A semantic approach for comparing Fog Service Placement Problems

Tanguy Godquin, Morgan Barbier, Chrystel Gaber, Jean-Luc Grimault, Jean-Marie Le Bars

A semantic approach for comparing Fog Service Placement Problems

Tanguy Godquin*†, Morgan Barbier*, Chrystel Gaber†
Jean-Luc Grimault†, Jean-Marie Le Bars*
*Normandie Univ, UNICAEN, ENSICAEN, CNRS, GREYC, 14000 Caen
†Orange Labs, France

***Abstract*—As fog computing has developed, new challenges have emerged regarding the placement of services within the network. Such issues are becoming especially important since the integration of IoT infrastructure and the relevant security concerns. There is significant variation in the definitions of service placement problems covered in the literature. Comparing these placements is a difficult task since the services, their placement objectives, their relationships with each other and the infrastructure of the network, strongly contribute to this diversity of definitions. This article proposes a methodology for comparing service placement algorithms and solutions supported by an ontology of service placement problems. This ontology is therefore used in our methodology to solve the problems annotated therein and to compare the results of the various algorithms as well as the quality of their solutions.**

***Index Terms*—IoT Security, Service Placement Problem, Ontology, Monadic Existential logic of the Second Order**

I. Introduction

Securing IoT devices may not always be achievable due to both their limited capabilities and restricted access imposed by the manufacturer. An alternative to securing every single device is to deploy security services at the edge of the network [1], following the fog computing paradigm [2], [3] which leads to Service Placement Problems (SPP) [4] or Fog Service Placement Problems (FSPP) [5].

Brogi et al. and Salaht et al. suggest [4], [6] classifying the algorithms for solving FSPPs into three categories: algorithms with a mathematical approach (mainly focused towards Integer Linear Programming (ILP) [5], [7] and Mixed Integer Linear Programming (MILP) [8]), search algorithms and heuristics (often based on greedy algorithms or heuristic behaviours [9]–[11]) and exotic algorithms (such as algorithms based on deep learning [12], graph theory [1] or even game theory [13]). The problems addressed in works on FSPPs follow a similar structure divided into three models (the infrastructure, application and deployment models [4]). However these problems are not identical, which complicates the comparison of solutions and the methods used. In addition, the objectives pursued in these works are also different. Some focus on improving QoS [7], [14], average response time [15], the overall security of the network [1] or even the resource usage [16]. Such diversity also occurs within the interactions among services and the network infrastructure where the number of devices, their type and characteristics fluctuate widely.

978-3-903176-32-4

In this article, we formalize FSPPs using Monadic Existential logic of Second-Order (MESO). In order to use and manipulate the proposed model, this formalization is then expressed within a semantic structure using an ontology which allows logical relationships. Finally, we provide a methodology for solving and comparing FSPPs solutions based on this ontology, which enables the comparison of placement solutions according to different resolution methods on the same infrastructure.

The structure of the article is as follows. Section II presents the FSPPs. Section III introduces ontologies. Section IV presents our methodology for solving and comparing service placement solutions.

II. The Fog Service Placement Problem (FSPP)

A. Defining a FSPP

According to Salaht et al. the definition of a service placement problem in the IoT involves defining the application, infrastructure and deployment models [4].

The infrastructure model describes the IoT network and covers the resources (physical devices such as sensors, actuators, servers, smartphones, etc.), their characteristics (such as processor frequency, memory, etc.), as well as those of the network (latency, bandwidth, etc.).

The application model, on the other hand, describes the relationships between services in the IoT network. Services can be either independent (monolithic) [4], [5], [7], [15], interdependant [8], [11] or all interacting [4], [16].

The last model, the deployment model, defines the conditions for deploying a service (see Section II-B2).

While the infrastructure model is linked to the network, the application and deployment models are specific to the services.

B. Formalizing a Fog Service Placement Problem

A Fog Service Placement Problem (FSPP) refers to identifying on which IoT devices security services should be deployed given specific conditions.

