1.1 LW-COEDGE AND FIWARE DATA MODEL

In order to enable the operation between LW-CoEdge and FIWARE, it is necessary to establish the relationship between their models. Initially, we describe the services and devices models of the FIWARE. Next, we present the relationship model.

The schemas to describe the services and devices are provided by the GE IoT Agent[1,2]. Such models are used to enable the FIWARE to interact with *end devices* for receiving the observations (raw data) and sending actuation commands. In order to facilitate the understanding of these schemas, we present them using a high-level representation. Service (see Figure 1) is the higher-level information of FIWARE. The Service model describes the essentials information to register a service that must be used by the device during the operation. It is worth mentioning that a device can use several registered services. Table 1 depicts the attributes.

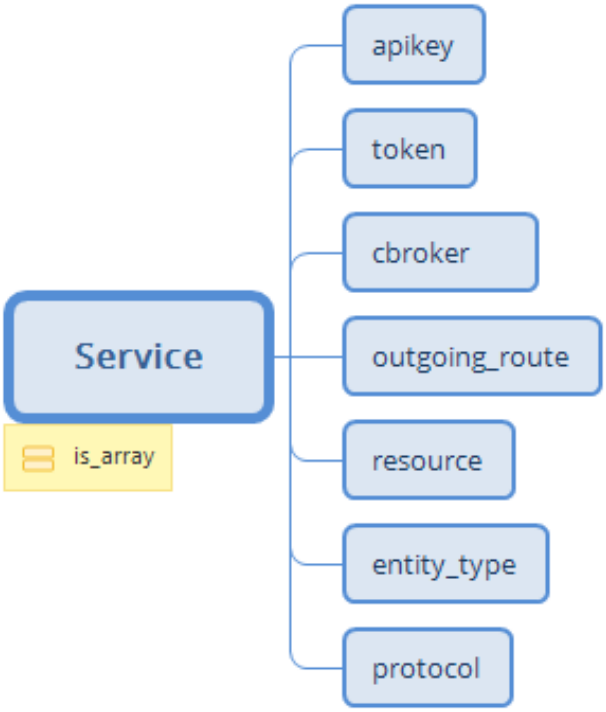

Figure 1. FIWARE GE IoT Agent Service model

[1] https://github.com/telefonicaid/iotagent-node-lib
[2] https://telefonicaiotiotagents.docs.apiary.io/#reference/configuration-api

Table 1. Service model attributes

Attribute	Description
Apikey	It identifies the service key which the devices use to send data (observations) or receive actuation command.
Token	A token used to verify the identity of IoT Agent when the authentication/authorization system is active.
Cbroker	URL to the GE Orion Broker.
outgoing_route	It is an optional field provided to identify the VPN/GRE tunnel used by the device.
entity_type	Entity type used in entity publication.
resource	It is a URI provides by IoTAgent and used by the device to send information when the communication protocol is HTTP.

Device (Figure 2) describes the essentials attributes to register a device and the information provided by it. An example of information is the temperature of a specific place. Moreover, the model also defines an optional attribute to specify the actuation command. Table 2 depicts the attributes.

