Towards interoperability of entity-based and event-based IoT platforms: The case of NGSI and EPCIS standards

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received February 19, 2021, accepted March 17, 2021, date of publication March 29, 2021, date of current version April 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3069194

Towards Interoperability of Entity-Based and Event-Based IoT Platforms: The Case of NGSI and EPCIS Standards

YALEW TOLCHA[1], AYALEW KASSAHUN[2], TEODORO MONTANARO[3], (Member, IEEE), DAVIDE CONZON[3], GEORG SCHWERING[4], JARISSA MASELYNE[5], AND DAEYOUNG KIM[1]

[1]Korea Advanced Institute of Science and Technology (KAIST), Daejeon 34141, Republic of Korea
[2]Information Technology (INF) Group, Wageningen University and Research, 6706 KN Wageningen, The Netherlands
[3]LINKS Foundation, 10138 Torino, Italy
[4]European EPC Competence Center GmbH (EECC), 41469 Neuss, Germany
[5]Flanders Research Institute for Agriculture, Fisheries and Food (ILVO), 9820 Merelbeke, Belgium

Corresponding author: Yalew Tolcha (yalewkidane@kaist.ac.kr)

This work was supported by in part by the IoF2020 Project through the European Union's Horizon 2020 Research and Innovation Program under Grant 731884, and in part by the Energy Cloud Technology Development Project through the Ministry of Science and ICT (MSIT) and National Research Foundation of Korea under Grant NRF-2016K1A3A7A0395205414.

ABSTRACT With the advancement of IoT devices and thanks to the unprecedented visibility and transparency they provide, diverse IoT-based applications are being developed. With the proliferation of IoT, both the amount and type of data items captured have increased dramatically. The data generated by IoT devices reside in different organizations and systems, and a major barrier to utilizing the data is the lack of interoperability among the standards used to capture the data. To reduce this barrier, two major standards have emerged: the Global Standards One (GS1) Electronic Product Code Information Service (EPCIS) and the FIWARE Next Generation Services Interface (NGSI). However, the two standards differ not only in the data encoding but also in the underlying philosophy of representing IoT data; namely, EPCIS is event-based, and NGSI is entity-based. Interoperability between FIWARE and EPCIS is essential for system integration. This paper presents OLIOT Mediation Gateway, now one of the incubated generic enablers offered by the FIWARE Foundation, that realizes the required interoperability between NGSI and EPCIS systems. It also demonstrates the applicability and feasibility of the Gateway by applying it to a real-life case study of integrating transparency systems used in a meat supply chain.

INDEX TERMS Agri-food, EPCIS, interoperability, IoT, NGSI, mediation gateway, tracking and tracing.

I. INTRODUCTION

Internet of Things (IoT) refers to the network of devices that can autonomously capture data, process them, and act on them. As the number of IoT-related devices and applications grows, as is the case of the agriculture and food (agri-food) sector [1], the communication between IoT systems has become increasingly complex. A major obstacle encountered by IoT adopters is the lack of interoperability among different systems and platforms. In fact, there are currently diverse IoT ecosystems, and they tend to use their own standards and formats for sharing and storing data [2], [3]. There are currently more than 450 IoT platforms on the market [4], which makes the IoT ecosystem highly fragmented.

While the opportunities provided by data generated by IoT devices are numerous, lack of interoperability creates a major barrier. For instance, realizing transparency systems in the agri-food sector requires sharing data captured by IoT devices across the supply chain operators, including farmers, food processors, third parties (such as logistic companies), and retailers. The interoperability of the data across the collaborating food operators is an essential requirement for using IoT data for transparency purposes.

Current efforts in standardizing IoT data and platforms has resulted in two major standards: the OMA/ETSI NGSI standard [5] and GS1 EPCIS standard [6]. In the agri-food sector, some food operators and retailers are adopting the

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

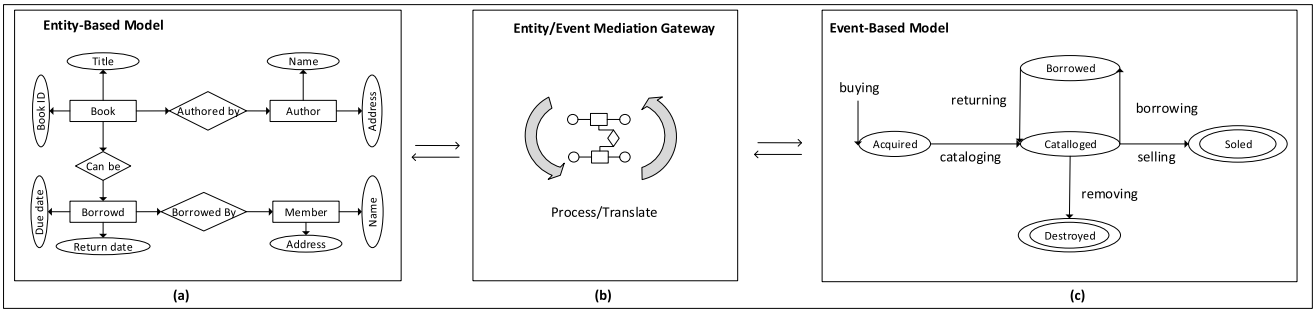


FIGURE 1. Interoperability between NGSI- and EPCIS-based systems.

EPCIS standard, while others are considering the NGSI standard. More specifically, various studies [7]–[10] have shown that NGSI platforms are more suitable and preferred by food operators that share real-time sensor data to maintain and manage resources. The EPCIS standard, on the other hand, is preferred by businesses that focus on tracking-and-tracing of products. For instance, Metro Group [11], one of the leading retailing companies in Europe, has adopted the EPCIS standard for its traceability system. The IBM food trust [12] platform, which is used by Walmart, also uses the EPCIS standard to capture traceability data. This makes the two standards complementary to each other, and the interoperability between them is crucial for a smooth end-to-end sharing of data among food operators.

Interoperability between these two standards is challenging because they differ not only in data encoding but also in the underlying philosophy of representing IoT data; namely, NGSI is entity-based and EPCIS is event-based. The entity-based approach has been studied in the context of business processes modeling, where business entities, which the activities act upon, are the key constituents of business processes. Business process models emphasize on the states and the transition between the states of the business entities [13]. The state transitions of a business entity, i.e., entity's life cycle [14], however, is better captured by event-based modeling because event information primarily captures the identity of the entity, the specific time and location of the event and add only the necessary contextualization information, relegating the remaining details about the entity to a separate master data repository.

The differences between the two standards' data models are schematically depicted in Fig. 1 using a simple example of the life cycle of books in a library. Fig. 1(a) shows the details about books as an E-R diagram—which is a suitable representation formalism for entities. Fig. 1(c) shows the same data as an event-model (comparable to the state-transition diagram [15]). This approach captures transparency information directly and is more suitable for representing tracing and tracking information, and for that reason, it has been adopted as an international standard for transparency. The former is the basis for the NGSI standard adopted by FIWARE, and the latter is the basis for the EPCIS standard adopted by GS1.

This work provides a solution to support interoperability between the NGSI and EPCIS through a Mediation Gateway, as shown in Fig. 1(b). This study considers a one-way translation from NGSI to EPCIS, which is most needed. Therefore, the Mediation Gateway needs to derive events from entity-based data. These, in turn, requires data processing and integration of the two different sets of APIs provided by NGSI and EPCIS.

This paper's remainder is organized as follows: Section II illustrates the works related to the present study and the information about NGSI and EPCIS data models, and their architecture and reference implementation. Section III explains the proposed technique for the interoperability of the event-based model with an entity-based model. Section IV presents the implementation scenario of the interoperability platform on a pig farming use case. In section V, the authors present the evaluation and discussion. Finally, concluding remarks are made in section VI.

II. RELATED WORKS AND BACKGROUND

A. IoT INTEROPERABILITY

In general, IoT interoperability can be categorized into four levels [16]: technical interoperability, syntactic interoperability, semantic interoperability, and organizational interoperability. This study can be classified as semantic interoperability, and thus we focus on related works on semantic interoperability.

Even though there are plenty of open IoT platforms available on the market [2], [17], only a few have focused on the integration of these platforms [18]–[23] and all of them focused only on entity-based platforms. For instance, Kovacs *et al.* [18] annotate OneM2M ontologies with semantic information to simplify the translation from OneM2M data to NGSI data. Similarly, An *et al.* [24] used a similar approach to translate data from NGSI to the oneM2M format. Sotres *et al.* [21] describe a smart parking use case by applying two entity-based platforms' interoperability. There is no previous work that tried to bridge entity-based(NGSI) with event-based(EPCIS) IoT platforms to the best of our knowledge. Moreover, this work is the first to introduce the entity-based life-cycle modeling to identify events for the IoT.