1) Modelling an IoT network: Finite Model Theory (FMT) allows defining the infrastructure with a structure $\mathcal{M} = (\mathcal{D}, \mathcal{R})$ where $\mathcal{D}$ refers to the domain including all the elements of the structure; here it contains all the devices within the IoT network. The symbol $\mathcal{R}$ stands for the vocabulary, it contains the relationships and functions affecting the elements of $\mathcal{D}$ (devices of the IoT). $\mathcal{R}$ may also include constants

(zero arity functions) as well as relationships ($=$, $<$ and $>$) comparing values to these constants.

This type of structure provides great flexibility without imposing limitations on the amount of information which can be contained. As an example, one may define the unary function $W : \mathcal{D} \to \mathbb{R}$ returning the processing power of a device and the binary function $B : \mathcal{D}^2 \to \mathbb{R}$ returning the bandwidth from one device to another.

2) Modelling a service placement: The domain elements which will support each service are defined by a unary relation variable. Let $\mathcal{S}$ be the set of these unary variables, we assume that these choices may be defined with a first-order formula Φ over the enhanced vocabulary $\mathcal{R} \cup \mathcal{S}$. Hence, the IoT security service placement problem can be expressed by a sentence of Monadic Existential Second-Order logic (MESO) as $\exists \mathcal{S} \Phi(\mathcal{R}, \mathcal{S})$.

a) Definition of a constraint: Let $\mathcal{S}^*$ be an interpretation of $\mathcal{S}$ in $\mathcal{D}$, a constraint C_i is an application from a subset of $(\mathcal{M}, \mathcal{S}^*)$ to $\{0, 1\}$.

b) Definition of an objective function: An objective function is expressed by a function $\mathcal{O}_i : (\mathcal{M}, \mathcal{S}^*) \to \mathbb{R}$. It evaluates and compares placement solutions and thereby defines a placement optimization problem. When several solutions are available, the objective function can be used to evaluate which solution is the best according to its evaluation criteria. The optimization problem can thus be solved either by minimizing or maximizing the value returned by this function.

3) Example on a toy model: Consider two security services S_1 and S_2 for which we define the following constraints:

C_1 : Each service is deployed on a single device.
C_2 : The S_1 service requires a power of 10 to operate.
C_3 : Neither service should be hosted on devices interacting.
C_4 : Each service must be reachable by a device with a minimum bandwidth capacity of 50.

The numerical values given are constants of the problem and can therefore be changed according to the actual data. Similarly, the listed constraints can be changed to match the services to be deployed. In this example, the constraints are expressed as the following first-order formulas where $E(x, y)$ is a binary relationship on $\mathcal{D}$ indicating whether x and y are linked together:

$$C_{1,i} : \exists x \forall y \; S_i(x) \wedge S_i(y) \to x = y.$$
$$C_2 : \forall x \; S_1(x) \to W(x) \geq 10.$$
$$C_3 : \forall x \forall y \; (S_1(x) \wedge S_2(y)) \to \neg E(x, y).$$
$$C_{4,i} : \forall x \exists y \; (S_i(x) \wedge E(x, y)) \to B(x, y) > 50.$$

The placement problem so described can be expressed using the following MESO formula: $\exists S_1 \exists S_2 \; C_{1,1} \wedge C_{1,2} \wedge C_2 \wedge C_3 \wedge C_{4,1} \wedge C_{4,2}$. Note that MESO sentences express NP-complete problems as well as easy problems.

III. Presentation of ontologies

Semantic structures seem appropriate to model our formal structure. They have been increasingly used in conjunction with IoT technologies for several years now. The scientific community refers to those structures as SWoT, an acronym that stands for Semantic Web of Things [17] and are commonly expressed using ontologies.

An ontology is a formal and explicit description of concepts within a specific field that can model the structure of a system, i.e. the relevant entities and relationships that arise from its observation [18].

A. Established IoT ontologies

The most commonly used ontology in SWoT research is the Semantic Sensor Network ontology (SSN[1]). This ontology defines all the concepts surrounding sensors and their observations. While the SSN ontology has a high level of detail in its design, it mainly covers sensor-based applications. However, the *Devices* class of SSN is particularly interesting as it allows a *Device* to be a composite of several smaller *Devices*.