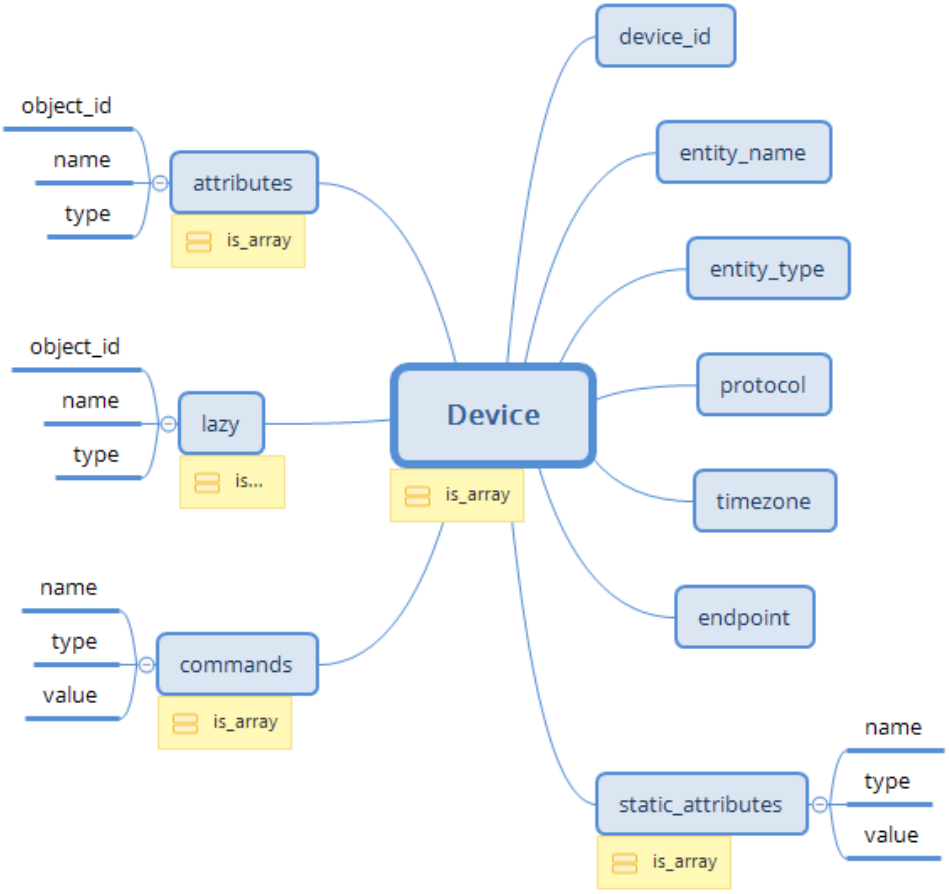

Figure 2. FIWARE GE IoT Agent Device model

In the model, we can observe that *attributes*, and *lazy* represent an object containing the *object_id*, *name*, and *type* attributes. The *object_id* is an optional attribute and represents the internal attribute name of the device. The *name* is a mandatory attribute that represents the name of the *object_id* of the target entity in the **GE Orion Broker**. Lastly, the *type* is a mandatory attribute that represents the type of the attribute in the target entity.

Table 2. Device model attributes

Attribute	**Description**
device_id	Unique device identifier (mandatory).
protocol	The protocol assigned to the device to communicate with the Platform (mandatory).
entity_name	The ID used to identify the entity at the GE Orion Broker.
entity_type	Type of the entity that will represent the device in the Context Broker.
timezone	Timezone for the device.
endpoint	URL for the device to receive commands.
attributes	List of active attributes of the entity that represents the device (populate via push).
Lazy	List of lazy attributes of the entity that represents the device (it is populated just if requested).
commands	List of commands supported by the device.
static_attributes	List of static attributes for attaching the entity in every observation sent by the devices.

Regarding the joint operation between LW-CoEdge and FIWARE, another critical question is "***How the VN is related to the GE IoT Agent device models for acquiring the data or performing actuation?***" To achieve this goal, we modeled the relationship (Figure 3) between the LW-CoEdge Datatype descriptor model (Figure 4) and the FIWARE device model (Figure 2). This relationship is defined as a tuple $contains = (R_a, R_b)$, where $R_a = element_i | \ I \geq 1$ represents an element that belongs to the array of elements of the Datatype descriptor; $R_b = device\_id$ represents the unique Device Identifier.

