

# Virtual Device Model extending NGSI-LD for FaaS at the Edge

Francesco Martella<sup>\*†</sup>, Giovanni Parrino<sup>§</sup>, Giuseppe Ciulla<sup>‡</sup>, Roberto Di Bernardo<sup>‡</sup>,  
Antonio Celesti<sup>\*¶</sup>, Maria Fazio<sup>\*¶</sup>, Massimo Villari<sup>\*†</sup>

<sup>\*</sup> Department MIFT, University of Messina, Italy - {gioparrino, fmartella, acelesti, mfazio, mvillari}@unime.it

<sup>†</sup> ALMA Digit S.R.L., Messina, Italy - {f.martella, m.villari}@almadigit.com

<sup>‡</sup> Research & Development Laboratory, Engineering Ingegneria Informatica, Palermo, Italy -  
{giuseppe.ciulla, roberto.dibernardo}@eng.it

<sup>§</sup> Engineer, Italy - {nanni.parrino@gmail.com}

<sup>¶</sup> Gruppo Nazionale per il Calcolo Scientifico (GNCS) - INdAM, Rome, Italy

**Abstract**—Smart environments are composed of an ever-increasing number of heterogeneous resources and devices for collecting and processing of a large amount of context data. These activities can be performed at the edge of the smart area, over a distributed and heterogeneous infrastructure, so to be close to the end-user and optimize response time. However, it is hard to define a data model able to support data exchange among different systems and between systems and users. This paper presents the key features of a smart environment and introduces the concept of virtual device, that is an abstracted component characterized by specific high-level functionalities. Then, the paper proposes a data model useful to represent and optimize the adoption of virtual device in smart environments. To better explain the data model features and benefits, we refer to a video surveillance use case, where a smart camera is able to provide the solid angle detection as a service.

**Index Terms**—Edge computing, NGSI-LD, IoT Cloud, Data Modeling, Virtual Device, Smart environments.

## I. INTRODUCTION

Smart environments are challenging contexts to offer a new generation of services on the basis of huge amount of gathered data and pervasive processing. In particular, distributed Edge computing solutions move data processing closer to end-users, handling data in real time, providing local feedback and adding robustness to connectivity. Many technologies can be exploited in smart environments for implementing Edge computing solutions, interconnecting different hardware and software components and deploying services and applications. This high heterogeneity of available resources need to be represented in an abstracted way, in order to provide a useful solution for the high level management of both simple and complex smart environments. One of the key challenges of Edge computing concerns the exchange of data between the various actors of the smart ecosystem. In this field, FIWARE NGSI [1] provides a standardized approach for the exchange of context information. NGSI-LD is an information model that allows applications to perform dynamic and flexible discovery and query of data, also getting information on the related context, such as the period of validity, geographic constraints, and other semantically important information.

Even if NGSI-LD is valuable for retrieval and interpret data, it cannot represent how, where and when data has to be managed and processed. Today we identify a limit in the existing data models, that characterize only the data and the data source, but not the services that allow data to be acquired or managed. Describing smart systems and possible changes in their behaviour is a big challenge that could simplify the deployment of services and applications in the same environment. In this paper, we introduce the innovative concept of Virtual Device, an abstracted component able to describe the behaviour of a computing node from the point of view of its potential in processing functionalities. To deal with the deployment of Virtual Device, an extension of the NGSI-LD standard is proposed in order to enrich data with related processing information. Our approach makes it possible to create geolocalized data models and provides designers and developers high flexibility in the characterization of a service deployment system based on the FaaS technology [2]. We show how it is possible to deploy multiple Virtual Devices on the same physical device simply by configuring the related data model. Finally, the proposed approach is applied to a specific use case by introducing the concept of solid angle in the video surveillance applications of a Smart City. The remainder of the paper is organized as follows. Related works and differences with the proposed study are discussed in Section II. A brief description of the NGSI-LD standard is reported in the Section III. In Section IV the concept of Virtual Device and the extension of NGSI-LD standard are discussed. A use case of the proposed data model is shown in Section V. Final remarks and possible future developments are summarized in Section VI.