Studies conducted by Bermudez-Edo et al. led to the development of the IoT-Lite ontology, a lightweight ontology that semantically describes the IoT [19] with fewer axioms than other ontologies such as SSN, Fiesta-IoT [20], OneM2M[2], M3 and M3-lite [21] or IoT-O [22]. The authors have shown that the axiom number of an ontology has a significant impact on its use in an IoT context where processing times are critical and capacity might be limited. One of the main benefits of this ontology lies in its lightweight, as well as in the defined concepts. IoT-Lite uses the definition of "Device" from SSN and describes a "Service" class with associated properties.

B. Related work for service placement based on ontologies

Allocation and, especially, the placement of services is highly dependent on the environment in which they are deployed. Having an extensive knowledge of the infrastructure is important for identifying the best location to deploy a security service. Ontologies enable the infrastructure to be described formally and therefore provide structure to the collected information.

The work of Tao et al. [23] illustrates the value of semantic structures for representing the infrastructure model. This study introduces a new and interesting ontology, however, it does not fit our application and has several limitations. Moreover, no other ontologies were reused in its design and it does not seem to be publicly accessible or listed on the LOV4IoT platform [17] thereby impeding reusability and usage with new ontologies.

The work of Nezami et al. [24] is based on IoT-O [22] associated with a second ontology, Cloud-O [25] forming a third ontology. The authors suggest using ontologies to allocate resources in the edge computing paradigm. They propose a methodology with four steps, resource discovery, modelling, selection and allocation.

Among the methodologies in the literature that rely on semantic structures, the work of Petrovic and Tosic stands out particularly [26]. In their study, the authors

[1] http://www.w3.org/ns/ssn/

[2] https://git.onem2m.org/MAS/BaseOntology

define a structural component, called SMADA-fog, which automatically generates code for the management of fog computing infrastructure. The authors use meta-model which defines the structure and constraints of a family of models on two different aspects of deployment in a fog computing architecture. This modelling is the core of SMADA-fog as it allows both the annotation of the deployment model in the ontology, as well as using adaptation strategies through code generation for adapting to changes in the environment.

Smada-fog is particularly interesting for its use of semantic structures for automated code generation. This operation is similar to service deployment principles due to the customizable code that can be deployed. The work of Petrivic and Tovic [26] proposes a highly in-depth approach on implementation demonstrating the benefits of semantic structures for complex data representation. SMADA-fog, however, meets its limits in the definition of its semantic structure and its optimization model since we do not know if it is interoperable with other ontologies. Remark that the ontology used is not listed in LOV4IoT nor publicly available. Moreover, this structure relies solely on linear optimization and the reasoning capabilities of ontologies to perform its code deployment.

IV. Solving and comparison methodology

Comparing placement solutions requires both context awareness and understanding placement problems related to these solutions.

We use our ontology to build a knowledge base and act as a cornerstone in our methodology for solving placement problems and comparisons of their solutions.

A. Knowledge base creation

Knowledge bases gather information about a specific topic under an accessible and interpretable format. For ontologies, a knowledge base is a set of individual instances of classes defined in an ontology.

1) Benefits of using an ontological knowledge base: The first benefit lies upon the formal part of the structure. As demonstrated by Shimizu, OWL (Web Ontology Language), commonly used for writing ontologies, is directly linked to descriptive logic [27]. Another benefit of using ontologies involves inferences. Inferences are especially interesting here since it allows, using logic, additional information to be retrieved from those that have been entered into the ontology.

2) Building the knowledge base: Building a knowledge base, such as designing an ontology, is an iterative process that can be expanded and refined. We build our knowledge base in three steps:

a) Infrastructure modelling: This modelling gathers all information available for the addressed IoT network. It includes physical resources such as available equipment (PCs, servers, sensors...), their characteristics (CPU, RAM...) and those of the network. To model the infrastructure we define the $\mathcal{M} = (\mathcal{D}, \mathcal{R})$ structure, introduced in Section II-B. Following the modelling, the model is annotated in the ontology using individual instances.

b) IoT services definition: In this step, we define the services to be deployed within our IoT infrastructure. Defining services requires each service to be defined individually according to its deployment model. The deployment model of a service covers the positioning constraints of the service and at least one objective function.

We must also describe how these services interact, which is part of the application model. The application model defines whether the services are monolithic, meaning the services form independent blocks that do not interact, whether some are interdependent, or whether all the services are mutually dependent. The positioning of interdependent services is more complex.