The only study we found in the literature that resembles ours is the work done by Overbeek *et al.* [25]. It proposed Event-Based data architecture for generating events by applying complex event processing from process and service layers of information systems. Our work differs from this work

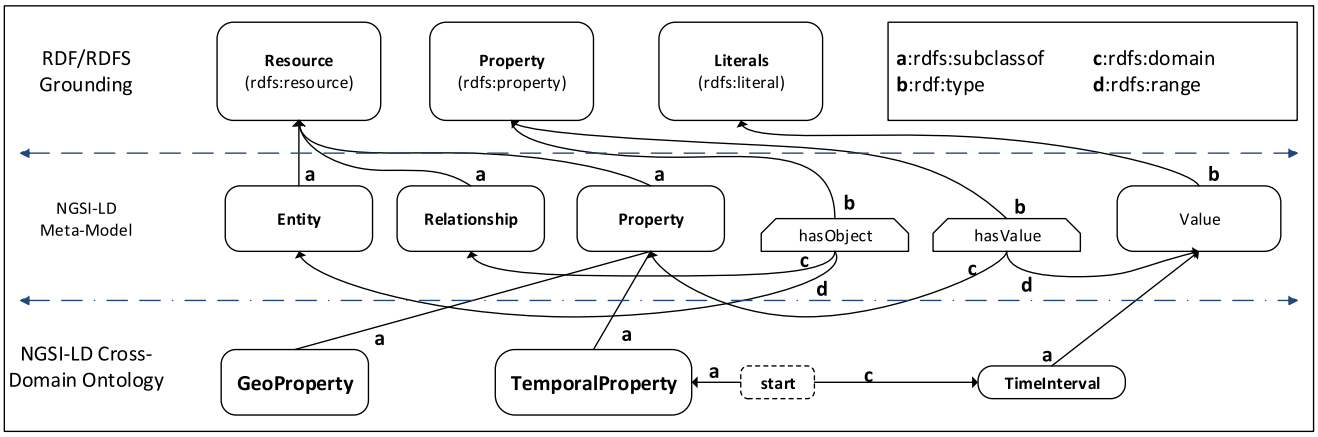


FIGURE 2. NGSI-LD core meta-model and the cross-domain ontology (adapted from [30]). The meta-model is composed of entity, relationship, and property, as shown in the middle. At the bottom, location and temporal entities are depicts as an example of cross-domain ontology.

because, unlike [25], our main objective is interoperability between two open standards. Moreover, the Mediation Gateway, in general, is different from complex event processing [26]–[28] since the former does not process event patterns over time. Instead, it only contextualizes the information received, constructs the event, and registers in the format of EPCIS.

B. NGSI

One of the standards currently promoted in Europe and used in large European research and innovation projects is the FIWARE NGSI standard. FIWARE is a foundation, but it also refers to a set of tools, called generic enablers, for supporting the development of Smart City, Smart Agri-Food, and Smart Industry applications. FIWARE is developed within the European Future Internet Public Private Partnership (FI-PPP) initiative [29].

The NGSI standard was first defined by the Open OMA in 2012 [30]. It is then enhanced by FIWARE, resulting in NGSI-v2 [31]. Finally, it evolved and standardized in November 2018 into the new NGSI-Linked Data (NGSI-LD) version of the standard [30] by the ETSI[1] Industry Specification Group for cross-cutting Context Information Management (ISG CIM).[2]

The NGSI-LD Information Model is based on the "Core Meta-Model" as represented in the central part of Fig. 2, and that corresponds to a formal specification of the foundation classes presented in the NGSI standard [5]. The upper part of the figure presents the classes that are used to represent context with a focus on the mapping of the NGSI-LD classes (also called resources) with the standard Resource Description Framework (RDF[3]) data model [32].

The central class of the NGSI-LD Meta-Model is the "Entity" resource that constitutes the virtual representation of physical objects in the real world. Its centrality is highlighted by the role of the other resources, namely, the "Property," the "Relationship," and the "Value" resources.

The prominent and most widely used implementation of the NGSI standard is the FIWARE Orion Context Broker [33] developed by the FIWARE Foundation in the Connecting Europe Facility (CEF) [34] initiative. The Orion Context Broker is the central part of the FIWARE platform and uses as a Representational State Transfer (REST) API [35] to capture, update, query, and subscribe to changes on context information.

C. EPCIS

The EPCIS standard [6] defines a data model along with a capturing and querying interface. Its Abstract data model defines two kinds of data: Event data and Master data, as depicted in Fig. 3. Event data is generally used to capture dynamic data from business processes in the form of EPCIS events. Master data, on the other hand, is the additional data that provides the necessary context for interpreting the event

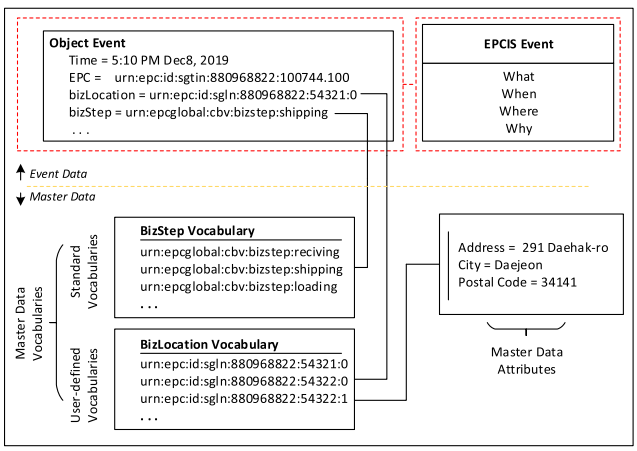


FIGURE 3. EPCIS data model architecture which shows event data and master data (standard and user defined vocabularies).

[1]https://www.etsi.org/
[2]https://portal.etsi.org/tb.aspx?tbid=854&SubTB=854#/
[3]https://www.w3.org/2009/07/NamedGraph.html

data. Based on the definitions, it is up to the industries (along with end-users) to model their real-world business information as EPCIS events and master data.

EPCIS uses multiple vocabularies to model processes involving physical or digital entities that happen in the real world. To create a common understanding of vocabularies' semantics among parties who exchange EPCIS events, GS1 published the Core Business Vocabulary (CBV) standard [36]. The standard defines different vocabulary structures and specific values of some of the vocabularies used during the construction of events.

EPCIS events are basically designed by annotating four pieces of contextual information that can describe business event and they are commonly referred to as the four dimensions of an EPCIS event, namely: the ''What'', ''When'', ''Why'' and ''Where'' of an event. The ''What'' dimension contains one or more unique identifiers for physical or digital objects (classes). The ''When'' dimension captures the moment in time at which the EPCIS events occurred. The ''Where'' dimension of the event describe where the event took place. The ''Why'' dimension denotes a specific activity within a business process of the event. More additional information can also be included to the event.

According to the GS1 Software certification program [37], there are currently more than 20 certified implementations of the EPCIS standard, including major software systems of SAP, IBM, Oracle, and Microsoft. IBM implemented EPCIS as part of its IBM Food Trust [12] project to enable traceability of food. Oliot-EPCIS [38] and Fosstrak [39] are some of the standard's well-known open-source implementations. The European EPC Competence Center (EECC) has implemented its commercial EPCIS 1.2 compliant solution ''EPCAT.''

III. OLIOT MEDIATION GATEWAY DESIGN AND IMPLEMENTATION

This section introduces the OLIOT Mediation Gateway, which was designed and developed by the authors and has now become part of the generic FIWARE enablers [40]. As depicted in Fig. 4, the Mediation Gateway enables automatic interoperability between the FIWARE Context Broker (the widely used implementation of the NGSI standard) and the EPCIS system (in this study, the EPCAT and OLIOT implementation of EPCIS from EECC and KAIST, respectively, were used to test the Mediation Gateway).

The Context Broker is used to receive data from IoT devices and construct NGSI compliant entity data. The data received from an IoT device is used to create a new NGSI entity or update the state of an existing NGSI entity. The Mediation Gateway receives the updates passively as notifications or actively queries for updates, and when Gateway receives NGSI entity data, it generates EPCIS events based on it. The EPCIS system captures the EPCIS events sent by the Mediation Gateway and stores them in an EPCIS repository so that any accessing application can access the events via the EPCIS standardized query interface.

The Context Broker provides synchronous and asynchronous interfaces to access entities generated by a context producer. The Context Broker has four components [41]: Entity Manager, HyperText Transfer Protocol(HTTP) Request Receiver, HTTP Response Sender, and a Repository of data about entities and subscriptions (left side of Fig. 4). When a context producer publishes an entity to the Context Broker, the Entity Manager receives the data via the HTTP interface and stores it into the repository. Similarly, subscriptions are stored in the repository by the Entity Manager. Upon any update of entities' attribute values or when a new entity is published, the Entity Manager sends notifications to the subscribers (in this case, the Mediation Gateway) via its HTTP response sender interface. From the perspective of the Context Broker, the Mediation Gateway is a context consumer.