## II. RELATED WORK

An interesting survey on Edge computing opportunities for smart cities has been carried out in [3], where authors highlight the role of Edge computing to realize the vision of smart cities with the objective to classify the literature by devising a comprehensive and meticulous taxonomy. They identify and discuss key requirements, and enumerate recently reported

synergies of edge computing enabled smart cities. Finally, several indispensable open challenges along with their causes and guidelines are discussed, serving as future research directions. Intelligent offloading for collaborative smart services in Edge computing [4] claims the weakness of long service response time and low QoS in scenarios with clouds. The authors remark that edge computing is nowadays integrated with the smart city to promote the inherent shortcomings of terminals in cities, to this they designed an intelligent offloading method for collaborative smart city services, named IOM. They try to achieve a trade-off among minimizing service response time, optimizing energy consumption and maintaining load balance while guaranteeing the privacy preservation during service offloading. A comprehensive and good analysis with mathematical models has been done.

Given the ever-increasing need to send and receive information from collection devices (sensors, cameras, etc.) to cloud computing devices [5] [6], the risk of increasing the percentage of errors during data transmission and packet loss is inevitable, especially in cloud and edge-based network architectures. Studies [6] and [5] aim at reducing the computational load of cloud devices to redistribute it to other hardware, according to performance improvement logics, and migrating the same load to computationally less powerful but very useful devices, positioned as interface infrastructures between field and cloud devices, the so-called edge devices. The possible computations in these devices concern a data pretreatment [7] [6] consisting of an encapsulation of the information within a pre-packaged data-model that allows a better management and analysis, a "filtering" to skim the number of data to travel on the network and reduce the number of consignments, which could lead to an improvement in terms of reduction of transmission errors and longer battery life in the case of devices not powered by current (energy consumption). With regard to the computational capabilities of edge devices, [8] highlights the possibility of obtaining, in addition to the "raw" data, also "indirect" information extrapolated from the individual measurements (in the case of sensors) fed into special statistical calculations or "predictions/predictions" made on dynamic information. [9] instead highlights the great capacity of job sorting within a network made up of cloud and edge devices, which can be managed and reconfigured through JSON messages and with a micro-service oriented architecture. In [10] it is highlighted how necessary a data model that offers advanced functionality for the description of the context can exploit different data sources. However, the data can be used in various contexts for different solutions. The authors highlight how FIWARE technologies help the scientific community and developers through the data models defined by ETSI ISG Context Information Management (ETSI CIM). The work concludes by highlighting how the development of IoT technologies is driven by semantic and context-sensitive data models. From the considerations carried out it emerges that interoperability and models with semantic annotations significantly increase the reusability of IoT resources outside their initial specificity.

The growing use of IoT device applications in smart envi-

ronments, associated with a greater demand for computational challenges, today sees cloud infrastructures increasingly used and with the need to rationalize resources. Regarding this new need, in [11] it is highlighted how it is possible to use customized generic Edge devices to carry out multiple activities simultaneously can be a solution to lighten the work of cloud infrastructures. The authors have implemented and tested, in a real solution in the city of Messina (Italy), a solution based on the Function as a Service (FaaS) paradigm. The proposed solution allows users to perform multiple activities on the same device such as vehicle counting, license plate recognition, identification objects etc. In this case, two cameras were connected to a Raspberry PI 4 and the performance compared. Nothing prevents you from connecting different sensors to the Edge device and imagining each sensor as a different service. Each service can be managed through the concept of Virtual Device with the use of specific Data Model. Furthermore, it is possible to imagine not only a single Virtual Device, but also a Virtual Device network [12]. A Virtual Device network is composed by services deployed on different Edge devices. Each device in the network with a specific service can work with the others in order to perform complex processing for services in a given area.

In our work we want to put together the concepts analyzed so far in order to formulate a data-model that, through the architectural paradigm of micro-services, allows to simply manage the configuration and display of field devices, the reduction of the amount of data transmitted, energy consumed and transmission errors, offering more services resulting from the installation of a single device (sensor, camera or IoT), a concept that we will present under the name of virtual-device.