The service definition step ends with the annotation of the services and related information into the knowledge base.

c) Design of resolution algorithms: Looking for solutions to service placement problems implies the use of resolution algorithms. Such algorithms can be of various natures [4] and therefore be annoted and linked to compatible services into the knowledge base. When annotating resolution algorithms into the knowledge base, the idea is to not only ensure that they are compatible with the infrastructure model, but also to indicate the services for which they can assist finding placement solutions.

B. Solving and comparing service placement problems

Our methodology is divided into five steps: information retrieval, solution identification, validation, evaluation and comparison.

a) Information retrieval: Using our knowledge base, we can interpret information about the network infrastructure, services data, their deployment model, the resolution algorithms and the set of relationships between entities described in the ontology.

b) Solution identification: We want to run the resolution algorithms compatible with the service we wish to deploy in order to identify placement solutions. These algorithms use the information extracted in the previous step and, more generally, the entire knowledge base (including constraints) to identify solutions. Depending on the algorithms, one or more solutions can be returned. Some algorithms may even be able to return the optimal solution directly, however, this means that these algorithms describe optimization concepts as discussed in the following steps.

c) Validation: The validation step is performed using constraint filtering on the solutions returned by the algorithms. Solutions that do not satisfy the constraints are not retained. No satisfactory solution may be found, in which case constraints may be weakened to offer looser placement conditions. Filtering can be redundant if some resolution algorithms already use constraints for identifying their solution. However, it remains necessary to validate the solutions returned by the algorithms.

d) Evaluation: Each suitable solution is evaluated by an evaluator (most of the time an objective function) to assign each one a score. Solving service placement problems

regularly involves optimization in the form of maximizing or minimizing scores. Examples of such evaluators could be functions that indicate the contribution of a solution on the QoS of the network [7], the security provided on the network [28] or the quality of the solution [1].

e) Comparison: The comparison step performs a score analysis of each solution and then selects the most appropriate one for service deployment. Using a single evaluator, the algorithm that returns the best result can be easily distinguished by tracing the origin of the solution that has obtained the best score according to the evaluation criteria. We define evaluators as functions that return a single numerical value which can compete with one another to compare placement solutions.

C. Overview of our ontology

The purpose of our ontology is to model service placement problems within the IoT, to unify the work on the service placement as well as to compare solutions. We named this ontology *FSPlacementOntology* in reference to the *Fog Service Placement Problem* (FSPP) defined by Skarlat et al. [5].

For the ontology to best models service placement problems, it needs to include all the necessary models that Salaht et al. [4] mentioned, namely the infrastructure, application and deployment model.

In addition, to match our methodology proposed in the previous sections, the ontology must also provide representations of the resolution algorithms and placement solutions.

1) External concepts: Service placement problems, and more particularly FSPP, are mainly centred around the concept of service. We therefore decided to design our model around the *Service* class from the IoT-Lite ontology [19].

We have modelled the rest of our application concepts around this class using a *middle-out* methodology [29].

The IoT-Lite ontology models the service exposition of a device using the *exposed* and *exposedBy* properties connecting the *Services* to the *Device* class from the SSN ontology [30].

In our application, services are hosted on the devices. Therefore, it is necessary to model the *Device* class, representative of them. As a result, the classes *Device* and *System* (superclass of *Device*) are imported into FSPlacementOntology, as well as the *exposed* and *exposedBy* properties.

2) Major concepts: The deployment model representation in ontology requires the definition of concepts related to constraints and objective functions. For this purpose, a *Constraint* class has been defined, defining the prerequisites to be met for a valid deployment solution.

In the interests of realism, when annotating a *Service* within the ontology, at least one placement constraint must be provided using the *hasConstraint* existential property.

A second class, *Information* connected to *Constraint* through the *involves* property has also been introduced to model the information used for the definition of a constraint. Among other information, this class lists *DeviceCharacteristics* and *SpecificHardwareComponent*.

The deployment model is completed by defining the *ObjectiveFunction* class linked to *Service* through the *ObjectiveFunction* existential property stating that a *Service* must always provide at least one *ObjectiveFunction*. As for the service dependencies, they are modelled using the *dependsOn* property.