The EPCIS system provides, just like the broker, both synchronous and asynchronous interfaces to capture event data. The modules of an EPCIS system in implementations like Oliot-EPCIS [38] and EPCAT [42] can be grouped into four parts: EPCIS Capturing Interface, EPCIS Repository, EPCIS Subscription Manager, and EPCIS Querying Interface (as shown right side of Fig. 4). The Capturing Interface validates the incoming events according to the standard schema and sends them to the EPCIS repository—the repository stores EPCIS event data. The repository also contains master data and subscription information. The Subscription Manager manages scheduled and triggered queries. Applications that want to get data from the EPCIS system interact via the Querying Interface. From the perspective

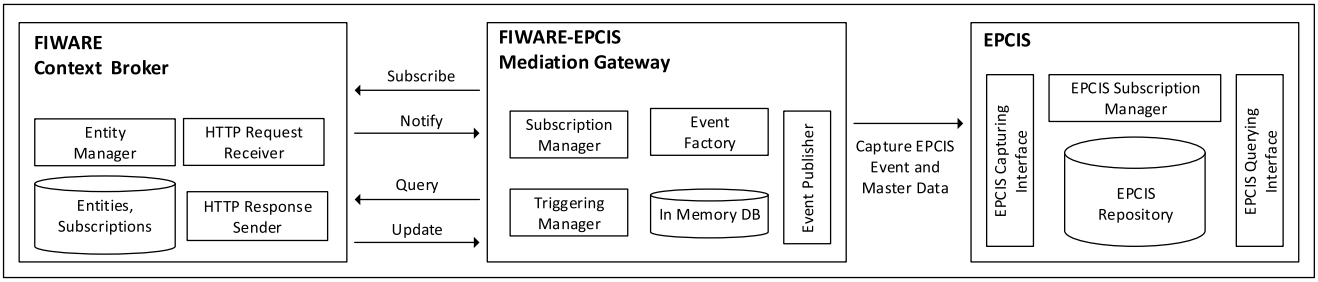


FIGURE 4. FIWARE-EPCIS mediation gateway.

of the EPCIS system, the Mediation Gateway is an EPCIS Capturing Application.

The Mediation Gateway (central part of Fig. 4) translates the information captured by the Context Broker in the form of NGSI entity data and capture into EPCIS events. To do so the Mediation Gateway has five components: Subscription Manager, Triggering Manager, Event Factory, Temporary Entity State Repository, and Event Publisher modules. These components are described below.

A. SUBSCRIPTION MANAGEMENT

The ''Subscription Manager'' manages asynchronous communication between the Context Broker and the Mediation Gateway. It opens a subscription endpoint corresponding to each entity in the Context Broker. Since NGSI entities are grouped by their domains, the endpoints are designed hierarchically and start with a domain name followed by entities. This helps to uniquely identify each subscription endpoint. The subscription Uniform Resource Locator (URL) is also designed to accommodate future NGSI models. It is composed of the context address, the NGSI version, application data domain, and a specific entity within the application data domain; {FIWARE_context_adress}/{NGSI_version}/subscription/{application_data_domain}/ {entity_name}. This makes the Mediation Gateway design extendible not only to new domains but also to new NGSI standards. Upon any notification of entities attribute's value change, the Subscription Manager receives data from the Context Broker and passes it to the Event Processor module.

B. TRIGGER MANAGEMENT

The ''Triggering Manager'' of the Mediation Gateway handles synchronous communication with the Context Broker. When an entity's status needs to be checked periodically or upon a status change of other related entities, the Triggering Manager triggers a request for the entity's status via a request-response interface. Whenever there is a state change, the Context Broker captures the information by updating the specific entity's attribute values in real-time. In NGSI, the ability to retrieve historical data is limited. Entity's attribute's value reflects only the latest update.

C. EVENT FACTORY

The ''Event Factory'' module receives data from the Subscription and Trigger Manager, determines if the status of entity has changed, and if the status has changed, produces an EPCIS event. The change in status, and thus the creation of an event, is identified using Entity Life-cycle History (ELH) modeling [15], [43], [44]. An ELH is a model that integrates entities (objects) with process to make an information system [14]. To construct an ELH model, first, all entities are identified from the E/R model representing the NGSI data model (for instance, the E/R model books ELH showed in Fig. 1). Secondly, all process steps (real-world transactions) that have a net effect of changing the state of an entity (e.g., acquire, catalog, sell, etc.) are identified—data flow analysis is used to identify the process steps [44]. Finally, events are identified by applying each process step to each entity. The effect of applying the process steps can be categorized as Create, Read, Update and Delete (CRUD) operations. Events are considered if only the process steps have resulted in an entity state change. This can be (using the example of the books life cycle), for instance, the process step acquire has a Create (C) effect on the entity Book; catalog has an Update (U) effect; sell has a Delete (D) effect. The creation of EPCIS events is triggered every time a ''significant state change'' (attribute's value change) occurs in the entity life history of the NGSI entity data model. Thus, this module's main functionalities are to map the entity status change with the corresponding events and annotate the events with the context information as defined in EPCIS.

An entity's attribute value change in the NGSI data model can cause the generation of one of the following three event types: Simple Translation, Simple Event, or Complex Event. When the change of the attribute value simply describes the status of the entity under consideration, the change can be directly translated into events, i.e., simple translation. For example, when there is a simple sensor reading in a building, the change can be easily translated to the creation of an event. This kind of translation can be configured to be triggered periodically or when the change meets certain conditions. Simple event generations are like simple translation in the sense that both occur when there is an update of one or more attributes within a single entity. But in the case of simple event generation, the change of attributes must result in a change in context. For example, growth events can be generated from a weight change. Lastly, events that are generated due to an update of attributes from multiple entities are referred to as complex event generation. In this case, the entities can be generated by multiple IoT devices. For example, data from sensors attached to an animal and sensor information from the environment can be combined to generate alerts about the health status of the animal.

During the construction of the events, each event is contextualized, which means the four key dimensions of the EPCIS event are determined, which are: object(s) that are the subject of the event (''What''), the date and time (''When''), the location at which the event occurred (''Where''), and the business context (''Why''). These four sets of data fully describe what happened in that specific time and location and fully describe the entity's specific life cycle.

D. TEMPORARY ENTITY STATE REPOSITORY

The ''Temporary Entity State Repository'' (Redis,[4] an in-memory database, is used in the current implementation) is used to store the previous entity state temporarily. It is important to store entity states in order to determine if a significant state change has occurred. For instance, in animal farming, to check a significant weight change and generate a growth event, an animal's last known weight needs to be stored.

[4]https://redis.io/

The use of in-memory database instead of the traditional database speeds up the translation process.

E. EVENT PUBLISHER

The "Event Publisher" module publishes the generated event, which means the data is translated into the EPCIS document standard, which is the EPCIS eXstensible MArkup Language (XML) format, and pushed via its standard interface.

The Mediation Gateway also captures or updates EPCIS master data whenever there is any change and is also responsible for mapping the local identification system used in FIWARE into the globally managed GS1 identification system.

IV. USE CASE: PIG FARMING

In this section, the case study validation is presented. In the first subsection (section IV-A), the use case is described in detail. The use case comes from a pig farm where the IoT systems are deployed, and data is gathered. Then the NGSI data model for entities of the case study is presented in section IV-B. Next, the corresponding EPCIS event model is presented in section IV-C. Finally, the application of the Mediation Gateway to convert the NGSI data into EPCIS events is presented in section IV-D.

A. THE USE CASE

The case study used in this study is one of the 33 IoT use cases of the Internet of Food & Farm 2020 (IoF2020) project. The case study is entitled "Pig Farm Management" and demonstrated IoT devices' application in precision pig farm management [45]. The test farm is located in Belgium and is part of the ILVO[5] research institute pig test farms. The "Pig Farm Management" use case aimed to innovate pig farm management through monitoring feed and water consumption, growth, and health parameters of an individual or groups of pigs.

Several IoT sensors were deployed at the test farm in order to collect real-time data. Fig. 5 shows the setup of the experimental compartment at the farm. The compartment contained 120 fattening pigs equally divided into eight pens. There was an equal number of male and female pigs, and a pen contained either exclusively male or exclusively female pigs. Each pig was identified uniquely using Radio Frequency IDentification (RFID) tags attached to both ears for ease of reading. To measure how much fodder was consumed by each pig, the PigWise[6] feeder and the Nedap[7] feeding station (feeder and feeding station, respectively, in Fig. 5) were used. The PigWise system uses High Frequency (HF) RFID tags for the identification of the pig that is eating [46]. While the Nedap feeding stations (pens 1, 3, 5, and 7) use Low Frequency (LF) RFID for the same purpose. The Nedap feeding station weighs the feed portions delivered to each pig and derives the amount of feed consumed. The PigWise feeder does not measure feed intake; only feeding patterns were registered. Furthermore, the PigWise drinker system was used to register water flow and drinking patterns [47]. The drinker uses a flow meter to measure the amount of water consumed and the drinking patterns. Two different solutions were used to weigh pigs: the one provided by the Nedap system (weighing scale in Fig. 5) and a regular scale found outside of the pens. The Nedap feeding stations weigh each pig while they are in the station, while the regular solution was used when the pigs are entering or exiting their pens. Finally, the climate of the pens was measured through a dedicated climate system (Climate Sensor in Fig. 5) that used common IoT sensors from Monnit[8] to measure the luminosity, temperature, and humidity.