### III. WHAT IS NGSI-LD

NGSI-LD is an open standard for managing context information. NGSI-LD was released in 2018 as an ETSI specification to enhance FIWARE's NGSIv2 standard. The improvement that led to the development of NGSI-LD concerns the improved support of linked data and the definition of properties and semantics structured according to the JSON-LD standard. NGSI-LD is based on the key concepts of: Entity, Property and Relationship. Entities are objects with specific properties and can be in relationship with other Entities. Properties are a combinations of attributes in the form (*key : value*). Relationships allow to establish logic connection instances through linked data using a property that points to another external resource (identified by a Uniform Resource Identifier (URI)). This characteristic is derived from JSON-LD and allows to define and connect entities in a clear and unique way. Properties and relationships can be part of other properties or relationships along a maximum of 1 or 2 levels of concatenation in NGSI-LD schemas. Properties characterize the entities involved and their interpretation is defined in the *@context* component. This feature allows the generalization of the data model and marks the difference between the NGSIv2 and NGSI-LD formats. Only some properties are standardized so to be implicitly defined and not contained in *@context*. To better explain the

key features of NGSI-LD, in Listing 1, we show a well-known example of a NGSI-LD data model for a sensing device [13].

```

1 {  "id": "urn:ngsi-ld:Device:device-9845A",
2    "type": "Device",
3    "category": {
4      "type": "Property",
5      "value": ["sensor"]
6    },
7    "batteryLevel": {
8      "type": "Property",
9      "value": 0.75
10   },
11   "dateFirstUsed": {
12     "type": "Property",
13     "value": {
14       "@type": "DateTime",
15       "@value": "2014-09-11T11:00:00Z"
16     }
17   },
18   "controlledAsset": {
19     "type": "Relationship",
20     "object": ["urn:ngsi-ld:wastecontainer-Osuna-100"]
21   },
22   "value": {
23     "type": "Property",
24     "value": "1%3D0.22%3Bt%3D21.2"
25   },
26   "refDeviceModel": {
27     "type": "Relationship",
28     "object": "urn:ngsi-ld:DeviceModel:myDevice-wastecontainer-sensor-345"
29   },
30   "rssi": {
31     "type": "Property",
32     "value": 0.86
33   },
34   "controlledProperty": {
35     "type": "Property",
36     "value": ["fillingLevel", "temperature"]
37   },
38   "owner": {
39     "type": "Property",
40     "value": ["http://person.org/leon"]
41   },
42   "deviceState": {
43     "type": "Property",
44     "value": "ok"
45   },
46   "@context": [

```

```

"https://schema.lab.fiware.org/ld/context",
"https://uri.etsi.org/ngsi-ld/v1/ngsi-ld-core-context.jsonld"
]
}

```

Listing 1. Example of NGSI-LD data format

The *id* attribute is an Uniform Resource Name (URN) that can be used by other entities to link this one; the attribute *type* clarifies what kind of entity it is. The *category* attribute specifies the assets of the attribute type. *ControlledAsset* and *refDeviceModel* link the entity with other entities, whereas the remaining attributes represent the current values (e.g., )payload) in the data model. Their structure, if not standardized, is defined in the *@context* attribute.

