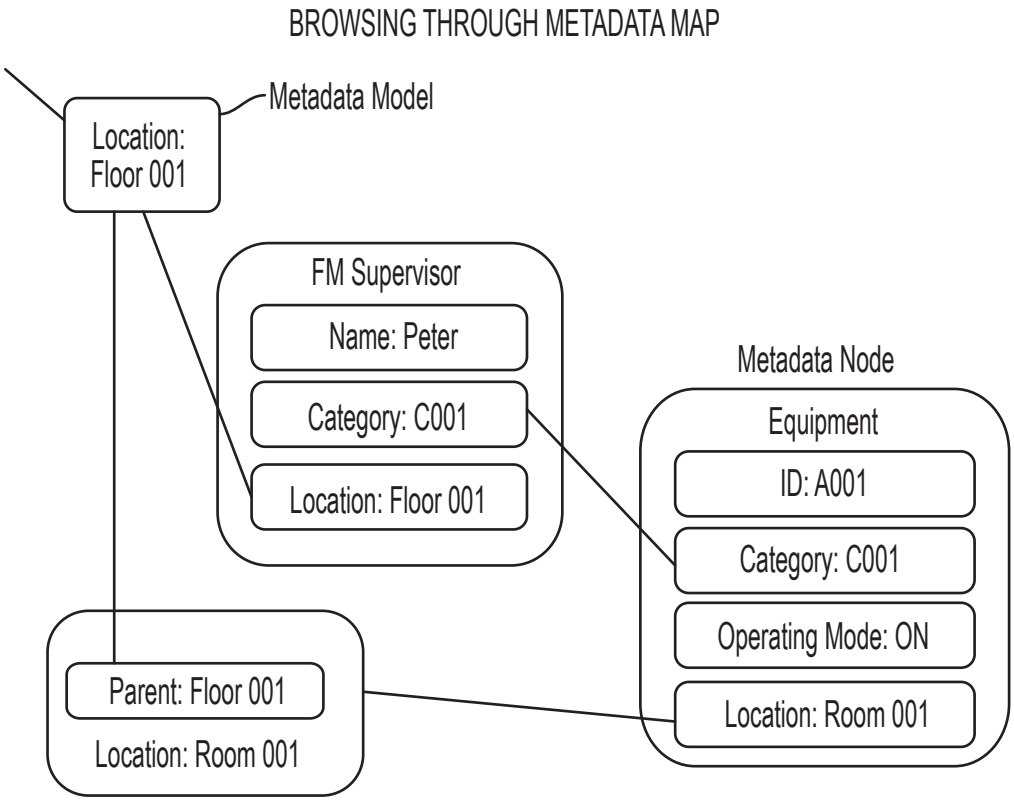

BROWSING THROUGH METADATA MAP
Metadata Model
Location:
Floor 001
FM Supervisor
Name: Peter
Category: C001
Location: Floor 001
Metadata Node
Equipment
ID: A001
Category: C001
Operating Mode: ON
Location: Room 001
Parent: Floor 001
Location: Room 001