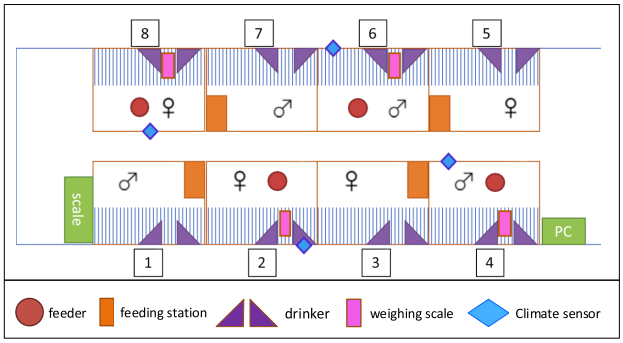


FIGURE 5. Testbed setup.

Among the collected pig-related information, the most important ones that are mapped to the NGSI data model used in this study are the Animal ID, the pig location that represents the pen in which the pig is located, the visit time that represents the time at which the pig starts the feeding visit, the duration, that is the duration of the feeding visit in seconds, the weight that is the median weight of the pig measured during the feeding visit in grams, and the feed intake that represents the feed intake of the pig during the feeding visit in grams

B. THE NGSI DATA MODEL

Fig. 6 shows the entities, their attributes, and their relationships used to represent pig-related information gathered using the Orion Context Broker deployed at the farm. There are six major entities, which are Pig, Pen, Building, Farm, Slaughterhouse, and SlaughterPig. The devices and sensors measure one or more parameters of these entities, and sometimes in groups (which is the case for pigs), and send updates to the Context Broker. The case depicted in Fig. 5 shows only one compartment. Generally, a pig farm contains one or more buildings (pig stables); each building will have multiple compartments, and each compartment will have multiple pens.

[5]https://www.ilvo.vlaanderen.be/
[6]https://ec.europa.eu/eip/agriculture/en/find-connect/projects/pigwise
[7]https://nedap.com/
[8]https://www.monnit.com/

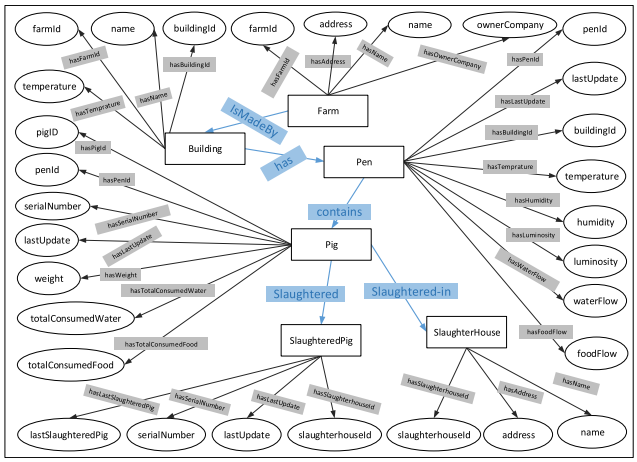


FIGURE 6. Pig farming NGSI data model.

Furthermore, information from the associated slaughterhouse is represented by two dedicated entities. A JavaScript Object Notation (JSON) example of pig entity data coming from the context broker is presented in appendix A.

C. EPCIS EVENT MODEL

The event model developed in the MEAT TRANSPARENCY AND TRACEABILITY [45] use case of the IoF2020 for the purpose of tracing and tracking of pigs and pork is shown in Fig. 7. An example of an EPCIS event data is shown in Appendix B. The concept of tracking and tracing that led to the event model is as follows. When piglets are born, the ''Birth'' event will be captured. When piglets are moved to (adopted by) another sow, an ''Adoption'' event will be registered. A ''Growth'' event represents any increase or decrease of weights of a pig or a sow in a period of time. A ''Feed intake'' or a ''water intake'' event is captured when there is any pig's food or water consumption. When artificial insemination is done, an ''Insemination'' event is captured. A ''Pen-Up'' event is meant for any addition of animals to a compartment or to a pen. ''Shipping'' and ''Receiving'' events represent any shipping or receiving of animals from or to a new owner. A ''Slaughter'' event is used when an animal is slaughtered at a slaughterhouse. The ''Vet'' event is registered when a veterinary inspects or treats an animal. Any events related to environmental sensor measurements are captured with an ''Environment'' event. In fact, all these events are captured as standard EPCIS object events and differentiated by the specific business step of the event, such as the birth and the adoption.

EPCIS events are constructed by assigning the values of the four dimensions of EPCIS events. Assigning a unique ID to each entity represents the ''What'' dimension of the events and establishes the foundation for traceability. GS1 defines multiple identification systems for assigning IDs to objects [26], including Global Trade Item Number (GTIN) and Global Location Number (GLN). GTIN is used to identify a specific class of objects from a specific company. For instance, all pigs from a farm can be assigned to a single GTIN number. A serial number can be added at the end of a GTIN number to create Serialized GTIN (SGTIN), which can be used to identify a specific entity of a product (for instance, a pig) or a specific service. If no individual identification is possible, lot numbers can be added, resulting in a Lot GTIN (LGTIN). GLNs are used to identify locations. Similarly, Serialized GLN (SGLN) can be used to identify specific (sub-) locations within a company. Accordingly, a hierarchical ID system was designed for the use case as shown in Fig. 7 (b). Each pig is assigned an SGTIN, and a group of pigs is assigned an LGTIN. SGLNs are assigned to each pen and compartment. GLN is assigned to the farm.

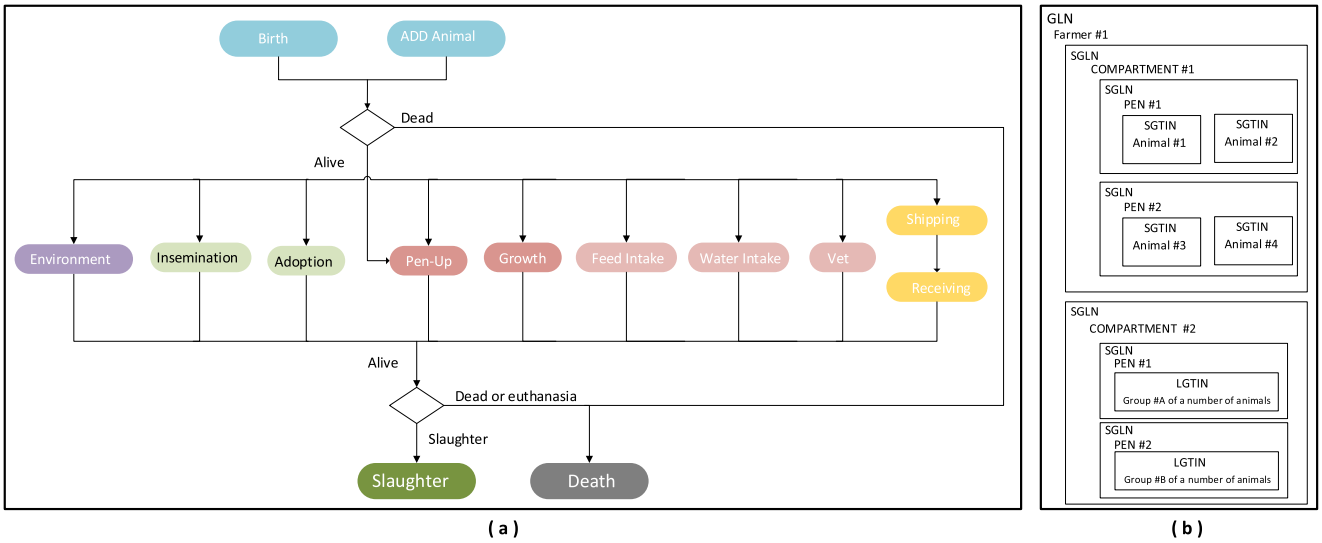


FIGURE 7. Pig event flow model.

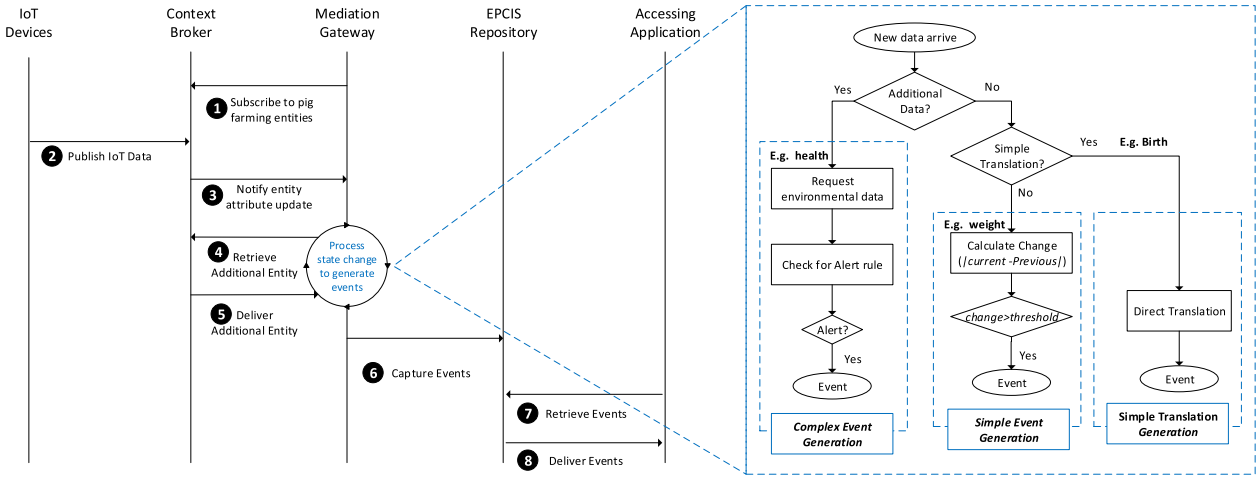


FIGURE 8. Overall data mediation process.