#### IV. THE NGSI-LD EXTENSION FOR REPRESENTING A VIRTUAL DEVICE MODEL

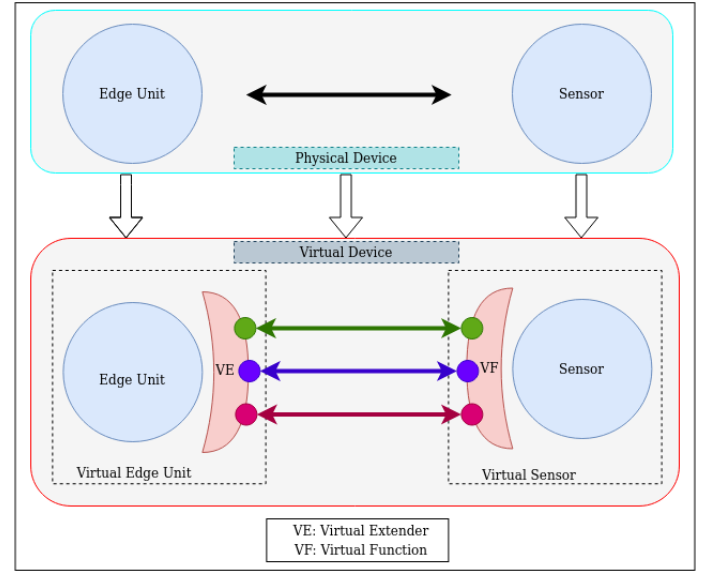


Fig. 1. Abstracting schema of a Virtual Device

A Virtual Device is an abstraction of a physical device that emphasize specific features or functionalities of the device itself. This concept is outlined in the Fig. 1: the Edge Unit is a physical device at the edge and it can be connected to a sensor (or more sensors) for gathering (and after processing) data from the environment, such as recording the measurement of temperature at specific time intervals. With this approach, data collection and processing tasks are strictly related to the type of binding between the Edge Unit and the sensor. A Virtual Device maps several behaviours of the Virtual Edge Unit with the ones of the Virtual Sensor. As shown in the lowest part of Fig. 1, the Virtual Device allows to decouple different possible activities executed into the Edge Unit by using data coming from the sensor, such as estimating the average temperature, identifying peaks in temperature measurements, and so on,

thus increasing the flexibility in data processing. In particular, through a Virtual Extender (VE), it is possible to connect N-functions (coloured dots in Fig. 1) in the Virtual Edge Unit with the same amount of functions in the Virtual Sensor. In this way, it is like N-different physical devices are available. The VE together with the Edge Unit identifies the Virtual Edge Unit, which communicates with the Virtual Sensor by means of the Virtual Function (VF) component. The VF together with the Physical Sensor define the concept of Virtual Sensor. The Virtual Edge Unit and the Virtual Sensor compose the Virtual Device.

Considering the capabilities reached by current Edge computing devices, it is possible to image several independent processes running at the Edge that elaborate data in a different ways. This entails a considerable economic advantage in the deployment of Edge solutions on a large scale, but also a considerable advantage in terms of environmental protection. In Figure 2 a schematic example is shown.

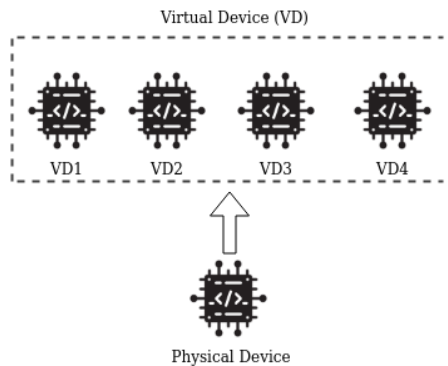


Fig. 2. General representation of a Virtual Device

The physical device can be, for example, a micro-controller with a temperature sensor attached. The Virtual Device VD1 can provide data in real time, the Virtual Device VD2 the daily average temperature, the Virtual Device VD3 the weekly average temperature and the Virtual Device VD4 the maximum weekly temperature value. The limit of the number of virtual devices depends only by the processing limits of the micro-controller. Applications asking for the service will see a dedicated service, but, in reality, they will access information processed at the Edge. The request and the access to the service from an application need to be defined and managed through a data model.