The resolution algorithms are modelled within the ontology (*ResolutionAlgorithm*) and related to the solutions (*PlacementSolution*) using the *findSolution* universal property to match as closely as possible the intended application. This property allows resolution algorithms to return exclusively instances of *PlacementSolution* or to return nothing at all. Placement solutions include at least one instance of the *Pairing* class using the *ContainPairs* existential property. The latter exclusively matches an instance of *Service* with an instance of *Device* through the *hasPairedService* and *hasPairedDevice* properties.

Finally, the *Device* class is linked to numerous other concepts relating to device performance (*CPU*, *Storage*, *Memory*), *CommunicationInterface* or more generally *DeviceCharacteristic*.

3) Extending the ontology: A simplified ontology has been chosen to limit the number of axioms it contains, allowing greater flexibility for usage within the IoT context.

The benefits of this ontology lie in the ability to annotate service placement problems as well as in its versatility to describe placement constraints using semantic restrictions [31].

Restrictions may refer to concepts not currently described in our ontology. The idea is to provide a core ontology that can be used and enriched by the community to match various applications. Note that our assumption that the problem is expressible by a MESO formula limits the scope of our ontology. Modelling the deployment using a different logic may lead to changes in the ontology that could be considered in its evolution.

V. Conclusion

The aim of this paper is to generalize and formalize Fog Service Placement Problems with Monadic Existential Second-Order sentences. Our approach provides a common structure and vocabulary to express Fog Service Placement Problems, thus assisting in the design of our ontology devoted to such problems. Using our ontology, information related to the placement problems can be gathered, stored and used to infer additional knowledge, to form a knowledge base. Finally, based on this knowledge base, we propose a methodology for solving and comparing solutions to placement problems. While our comparison methodology remains invariant to the logic used in formalization, our ontology may not be suitable for placement problems that cannot be modelled according to Monadic Existential Second-Order logic. Therefore, it might be relevant to consider this category of problems as the ontology evolves.

REFERENCES

[1] T. Godquin, M. Barbier, C. Gaber, J.-L. Grimault, and J.-M. Le Bars, "Applied graph theory to security: A qualitative placement of security solutions within IoT networks," vol. 55, p. 102640. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2214212620308000

[2] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing - MCC '12*. ACM Press, p. 13. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2342509.2342513

[3] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog Computing: A Platform for Internet of Things and Analytics," in *Big Data and Internet of Things: A Roadmap for Smart Environments*, N. Bessis and C. Dobre, Eds. Springer International Publishing, vol. 546, pp. 169–186. [Online]. Available: http://link.springer.com/10.1007/978-3-319-05029-4_7

[4] F. A. Salaht, F. Desprez, and A. Lebre, "An overview of service placement problem in Fog and Edge Computing." [Online]. Available: https://hal.inria.fr/hal-02313711

[5] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized IoT service placement in the fog," vol. 11, no. 4, pp. 427–443. [Online]. Available: http://link.springer.com/10.1007/s11761-017-0219-8

[6] A. Brogi, S. Forti, C. Guerrero, and I. Lera, "How to Place Your Apps in the Fog – State of the Art and Open Challenges," vol. 50, no. 5, pp. 719–740. [Online]. Available: http://arxiv.org/abs/1901.05717

[7] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards QoS-Aware Fog Service Placement," in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*. IEEE, pp. 89–96. [Online]. Available: http://ieeexplore.ieee.org/document/8014364/

[8] R. Doriguzzi-Corin, S. Scott-Hayward, D. Siracusa, M. Savi, and E. Salvadori, "Dynamic and Application-Aware Provisioning of Chained Virtual Security Network Functions," vol. 17, no. 1, pp. 294–307. [Online]. Available: http://arxiv.org/abs/1901.01704

[9] H. Gupta, A. V. Dastjerdi, S. K. Ghosh, and R. Buyya, "iFogSim: A Toolkit for Modeling and Simulation of Resource Management Techniques in Internet of Things, Edge and Fog Computing Environments." [Online]. Available: http://arxiv.org/abs/1606.02007

[10] R. Mahmud and R. Buyya, "Modelling and simulation of fog and edge computing environments using iFogSim toolkit," pp. 1–35.