D. DATA MEDIATION VIA THE GATEWAY

Fig. 8 shows a sequence diagram from data capture to NGSI up to data accessing through EPCIS by applying the Mediation Gateway. First, the Mediation Gateway subscribes to the context broker to get notifications and new data. After a new update is pushed to the Context Broker by the IoT devices, the Context Broker notifies the mediation gateway with the newly updated data. The Mediation Getaway processes the updated information to check for state change and generate events. If the data processing needs additional entity information, the Mediation Gateway gets the data with a request-response method. It then captures the events to EPCIS for later retrieval by an EPCIS application.

Fig. 8 also presents a sample workflow of the Mediation Gateway to show the three types of event generation presented in section III-C via three use cases. In the first case, an update of health information requires additional environment entity information to generate an alert event (a type of Complex Event). The second case shows the workflow when a weight update information is presented. In this case, the change with the previous weight is compared against a threshold value to generate a growth event (a type of Simple event). The last case presented a workflow of generating birth events from birth update information (a type of Simple Translation).

V. DISCUSSION AND EVALUATION

A. DISCUSSION AND FUTURE WORK

Due to globalization, the distance that food travels from source (producer) to destination (consumer) has increased. This makes food quality and safety one of the major concerns in the food industry. To ensure the integrity of the food supply chain, all involved parties demand verifiable evidence of the source and destination of food. To tackle these requirements, traceability systems that provide information on the origin, processing, and distribution of foodstuffs are required. Although the pilot study used in this paper demonstrates the ability to capture information only up to the slaughtering step, full traceability can be achieved when subsequent organizations capture and open their data either via their own EPCIS instances or using the FIWARE context broker. For instance, in Europe, METRO Group is using EPCIS for its ProTrace Application to Track and Trace meat, fish, etc. [11]. GS1 Germany provides the service using its fTrace system developed by a daughter company [49].

Even though Mediation Gateway's current implementation considers only pig farming, it can be easily extended to other domains. Except for the Event Processor module, all the components are generic and can be used to other domains with little or no modifications. Interfaces for the domains specified by FIWARE Smart Data Models are already included. The subscription URL design considers both the version of the FIWARE standard and the various domains. FIWARE provides a framework (FIWARE Catalog) [40] to assemble open-source platform components with other third-party platform components (FIWARE Enabler) to accelerate the development of Smart Solutions. The Gateway has been accepted as one of the FIWARE enabler which helps for the opensource community to extend to different other domains.

The current implementation considers the NGSI-V2 data model instead of the latest version NGSI-LD. During the design and implementation of the FIWARE system for the pig use case, there was no stable version NGSI-LD standard implementation. Considering the minimum difference in the data model, the authors will provide the NGSI-LD version through the FIWARE Enabler open-source project initiative as soon as there is a stable implementation of NGSI-LD.

B. PERFORMANCE EVALUATION

Since the live data coming from the ILVO farm in the use case is small, to evaluate the Mediation Gateway performance, the authors conducted a load testing experiment using

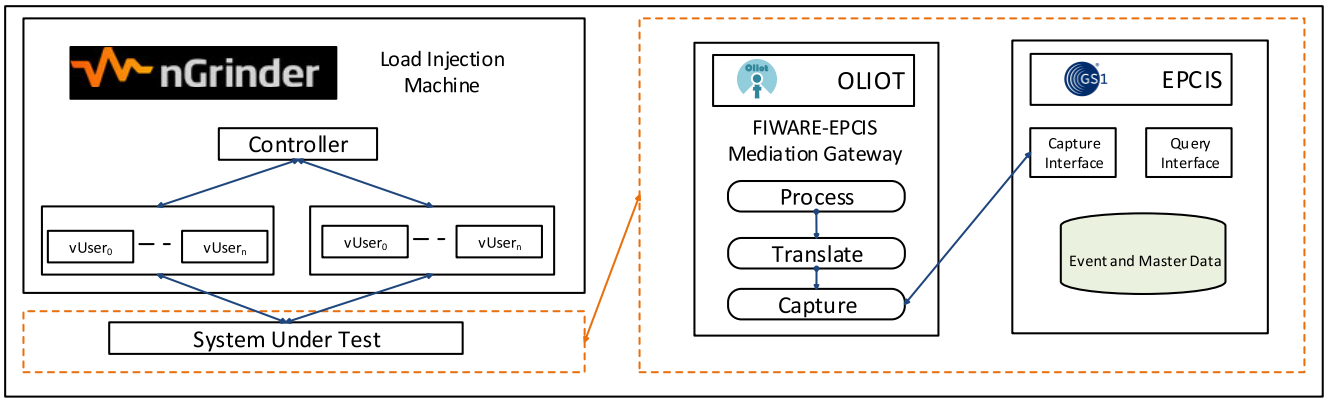
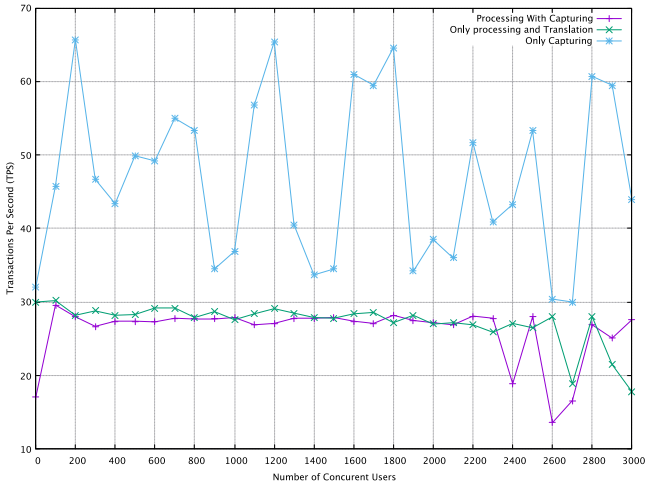
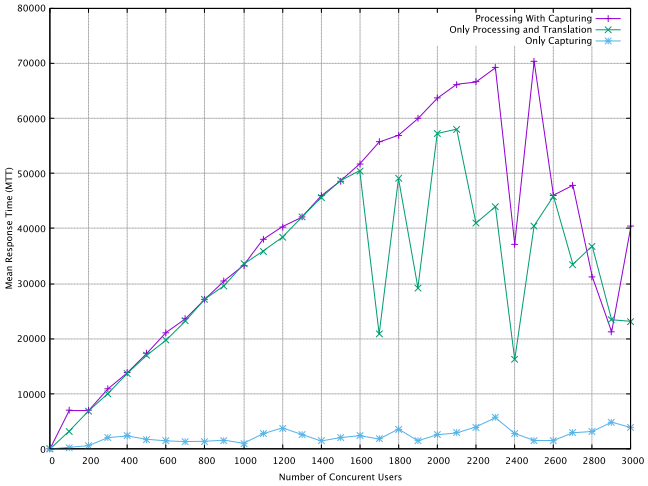


FIGURE 9. Experimental setups.



(a) TPS Performance Result



(b) MRT Performance Result

FIGURE 10. Performance evaluation result.

nGrinder,[9] a framework for running test scripts across several machines. An experimental setup has been prepared, which contained nGrinder load generator, Mediation Gateway, and EPCIS, as depicted in Fig. 9. The load generator contains a controller, which controls the different agents who are available in the same machine or in a different machine. Each agent can create multiple virtual users (vUsers) that concurrently execute the controller's job to inject load into the system under test. The experimental setup of each system is described in Table 1.

To evaluate the Mediation Gateway under different test loads, the authors set up two agents to create concurrent users starting from 1 up to 3000 in a range of 100. Each concurrent user generated an NGSI based pig model that simulated feed intake, water intake, and weight change. This means each virtual user simulates users running in parallel and pushes NGSI data to the Mediation Gateway. Therefore, for each concurrent data feed, three events will be generated:

[9]http://naver.github.io/ngrinder/

TABLE 1. Experiment environment setting.