To include the concept of Virtual Device within the NGSI-LD data model, it is possible to create an extension of the standard that abstracts the concept of physical device. The extension is accomplished by adding information to the payload of the model, thus to have a more flexible data model.

```
1 {
2   "id": "urn:ngsi-ld:Device:device-9845",
3   "type": "Device",
4   "category": {
```

```
    "type": "Property",
    "value": ["sensor"]
  },
  "batteryLevel": {
    "type": "Property",
    "value": 0.75
  },
  "deploymentInfo": {
    "version": 1.0,
    "provider": {
      "name": "openfaas",
      "gateway": "http://127.0.0.1:8222"
    }
  },
  "type": "functions",
  "datafunctions-arm": {
    "lang": "python3-flask-debian",
    "handler": "./datafunctions-arm",
    "image": "urbanite-messina/datafunctions-arm:latest",
    "environment": {
      "physical_dimension": "meteo",
      "measurement": "Temp",
      "operation": "deploy",
      "date": "2018-04-06T11:00:00Z",
      "place": "S.Marco",
      "read_timeout": "50s",
      "write_timeout": "50s",
      "upstream_timeout": "50s",
      "exec_timeout": "50s"
    }
  },
  "dateFirstUsed": {
    "type": "Property",
    "value": {
      "@type": "DateTime",
      "@value": "2014-09-11T11:00:00Z"
    }
  },
  "value": {
    "type": "Property",
    "value": "22.3"
  },
  "refDeviceModel": {
    "type": "Relationship",
    "object": "urn:ngsi-ld:DeviceModel:urbanite-device-sensor-temp"
  },
}
```

```

51  "rssi": {
52      "type": "Property",
53      "value": 0.86
54  },
55  "controlledProperty": {
56      "type": "Property",
57      "value": ["fillingLevel", "
                    temperature"]
58  },
59  "owner": {
60      "type": "Property",
61      "value": ["http://fcrlab.unime.it
                    "]
62  },
63  "deviceState": {
64      "type": "Property",
65      "value": "ok"
66  },
67  "location": {
68      "type": "GeoProperty",
69      "value": {
70          "type": "Point",
71          "coordinates": [38.1885046, 15
                          .5535013]
72      },
73  "@context": [
74      "https://schema.lab.fiware.org/ld
          /context",
75      "https://uri.etsi.org/ngsi-ld/v1/
          ngsi-ld-core-context.jsonld",
76      "fake:https://fcrlab.unime.it/
          urbanite-messina"
77  ]
78 }

```

Listing 2. Extension of NGSI-LD Physical Device data format

In Listing 2, an example of the extension of NGSI-LD is shown in the case of an Edge device with the ability to support Virtual Device. The device is linked to a specific model by the *refDeviceModel* property, but thanks to the *deploymentInfo* property it can be set to host Virtual Device. The *deploymentInfo* property is read and converted into a file yml which is used to call the openFaas function which deploys or starts the service as in this case. Each field of the *deploymentInfo* attribute will then be defined in *@context*. The location attribute determines the location of the physical device.

```

1 {
2   "id": "urn:ngsi-ld:Device:device-9845
      A-Pugliatti-1",
3   "type": "VirtualDevice",
4   "category": {
5       "type": "Property",
6       "value": ["Virtualsensor"]
7   },

```

```

8   "deploymentInfo":{
9       "version":1.0,
10      "provider":{
11          "name": "openfaas",
12          "gateway": http://127.0.0.1:8
              222
13      }
14      "type": "functions",
15      "datafunctions-arm":{
16          "lang": "python3-flask-debian
              ",
17          "handler": "./datafunctions-
              arm",
18          "image": "urbanite-messina/
              datafunctions-arm:latest",
19          "environment":{
20              "physical_dimension": "
                  meteo",
21              "measurement": "Temp",
22              "operation": "average",
23              "date": "2018-05-06T11
                  :00:00Z",
24              "place": "S.Marco",
25              "read_timeout": "50s",
26              "write_timeout": "50s",
27              "upstream_timeout": "50s
                  ",
28              "exec_timeout": "50s"
29          }
30      }
31  }
32  "value": {
33      "type": "Property",
34      "value": "21.1"
35  },
36  "refDevice": {
37      "type": "Relationship",
38      "object": "urn:ngsi-ld:Device:
          device-9845A"
39  },
40  "controlledProperty": {
41      "type": "Property",
42      "value": [ "
          DailyAveragetemperature" ]
43  },
44  "deviceState": {
45      "type": "Property",
46      "value": "ok"
47  },
48  "@context": [
49      "https://schema.lab.fiware.org/ld
          /context",
50      "https://uri.etsi.org/ngsi-ld/v1/
          ngsi-ld-core-context.jsonld",
51      "fake:https://fcrlab.unime.it/
          urbanite-messina"

```

Listing 3. Extension of NGSI-LD Virtual Device data format

In Listing 3 an example of Virtual Device deployment is shown. In this case, respect to the Listing 2, it is sufficient to change the value of "operation" in the *infoDeployment* property to deploy a new service that returns the weekly average of the measured temperature. The virtual device is tied to the physical device by the *refDevice* property. By doing this, it is possible to deploy different services simply by changing a field or querying them and defining the various rules in the payload of the proposed NGSI-LD extension. The Virtual Device can be deployed or stopped as needed, but the physical device will always be ready to host new services. It is also possible to think of different types of sensors on each device. Each sensor can then be virtualized by creating heterogeneous Virtual Devices on the same physical device.