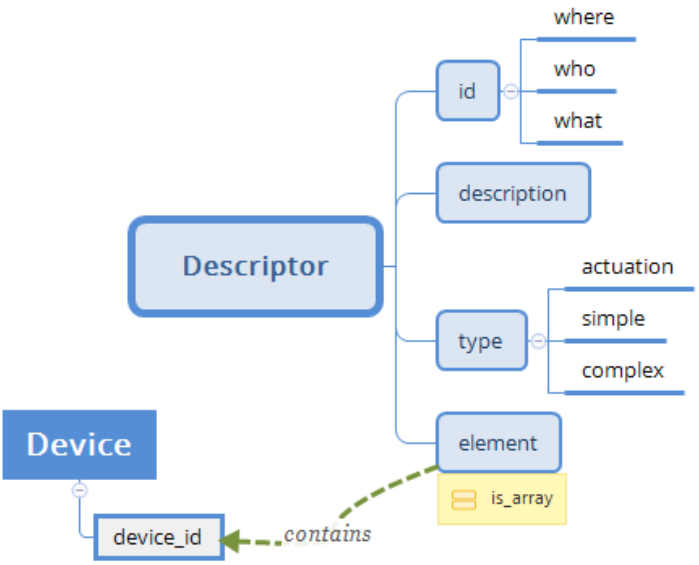

Figure 3. The relationship between Datatype descriptor and Device

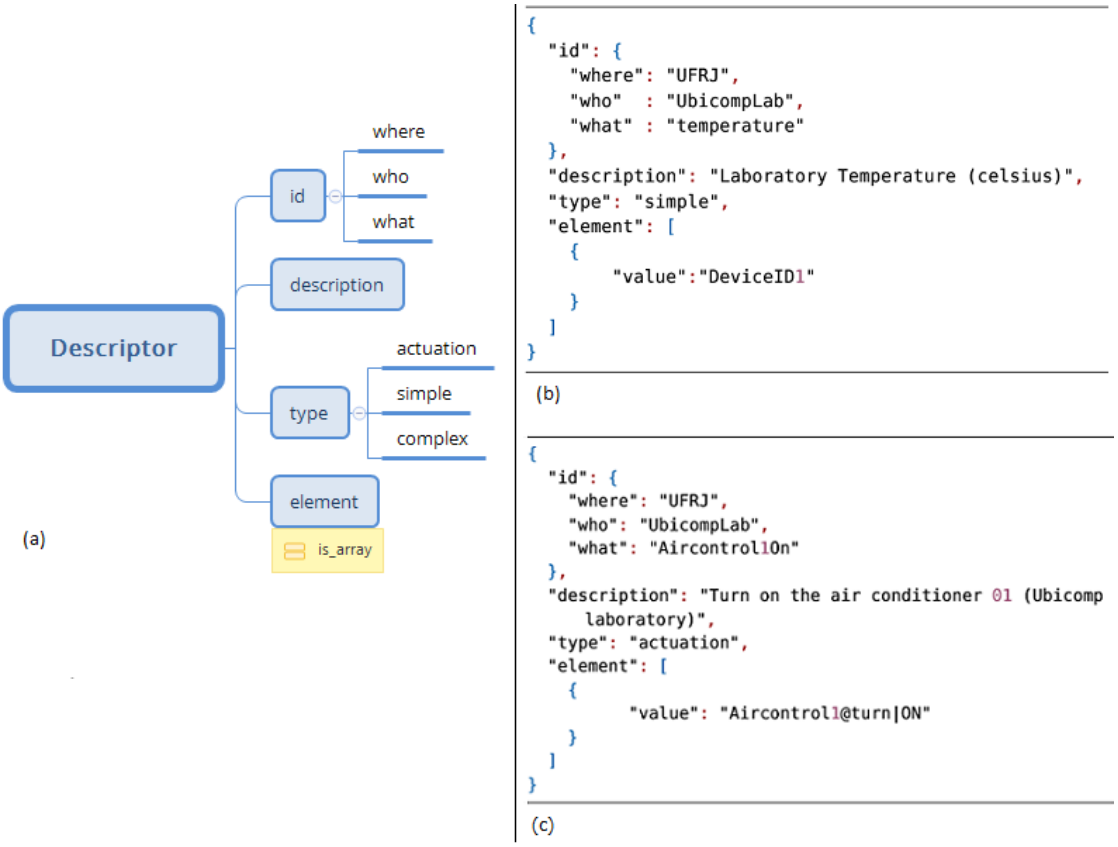

Figure 4. (a) The representation of the datatype descriptor. (b) Datatype descriptor example for a simple datatype. (c) Datatype descriptor example for an actuation