	Load Generator	Mediation Gateway	EPCIS
CPU Type	Intel core i7	Intel core i7	Intel core i7
# of CPU core	8	8	8
Speed Per core	2.80GHz	3.40GHz	2.80GHz
Memory	8GB	8GB	8GB
OS	Windows8	CentOS Linux 7	Linux Mint 18.2
Web App. Server	-	Apache 8.5	Apache 8.5
Database	-	Redis	Mongodb 3.4

a feed-intake event, a water-intake event, and a growth event. To reveal the Mediation Gateway overhead due to processing and translation, the authors created three scenarios: 1) Processing with Capturing: the Mediation Gateway processes the NGSI data from the load injector and translates it to events, and captures it to EPCIS. 2) Only Processing and Translation: in this scenario, the Mediation Gateway only processes and

```
{
"id": "Pig-907e8b9d-2d6b-4149-a1ee-6aa7cefc2973",
"type": "Pig",
"arrivalTimestamp": { "type": "Text",
"value": "", "metadata": {}},
"buildingId": { "type": "Text",
"value": "", "metadata": {}},
"companyId": { "type": "Text",
"value": "8b6e0aa4-08fc-4f6f-960d-5a65**",
"metadata": {}},
"compartmentId": { "type": "Text",
"value": "", "metadata": {}},
"endTimestampAcquisition": { "type": "Number",
"value": 1534895999, "metadata": {}},
"endTimestampMonitoring": { "type": "Number",
"value": 1534895999, "metadata": {}},
"farmId": { "type": "Text",
"value": "9a68ea4e-348e-424e-9346-6f***",
"metadata": {}},
"lastUpdate": { "type": "DateTime",
"value": "2018-04-26T19:03:25.00Z",
"metadata": {}},
"penId": { "type": "Text",
"value": "6053fdc7-33c7-4af9-907a-957***",
"metadata": {}},
"pigId": { "type": "Text",
"value": "907e8b9d-2d6b-4149-a1ee-6ac***",
"metadata": {}},
"serialNumber": { "type": "Text",
"value": "", "metadata": {}},
"sex": { "type": "Text",
"value": "B", "metadata": {}    },
"startTimestampAcquisition": { "type": "Number",
"value": 1519430400, "metadata": {}},
"startTimestampMonitoring": { "type": "Number",
"value": 1519430400, "metadata": {}},
"totalConsumedFood": { "type": "Number",
"value": 198, "metadata": {}},
"totalConsumedWater": { "type": "Text",
"value": "", "metadata": {}},
"totalTimeConsumedFood": { "type": "Number",
"value": 430, "metadata": {}},
"totalTimeConsumedWater": { "type": "Text",
"value": "", "metadata": {}},
"weight": { "type": "Number",
"value": 30000, "metadata": {}}
}
```


LISTING 1. FIWARE pig entity example.


```
<?xml version="1.0" encoding="UTF-8"
standalone="yes"?><EPCISQueryDocumentType
xmlns:ns2="http://www.unece.org/cefact/namespaces/
StandardBusinessDocumentHeader"
xmlns:ns4="urn:epcglobal:epcis:xsd:1"
xmlns:ns3="urn:epcglobal:epcis-query:xsd:1">
<EPCISBody> <ns3:QueryResults>
<queryName>SimpleEventQuery</queryName>
<resultsBody><EventList><ObjectEvent>
<eventTime> 2019-11-18T10:57:15.138Z
</eventTime><recordTime>2019-11-18T10:57:15.927Z
</recordTime><eventTimeZoneOffset>
-05:00</eventTimeZoneOffset>
<baseExtension> <eventID>
7903bbaf-5550-4e34-967f-cbf901591d6d </eventID>
</baseExtension>
<epcList> <epc>
urn:epc:id:sgtin:88000269.444.5af14002-
3f27-4989-ae91-3dad4b5c96c900224
</epc></epcList>
<action>OBSERVE</action>
<bizStep>urn:gs1:epcisapp:farm:pig:growth</bizStep>
<disposition>
urn:epc:id:sgln:88000269.444.45b01a1c-
6fa2-4cec-98e5-b667f90b424c</disposition>
<readPoint>
<id>urn:epc:id:sgln:88000269.444.
8b6e0aa4-08fc-4f6f-960d-5a65a748b0e7</id>
</readPoint>
<bizTransactionList> <bizTransaction
type="urn:gs1:epcisapp:farm:pig:status">
urn:gs1:epcisapp:farm:pig:growth</bizTransaction>
</bizTransactionList>
<PF:ID xmlns:PF="urn:gs1:epcisapp:farm:pig">
5af14002-3f27-4989-ae91-3dad4b5c96c**</PF:ID>
<PF:Type xmlns:PF="urn:gs1:epcisapp:farm:pig">
Pig</PF:Type>
<PF:companyId
xmlns:PF="urn:gs1:epcisapp:farm:pig">
8b6e0aa4-08fc-4f6f-960d-5a65a748b0e7
</PF:companyId>
<PF:farmId xmlns:PF="urn:gs1:epcisapp:farm:pig">
9a68ea4e-348e-424e-9346-6e9fefaf18db
</PF:farmId>
<PF:penId xmlns:PF="urn:gs1:epcisapp:farm:pig">
45b01a1c-6fa2-4cec-98e5-b667f90b424c</PF:penId>
<PF:pigId xmlns:PF="urn:gs1:epcisapp:farm:pig">
5af14002-3f27-4989-ae91-3dad4b5c96c900224
</PF:pigId>
<PF:sex xmlns:PF="urn:gs1:epcisapp:farm:pig">
Z</PF:sex>
<PF:growth xmlns:PF="urn:gs1:epcisapp:farm:pig">
26.0</PF:growth>
<PF:weight xmlns:PF="urn:gs1:epcisapp:farm:pig">
139.0</PF:weight>
</ObjectEvent></EventList></resultsBody>
</ns3:QueryResults></EPCISBody>
</EPCISQueryDocumentType>
```


LISTING 2. EPCIS growth event example.

translates the event without capturing it to EPCIS repository. 3) Only Capturing: the Mediation Gateway only captures events without any processing and translations.

The authors used two metrics to evaluate the performance: transactions per second (TPS) and mean response time (MRT). TPS measures how many transactions can be dealt with in a second, whereas MRT measures how fast each request can be executed. The TPS results for the three scenarios are shown in Fig. 10a. Scenario one and two have an average TPS of 26.09 and 27.26 under the environment described in Table 1. Scenario three, instead, has an average TPS of 46.8. This shows that the Mediation Gateway with processing and capturing can achieve 55% of the TPS compared to the Mediation Gateway with only capturing, which indicates that the Mediation Gateway is feasible for IoT systems. Likewise, scenario one and two have an average MRT of 38421ms and 31120ms, respectively. Scenario three has an average MRT of 2340ms. Fig. 10b shows the MRT for the three scenarios.

VI. CONCLUSION

To facilitate the compatibility of IoT platforms, there are currently different efforts in standardizing IoT data and platforms, resulting in two major standards: the NGSI and EPCIS standards. The two standards differ not only in the data encoding but also in the underlying philosophy of representing IoT data; namely, EPCIS is event-based, and NGSI is

entity-based. Entity based models are best to capture snapshot information at a specific time and are preferred by data provides who wants to share real-time entity information. In contrast, the event-based model is best for explicit modeling of system dynamics (entity transformation) and is preferred by data providers interested in entity traceability. This creates fragmentation and makes end-to-end sharing of data burdensome. This work presents an interoperability solution between entity-based information systems (NGSI) and event-based information systems (EPCIS), named a Mediation Gateway, which enhances end to end traceability. It introduces a methodology and implementation of how entity-based models are translated into event-based models.

To prove the concept, the proposed Mediation Gateway is applied to the real-life IoF2020 pig use case. In addition to the real live data from pig farming, a load test evaluation using high-speed multi-user simulation was done to show the performance of the designed interoperability system when the Mediation Gateway is applied.

The Mediation Gateway has been designed to be easily extendable to other domains, and the FIWARE Foundation has accepted it as one of the FIWARE Enablers. Consequently, the authors keep providing continuous support to the open-source community to extend it to other domains. As future work, as soon as the NGSI-LD Orion Context Broker stable version will be released, the proposed Mediation Gateway will be extended and tested to verify the compatibility with the latest NGSI-LD standard version. We are also committed to provide support, to improve and extend to other domains (delivery robots, medical sector, etc.) through the FIWARE enabler open-source initiative.

APPENDIX A
FIWARE PIG ENTITY EXAMPLE

See Listing 1.

APPENDIX B
EPCIS GROWTH EVENT EXAMPLE

See Listing 2.

ACKNOWLEDGMENT

The ''pig farm management'' use case is a collaboration between ILVO, Links Foundation, Evonik Porphyrio, ZLTO and Vion Food Group. The ''meat transparency and traceability'' use case is a collaboration between Wageningen University and Research, GS1 Germany, Korea Advanced Institute of Science and Technology, and European EPC Competence Center.