#### V. USE CASE: SMART ENVIRONMENT MONITORING WITH A VIRTUAL CAMERA

The use case we analyzed in this paper concerns a video surveillance system in a smart city. Edge devices are designed to offer miscellaneous micro-services; based on the actor specific measurement request in a precise moment a single device can offer one or more aggregated data obtained by different sensors integrated on the same board. The only limit for requests specifications is the physical one. To simplify the concept: an ip-cam can acquire 25 frames per second (physical limit for this device). The Virtual Device could offer to a specific actor 10 frames per second, but it will never can offer 30 frames per second (fps). In order to help this type of requests that can be differentiated but always addressed to the same device, we introduced in Section IV the concept of Virtual Device as an abstraction of the physical device. In particular, in our vision, a Virtual Device is a extension of an NGSI-LD Physical Device definition type in which is introduced the specific configuration (e.g. 5 fps, 10 fps or 25 fps or solid angle). In order to further improve and enrich the data model that we want to propose, it has been thought to add a measure that can give an account of the angle and the surface that can be covered by the "framing" of a device that can acquire images from a given scene. This measurement has been identified with the *solid angle*.

Before arriving at a more precise description of the solid angle, it is necessary to make a premise by starting to describe what is called *Field of View*. The Field of View (**FoV**) is a measure of the observable world that can be seen at a given time. In the case of optical instruments or sensors it can be assimilated to the solid angle through which a detector is sensitive to magnetic radiation at a given time. Another well-known element in the field of photography is what is usually called the *Angle of View* (**AoV**) which describes the angular range of a given scene framed by a camera. The term angle of view can usually be assimilated with the more general term Field of View.

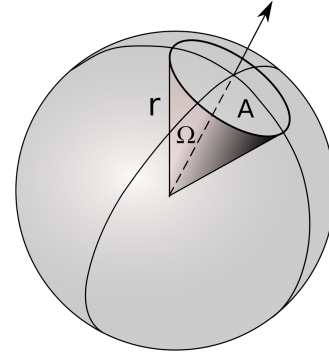


Fig. 3. Solid Angle representation ("http://pngegg.com")

Returning to the solid angle, we can define it as the extension in three-dimensional space of the plane angle. The unit of measurement of the solid angle is the *Steradian* and can be calculated as:

$$\Omega = A/R^2(\text{SolidAngle})$$

In which: **A** is the area of the spherical portion of radius **R** seen under the angle.

As we can see from the previous formula we can note how the ratio, even in the three dimensions, between the portion of circumference, the radius and the subtended angle is maintained. As in the planar angle, the solid angle is the ratio between the area of the spherical surface and the radius of the sphere considered.

We can further determine the relationship between the two corners:

$$d\Omega = (2\pi \sin\Theta)d\Theta$$

where  $\Omega$  is the solid angle and  $\Theta$  is the plane angle.

To get a clearer picture of the formulas expressed, let's try to imagine a light bulb in the centre of a sphere. For the whole sphere the solid angle through which the light rays pass is valid:

$$\Omega = S/R^2 = (4\pi R^2)/R^2 = 4\pi$$

If, instead of considering the whole sphere, we consider the only part of the spherical surface crossed by the light rays, using the differentials we obtain:

$$d\Omega = dS/(R^2) = (R^2 \sin\theta d\theta d\phi)/R^2 = \sin\theta d\theta d\phi$$

where  $\theta$  is the *colatitude* (angle from the north pole) and  $\phi$  is the *longitude*.

This value can therefore be useful to understand how large the portion of surface that a given camera can be able to frame given a certain angle. The solid angle in this case is a specific service deployed as Virtual Device. The solid angle calculated by taking as reference the point where the device is located and can be useful in a video surveillance service (Fig. 4).