FIG. 6

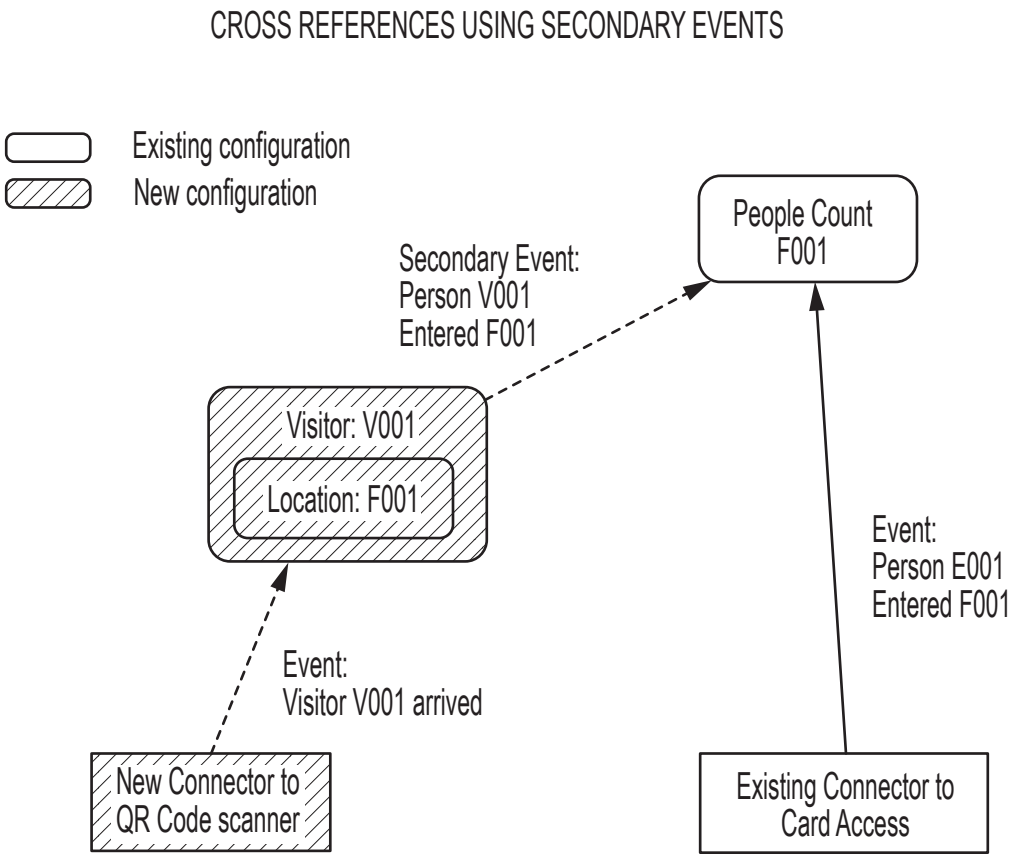

CROSS REFERENCES USING SECONDARY EVENTS
Existing configuration
New configuration
People Count
F001
Secondary Event:
Person V001
Entered F001
Visitor: V001
Location: F001
Event:
Person E001
Entered F001
Event:
Visitor V001 arrived
New Connector to
QR Code scanner
Existing Connector to
Card Access

FIG. 7

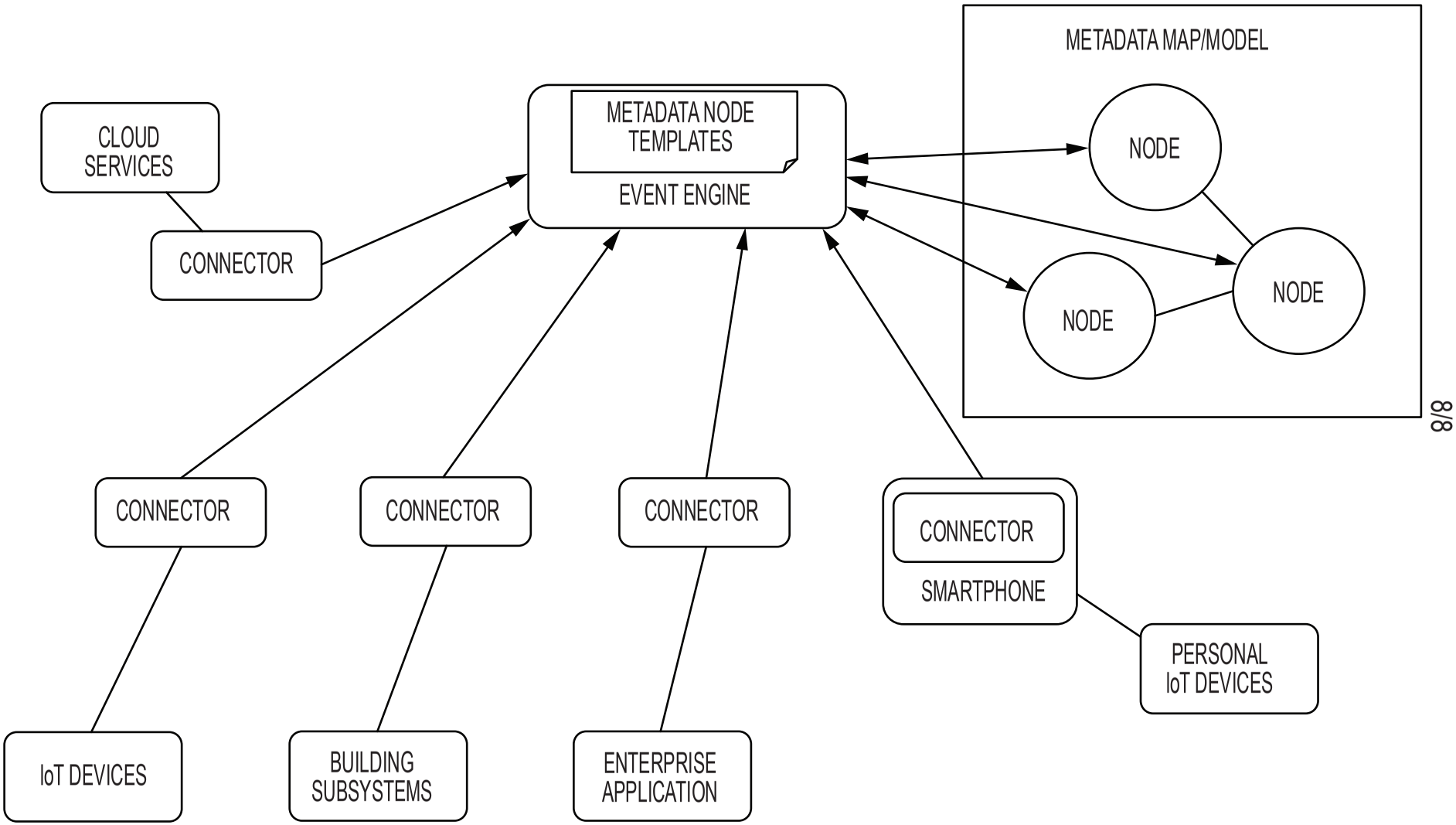

METADATA MAP/MODEL
NODE
NODE
NODE
METADATA NODE TEMPLATES
EVENT ENGINE
CLOUD SERVICES
CONNECTOR
CONNECTOR
CONNECTOR
CONNECTOR
CONNECTOR
SMARTPHONE
PERSONAL IoT DEVICES
IoT DEVICES
BUILDING SUBSYSTEMS
ENTERPRISE APPLICATION

FIG. 8