REFERENCES

[1] J. M. Talavera, L. E. Tobón, J. A. Gómez, M. A. Culman, J. M. Aranda, D. T. Parra, L. A. Quiroz, A. Hoyos, and L. E. Garreta, ''Review of IoT applications in agro-industrial and environmental fields,'' *Comput. Electron. Agricult.*, vol. 142, pp. 283–297, Nov. 2017.

[2] P. P. Ray, ''A survey on Internet of Things architectures,'' *J. King Saud Univ., Comput. Inf. Sci.*, vol. 30, no. 3, pp. 291–319, Jul. 2018, doi: 10.1016/j.jksuci.2016.10.003.

[3] G. Aloi, G. Caliciuri, G. Fortino, R. Gravina, P. Pace, W. Russo, and C. Savaglio, ''Enabling IoT interoperability through opportunistic smartphone-based mobile gateways,'' *J. Netw. Comput. Appl.*, vol. 81, pp. 74–84, Mar. 2017, doi: 10.1016/j.jnca.2016.10.013.

[4] Z. D. Willams. (2017). IoT platform company list 2017. IoT Analytics. Accessed: Dec. 4, 2019. [Online]. Available: https://iot-analytics.com/iot-platforms-company-list-2017-update/

[5] *Context Information Management (CIM); NGSI-LD API*, Standard ETSI GS CIM 009, Version 1.1.1, 2019, pp. 1–159. [Online]. Available: https://portal.etsi.org/TB/ETSIDeliverableStatus.aspx

[6] GS1. (2016). *EPC Information Services (EPCIS) Standard Version 1.2*. [Online]. Available: https://www.gs1.org/sites/default/files/docs/epc/EPCIS-Standard-1.2-r-2016-09-29.pdf

[7] P. Corista, D. Ferreira, J. Gião, J. Sarraipa, and R. J. Gonçalves, ''An IoT agriculture system using FIWARE,'' in *Proc. IEEE Int. Conf. Eng., Technol. Innov. (ICE/ITMC)*, Jun. 2018, pp. 1–6, doi: 10.1109/ICE.2018.8436381.

[8] M. A. Zamora-Izquierdo, J. Santa, J. A. Martínez, V. Martínez, and A. F. Skarmeta, ''Smart farming IoT platform based on edge and cloud computing,'' *Biosyst. Eng.*, vol. 177, pp. 4–17, Jan. 2019, doi: 10.1016/j.biosystemseng.2018.10.014.

[9] J. A. López-Riquelme, N. Pavón-Pulido, H. Navarro-Hellín, F. Soto-Valles, and R. Torres-Sánchez, ''A software architecture based on FIWARE cloud for precision agriculture,'' *Agricult. Water Manage.*, vol. 183, pp. 123–135, Mar. 2017, doi: 10.1016/j.agwat.2016.10.020.

[10] D. Ferreira, P. Corista, J. Giao, S. Ghimire, J. Sarraipa, and R. Jardim-Goncalves, ''Towards smart agriculture using FIWARE enablers,'' in *Proc. Int. Conf. Eng., Technol. Innov. Manage. Beyond New Challenges, New Approaches (ICE/ITMC)*, Jun. 2017, pp. 1544–1551, doi: 10.1109/ICE.2017.8280066.

[11] GS1. (2014). *METRO GROUP Visibility From Catch to Customer*. [Online]. Available: https://www.gs1.org/docs/casestudies/GS1_Metro_traceability_sustainability_case_study.pdf

[12] IBM. (2020). *IBM Food Trust. A New Era for the World'S Food Supply*. Accessed: Mar. 30, 2020. [Online]. Available: https://www.ibm.com/blockchain/solutions/food-trust

[13] G. Bruno, ''Business process models and entity life cycles,'' *Int. J. Inf. Syst. Project Manage.*, vol. 7, no. 3, pp. 65–77, 2019, doi: 10.12821/ijispm070304.

[14] J. L. C. Sanz, ''Entity-centric operations modeling for business process management—A multidisciplinary review of the state-of-the-art,'' in *Proc. 6th IEEE Int. Symp. Service Oriented Syst. Eng. (SOSE)*, vol. 25330, Dec. 2011, pp. 152–163, doi: 10.1109/SOSE.2011.6139104.

[15] C. J. Rosenquist, ''Entity life cycle models and their applicability to information systems development life cycles: A framework for information systems design and implementation,'' *Comput. J.*, vol. 25, no. 3, pp. 307–315, Aug. 1982, doi: 10.1093/comjnl/25.3.307.

[16] M. Noura, M. Atiquzzaman, and M. Gaedke, ''Interoperability in Internet of Things: Taxonomies and open challenges,'' *Mobile Netw. Appl.*, vol. 24, no. 3, pp. 796–809, Jun. 2019, doi: 10.1007/s11036-018-1089-9.

[17] M. Zdravković, M. Trajanović, J. Sarraipa, R. Jardim-Gonçalves, M. Lezoche, A. Aubry, and H. Panetto, ''Survey of Internet-of-Things platforms,'' in *Proc. 6th Int. Conf. Inf. Soc. Technol.*, vol. 1, 2016, pp. 216–220.

[18] E. Kovacs, M. Bauer, J. Kim, J. Yun, F. L. Gall, and M. Zhao, ''Standards-based worldwide semantic interoperability for IoT,'' *IEEE Commun. Mag.*, vol. 54, no. 12, pp. 40–46, Dec. 2016, doi: 10.1109/MCOM.2016.1600460CM.

[19] J. Kim, J. Yun, S.-C. Choi, D. N. Seed, G. Lu, M. Bauer, A. Al-Hezmi, K. Campowsky, and J. Song, ''Standard-based IoT platforms interworking: Implementation, experiences, and lessons learned,'' *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 48–54, Jul. 2016, doi: 10.1109/MCOM.2016.7514163.

[20] M. Djurica, G. Romano, G. Karagiannis, Y. Lassoued, and G. Solmaz, ''OneM2M-based, open, and interoperability IoT platform for connected automated driving,'' in *Proc. 13th ITS Eur. Congr. Brainport*, Amsterdam, The Netherlands, 2019, pp. 1–11. [Online]. Available: https://autopilot-project.eu/wp-content/uploads/sites/16/2019/05/ITS19-Djurica-oneM2M-Based-Open-and-Interoperability-IoT-Platform-for-Connected-Automated-Driving.pdf

[21] P. Sotres, J. Lanza, L. Sánchez, J. R. Santana, C. López, and L. Muñoz, ''Breaking vendors and city locks through a semantic-enabled global interoperable Internet-of-Things system: A smart parking case,'' *Sensors*, vol. 19, no. 2, p. 229, Jan. 2019, doi: 10.3390/s19020229.