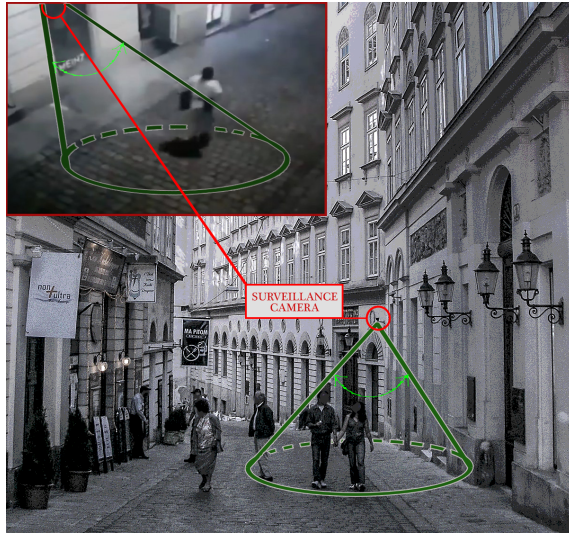


Fig. 4. Solid Angle use in real scenario (top-left: "Vienna 2020, photo from the web", main: "Vienna's imagine of repertory published on wikipedia.com")

#### A. Virtual Device enriched data model

Going forward to the concepts of Virtual Device, Physical Device and the data model that we want to propose in order to realize micro-services on the Edge, we want to start to introduce the above mentioned concept of solid angle.

We can so describe the Virtual Device we want to define in our data model as shown in Listing 4.

```

1 {
2   "id": "urn:ngsi-ld:Device:device-9845
   A-Pugliatti-2",
3   "type": "VirtualDevice",
4   "category": {
5     "type": "Streaming",
6     "value": ["5", "10", "25", "
       solidAngle"]
7   },
8   "deploymentInfo": {
9     "version": 1.0,
10    "provider": {
11      "name": "openfaas",
12      "gateway": http://127.0.0.1:8
        222
13    }
14    "type": "functions",
15    "datafunctions-arm": {
16      "lang": "python3-flask-debian
        ",
17      "handler": "./datafunctions-
        arm",
18      "image": "urbanite-messina/
        datafunctions-arm:latest",
19      "environment": {
20        "operation": "solidAngle
        ",

```

```

21     "date": "2018-05-06T11
        :00:00Z",
22     "place": "P.Pugliatti"
23   }
24 },
25 },
26 },
27 "refDevice": {
28   "type": "Relationship",
29   "object": "urn:ngsi-ld:Device:
        device-9845A"
30 },
31 "deviceState": {
32   "type": "Property",
33   "value": "ok"
34 },
35 "@context": [
36   "https://schema.lab.fiware.org/ld
        /context",
37   "https://uri.etsi.org/ngsi-ld/v1/
        ngsi-ld-core-context.jsonld",
38   "fake:https://fcrlab.unime.it/
        urbanite-messina"
39 ]
40 }
41 }

```

Listing 4. NGSI-LD data format for a Virtual Device

The configuration, and the relative micro-service, can be changed and deployed in our Virtual Device, simply editing the "operation" field in Listing 4.

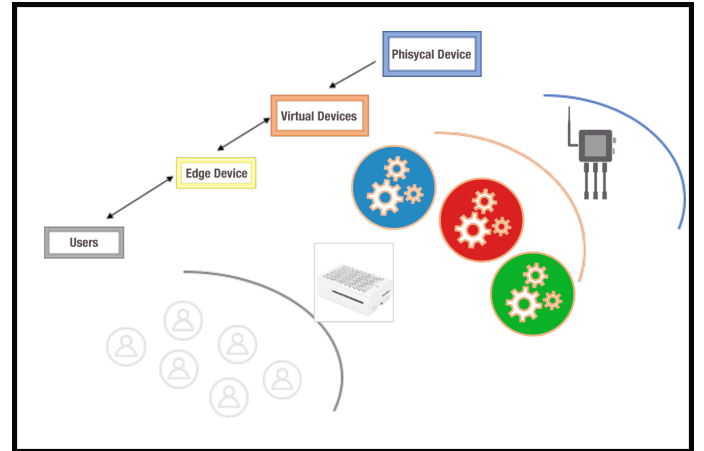


Fig. 5. Virtual Device - high level schema

Considering that an ip-cam could offer different micro-services and serve a multiplicity of users with different aims, i.e video quality, framing, frame rate, etc., the virtualization of the device, externally will give the impression that each user is talking to a different camera, but thanks to the processing on the Edge, the device will be physically one (Fig. 5). Thanks

to the data model designed the camera will be able to show itself virtually different to each user who makes a request.

## VI. CONCLUSIONS

The paper proposes an innovative method of managing services and related data in a smart environment at the Edge. A study was conducted on the current data models used for exchanging data among heterogeneous systems and especially to Edge devices in smart environments. To the best of our knowledge, there are not models to clearly describe both data and services that need to be executed, we presented the concept of Virtual Device, which extends the concept of physical device in an innovative way. Then, we presented how a Virtual Device can be described with an extension of the NGSI-LD standard, in order to increase the flexibility of the data model and specifying important information on how data are processed and services are provided. In the proposed use case, the concept of solid angle was presented as possible service of a Virtual Device for a video surveillance service and represented through the proposed data model. In future works, we would like to investigate the performance of Edge devices in case of service deployed using the proposed NGSI-LD based description.

## VII. ACKNOWLEDGMENTS

The research leading to these results has received co-funding from the European Union's Horizon 2020 research and innovation program under grant agreement N° 870338. This work was co-financed by the European Union - FSE, PON Research and Innovation 2014-2020 Axis I - Action I.1 "Dottorati innovativi con caratterizzazione industriale".

## REFERENCES

- [1] "FIWARE DATA MODELS." <https://fiware-datamodels.readthedocs.io/en/latest/index.html>.
- [2] A. Buzachis, M. Fazio, A. Celesti, and M. Villari, "Osmotic flow deployment leveraging faas capabilities," in *Internet and Distributed Computing Systems* (R. Montella, A. Ciaramella, G. Fortino, A. Guerrieri, and A. Liotta, eds.), (Cham), pp. 391–401, Springer International Publishing, 2019.
- [3] L. U. Khan, I. Yaqoob, N. H. Tran, S. M. A. Kazmi, T. N. Dang, and C. S. Hong, "Edge computing enabled smart cities: A comprehensive survey," *IEEE Internet of Things Journal*, pp. 1–1, 2020.
- [4] X. Xu, Q. Huang, X. Yin, M. Abbasi, M. R. Khosravi, and L. Qi, "Intelligent offloading for collaborative smart city services in edge computing," *IEEE Internet of Things Journal*, pp. 1–1, 2020.
- [5] N. A. M. Alduais, J. Abdullah, and A. Jamil, "Rdem: An efficient real-time data collection model for iot/wsn edge with multivariate sensors," *IEEE Access*, vol. 7, pp. 89063–89082, 2019.
- [6] T. Rahman, X. Yao, G. Tao, H. Ning, and Z. Zhou, "Efficient edge nodes reconfiguration and selection for the internet of things," *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4672–4679, 2019.
- [7] I. Filip, A. V. Postoaca, R. Stochitoiu, D. Neatu, C. Negru, and F. Pop, "Data capsule: Representation of heterogeneous data in cloud-edge computing," *IEEE Access*, vol. 7, pp. 49558–49567, 2019.
- [8] Y. Watanabe, K. Sato, and H. Takada, "Dynamicmap 2.0: A traffic data management platform leveraging clouds, edges and embedded systems," *International Journal of Intelligent Transportation Systems Research*, vol. 18, 11 2018.
- [9] B. D. Cruz, J. Cheng, Z. Song, and E. Tilevich, "Understanding the potential of edge-based participatory sensing: an experimental study," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, pp. 1–5, 2020.
- [10] A. J. Jara, Y. Bocchi, D. Fernandez, G. Molina, and A. Gomez, "An analysis of context-aware data models for smart cities: Towards fiware and etsi cim emerging data model," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLII-4/W3, pp. 43–50, 2017.
- [11] A. Galletta, A. Ruggeri, M. Fazio, G. Dini, and M. Villari, "Mesmart-pro: Advanced processing at the edge for smart urban monitoring and reconfigurable services," *Journal of Sensor and Actuator Networks*, vol. 9, p. 55, Dec 2020.
- [12] N. Ota and W. Kramer, "Tinyml: Meta-data for wireless networks," 01 2021.
- [13] R. the Docs., "FIWARE Data Models: NGSI-LD Device." Last access on February 1st, 2021.