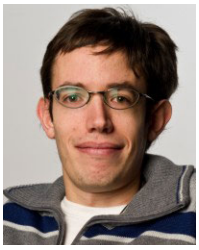
[22] M. Schneider, B. Hippchen, S. Abeck, M. Jacoby, and R. Herzog, "Enabling IoT platform interoperability using a systematic development approach by example," in *Proc. Global Internet Things Summit (GIoTS)*, Jun. 2018, pp. 1–6, doi: 10.1109/GIOTS.2018.8534549.
[23] D. Conzon, M. R. A. Rashid, X. Tao, A. Soriano, R. Nicholson, and E. Ferrera, "BRAIN-IoT: Model-based framework for dependable sensing and actuation in intelligent decentralized IoT systems," in *Proc. 4th Int. Conf. Comput., Commun. Secur. (ICCCS)*, Oct. 2019, pp. 1–8, doi: 10.1109/CCCS.2019.8888136.
[24] J. An, F. L. Gall, J. Kim, J. Yun, J. Hwang, M. Bauer, M. Zhao, and J. Song, "Toward global IoT-enabled smart cities interworking using adaptive semantic adapter," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 5753–5765, Jun. 2019, doi: 10.1109/JIOT.2019.2905275.
[25] S. Overbeek, M. Janssen, and Y. H. Tan, "An ontology-based event-driven architecture for integrating information, processes and services applied to international trade," in *Proc. Int. Conf. Adv. Inf. Syst. Eng.*, in Lecture Notes in Business Information Processing, vol. 112, 2012, pp. 143–158, doi: 10.1007/978-3-642-31069-0_13.
[26] I. Konovalenko and A. Ludwig, "Event processing in supply chain management—The status quo and research outlook," *Comput. Ind.*, vol. 105, pp. 229–249, Feb. 2019, doi: 10.1016/j.compind.2018.12.009.
[27] R. Tommasini, P. Bonte, E. D. Valle, E. Mannens, F. De Turck, and F. Ongenae, "Towards ontology-based event processing," in *Proc. Int. Exper. Directions Workshop OWL*, in Lecture Notes in Computer Science: Including Subseries Lecture Notes Artificial Intelligent Lecture Notes Bioinformatics, vol. 10161, 2017, pp. 115–127, doi: 10.1007/978-3-319-54627-8_9.
[28] Z. Li, C.-H. Chu, W. Yao, and R. A. Behr, "Ontology-driven event detection and indexing in smart spaces," in *Proc. IEEE 4th Int. Conf. Semantic Comput. (ICSC)*, Sep. 2010, pp. 285–292, doi: 10.1109/ICSC.2010.63.
[29] European Commission. (2020). *Future Internet Public-Private Partnership Programme (FI-PPP)*. Accessed: Mar. 30, 2020. [Online]. Available: https://www.fi-ppp.eu/
[30] ETSI. (2019). *Context Information Management (CIM); NGSI-LD API*. [Online]. Available: https://www.etsi.org/deliver/etsi_gs/CIM/001_099/009/01.01.01_60/gs_CIM009v010101p.pdf
[31] FIWARE. (2020). *FIWARE-NGSI V2 Specification*. Accessed: Mar. 30, 2020. [Online]. Available: https://fiware.github.io/specifications/ngsiv2/stable/
[32] L. Ora and S. Ralph. (1999). Resource description framework(RDF) model and syntax specification. World Wide Web Consortium. [Online]. Available: http://www.w3.org/TR/REC-rdf-syntax/
[33] FIWARE. (2020). *FIWARE Orion Context Broker*. Accessed: Apr. 4, 2020. [Online]. Available: https://fiware-orion.readthedocs.io/en/master/
[34] CEF Digital. *Context Broker: Make Data-Driven Decisions in Real Time, at the Right Time*. Accessed: Mar. 30, 2020. [Online]. Available: https://ec.europa.eu/cefdigital/wiki/display/CEFDIGITAL/Context+Broker
[35] *FIWARE-NGSI V2 Specification*, FIWARE, Berlin, Germany, 2020.
[36] *Core Business Vocabulary Standard Version 1.2.1 Specification*, CBV, Toronto, ON, Canada, 2010.
[37] GS1. (2020). *Software Certification Program*. Accessed: Mar. 30, 2020. [Online]. Available: https://www.gs1.org/standards/epc-rfid/software-certification-program
[38] J. Byun, S. Woo, Y. Tolcha, and D. Kim, "Oliot EPCIS: Engineering a Web information system complying with EPC information services standard towards the Internet of Things," *Comput. Ind.*, vol. 94, pp. 82–97, Jan. 2018, doi: 10.1016/j.compind.2017.10.004.
[39] C. Floerkemeier, C. Roduner, and M. Lampe, "RFID application development with the accada middleware platform," *IEEE Syst. J.*, vol. 1, no. 2, pp. 82–94, Dec. 2007, doi: 10.1109/JSYST.2007.909778.
[40] FIWARE. *FIWARE Catalogue of Generic Enablers*. Accessed: Dec. 10, 2020. [Online]. Available: https://www.fiware.org/developers/catalogue/
[41] *FIWARE Orion Context Broker*, FIWARE, Berlin, Germany, 2020.
[42] EECC. (2020). *EPCAT*. Accessed: Jun. 9, 2020. [Online]. Available: https://www.eecc.info/epcat.html
[43] K. A. Robinson, "Entity/event data modelling method," *Comput. J.*, vol. 22, no. 3, pp. 270–281, 1979, doi: 10.1093/comjnl/22.3.270.
[44] B.-D. Paul, *Information Systems Development*, 3rd ed. Basingstoke, U.K.: Macmillan Press, 1998.
[45] IoF. (2020). *Internet of Food & Farming 2020*. [Online]. Available: http://www.iof2020.eu
[46] J. Maselyne, W. Saeys, B. De Ketelaere, K. Mertens, J. Vangeyte, E. F. Hessel, S. Millet, and A. Van Nuffel, "Validation of a high frequency radio frequency identification (HF RFID) system for registering feeding patterns of growing-finishing pigs," *Comput. Electron. Agricult.*, vol. 102, pp. 10–18, Mar. 2014, doi: 10.1016/j.compag.2013.12.015.
[47] J. Maselyne, I. Adriaens, T. Huybrechts, B. De Ketelaere, S. Millet, J. Vangeyte, A. Van Nuffel, and W. Saeys, "Measuring the drinking behaviour of individual pigs housed in group using radio frequency identification (RFID)," *Animal*, vol. 10, no. 9, pp. 1557–1566, 2016, doi: 10.1017/S1751731115000774.
[48] GS1. (2014). *EPC Tag Data Standard Version 1.9 Specification*. [Online]. Available: http://www.gs1.org/gsmp/kc/epcglobal/tds/
[49] GS1 Germany. (2020). *fTrace*. Accessed: Jun. 9, 2020. [Online]. Available: https://www.ftrace.com/en/gb

YALEW TOLCHA received the B.Sc. degree in electrical and computer engineering from the Addis Ababa Institute of Technology, Ethiopia, in 2011, and the M.Sc. degree in computer science from the Korea Advanced Institute of Science and Technology, in 2016, where he is currently pursuing the Ph.D. degree with the School of Computing. From 2011 to 2014, he has worked as an Assistant Lecturer with the Addis Ababa Institute of Technology. His research interests include the IoT, big data, machine learning, and artificial intelligence.

AYALEW KASSAHUN received the B.Sc. degree in civil engineering from the Addis Ababa University, in 1991, the M.Sc. degree in hydrology and hydrological modeling from Dar es Salaam University, in 1993, and the M.Sc. degree in environmental sciences and modeling and the Ph.D. degree in knowledge systems from Wageningen University, in 1996 and 2017, respectively. From 1997 to 2002, he worked as a Software Engineer with Infor (former Baan) and Inforay (currently part of SDB Group). He currently works as a Researcher and a Lecturer with Wageningen University. He has extensive research experience and published many scientific articles in several interdisciplinary fields, including transparency and traceability systems, agri-food supply chain management, management information systems, environmental modeling, and machine learning.

TEODORO MONTANARO (Member, IEEE) received the M.S. degree in computer engineering and the Ph.D. degree in computer and control engineering from the Politecnico di Torino, Turin, Italy, in 2014 and 2018, respectively.

Since 2017, he has been collaborating with the LINKS Foundation, by cooperating to different researches funded by the European commissions, like IoF2020, S4G, MONICA, and MAESTRI, that brought innovations in different fields, such as food and farm, grid, city, and health. He has authored different papers on international journals and conferences. His current research interests include the IoT applications specifically focused on the exploitation of fog computing, DLT, blockchain, and AI in different domains, such as smart grids, smart homes, smart cities, industrial processes, food traceability, and smart health.



DAVIDE CONZON received the B.S. and M.S. degrees in computer engineering from the Politecnico di Torino, in 2005 and 2007, respectively.

He is currently working with LINKS, leading the Architectures and Interoperability Solutions Team in the IoT and pervasive technologies research area. He participated in several national research and development projects, including EU funded research and development projects, where he has designed and developed platforms for the easy integration and virtualization of devices and robotics simulation tools. He is the coauthor of several conference papers related to this topic. His main research interest includes the Internet of Things infrastructure and applications. Since April 2015, he has been a member of the XMPP Standard Foundation.

GEORG SCHWERING received the Diploma degree in chemistry from the University of Münster, in 1998, and the Ph.D. degree in physical chemistry from the Max Planck Institute for Solid State Research, in 2003.

From 2005 to 2016, he worked in various divisions of the international wholesaler METRO Group, focusing on the implementation of RFID and the IoT technology, EPCIS-based traceability solutions and cross-partner data exchange, and mobile applications. Since 2017, he has been the Head of Scientific Research Projects with the European EPC Competence Center (EECC) at Neuss and Cologne, Germany. His research interest includes the development of solutions for full traceability of things and services in various industries and the exchange of data using standardized interfaces. He is always on the lookout for new IoT solutions to make things and processes self-managing and self-optimizing.

JARISSA MASELYNE received the Ph.D. degree in bioscience engineering from KU Leuven. Her Ph.D. Topic was automated monitoring of feeding and drinking patterns in growing-finishing pigs: towards a warning system for performance, health and welfare in individual pigs. She is currently working as a Researcher with the Group of Agricultural Engineering, Flanders Research Institute for Agriculture, Fisheries and Food (ILVO), with a main focus on precision pig farming. She is currently the Vice-President of the Precision Livestock Farming Committee, EAAP. She is also an Electromechanical Engineer. She is involved in various European projects, as a WP (co-)lead or task lead, on the topics of the Internet of Things (IOF2020), high performance computing (CYBELE), and renewable energy systems (RES4LIVE) for livestock farming. She is the (co)author of several articles in the field of precision livestock farming.

DAEYOUNG KIM received the B.S. and M.S. degrees in computer science from Pusan National University, South Korea, in 1990 and 1992, respectively, and the Ph.D. degree in computer engineering from the University of Florida, in August 2001. From 1992 to 1997, he worked as a Research Staff Member with ETRI, South Korea. From September 2001 to 2002, he was a Research Assistant Professor with Arizona State University. From 2002 to 2014, he was an Assistant/Associate Professor with the Department of Computer Science, KAIST, South Korea, where he has been a Professor with the School of Computing, since March 2014. He is currently the Director of the Auto-ID Laboratory, South Korea and the Global USN National Research Laboratory. His research interests include sensor networks, real-time and embedded systems, and the Internet of Things.