[11] Y. Xia, X. Etchevers, L. Letondeur, T. Coupaye, and F. Desprez, "Combining hardware nodes and software components ordering-based heuristics for optimizing the placement of distributed IoT applications in the fog," in *Proceedings of the 33rd Annual ACM Symposium on Applied Computing - SAC '18*. ACM Press, pp. 751–760. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3167132.3167215

[12] Z. Tang, X. Zhou, F. Zhang, W. Jia, and W. Zhao, "Migration Modeling and Learning Algorithms for Containers in Fog Computing," vol. 12, no. 5, pp. 712–725.

[13] H. Zhang, Y. Xiao, S. Bu, D. Niyato, R. Yu, and Z. Han, "Computing Resource Allocation in Three-Tier IoT Fog Networks: A Joint Optimization Approach Combining Stackelberg Game and Matching." [Online]. Available: http://arxiv.org/abs/1701.03922

[14] A. Yousefpour, A. Patil, G. Ishigaki, I. Kim, X. Wang, H. C. Cankaya, Q. Zhang, W. Xie, and J. P. Jue, "QoS-aware Dynamic Fog Service Provisioning." [Online]. Available: http://arxiv.org/abs/1802.00800

[15] F. B. Jemaa, G. Pujolle, and M. Pariente, "QoS-Aware VNF Placement Optimization in Edge-Central Carrier Cloud Architecture," in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7.

[16] B. Donassolo, I. Fajjari, A. Legrand, and P. Mertikopoulos, "Fog Based Framework for IoT Service Provisioning," in *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8651835/

[17] A. Gyrard, C. Bonnet, K. Boudaoud, and M. Serrano, "LOV4IoT: A Second Life for Ontology-Based Domain Knowledge to Build Semantic Web of Things Applications," in *2016 IEEE 4th International Conference on Future Internet of Things and Cloud (FiCloud)*, pp. 254–261.

[18] N. Guarino, D. Oberle, and S. Staab, "What Is an Ontology?" in *Handbook on Ontologies*, pp. 1–17.

[19] M. Bermudez-Edo, T. Elsaleh, P. Barnaghi, and K. Taylor, "IoT-Lite: A lightweight semantic model for the internet of things and its use with dynamic semantics," vol. 21, no. 3, pp. 475–487. [Online]. Available: http://link.springer.com/10.1007/s00779-017-1010-8

[20] R. Agarwal, D. G. Fernandez, T. Elsaleh, A. Gyrard, J. Lanza, L. Sanchez, N. Georgantas, and V. Issarny, "Unified IoT ontology to enable interoperability and federation of testbeds," in *2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*. IEEE, pp. 70–75. [Online]. Available: http://ieeexplore.ieee.org/document/7845470/

[21] A. Gyrard, S. K. Datta, C. Bonnet, and K. Boudaoud, "Cross-Domain Internet of Things Application Development: M3 Framework and Evaluation," in *2015 3rd International Conference on Future Internet of Things and Cloud*. IEEE, pp. 9–16. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7300791

[22] N. Seydoux, K. Drira, N. Hernandez, and T. Monteil, "IoT-O, a Core-Domain IoT Ontology to Represent Connected Devices Networks," in *Knowledge Engineering and Knowledge Management*, ser. Lecture Notes in Computer Science, E. Blomqvist, P. Ciancarini, F. Poggi, and F. Vitali, Eds. Springer International Publishing, vol. 10024, pp. 561–576. [Online]. Available: http://link.springer.com/10.1007/978-3-319-49004-5_36

[23] M. Tao, J. Zuo, Z. Liu, A. Castiglione, and F. Palmieri, "Multi-layer cloud architectural model and ontology-based security service framework for IoT-based smart homes," vol. 78, pp. 1040–1051. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0167739X16305775

[24] Z. Nezami, K. Zamanifar, D. Arena, and D. Kiritsis, "Ontology-Based Resource Allocation for Internet of Things," in *Advances in Production Management Systems. Production Management for the Factory of the Future*. Springer, Cham, pp. 323–330. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-30000-5_41

[25] M. Rekik, K. Boukadi, and H. Ben-abdallah, "Cloud Description Ontology for Service Discovery and Selection:," in *Proceedings of the 10th International Conference on Software Engineering and Applications*. SCITEPRESS - Science and and Technology Publications, pp. 26–36. [Online]. Available: http://www.scitepress.org/DigitalLibrary/Link.aspx?doi=10.5220/0005556400260036

[26] N. Petrovic and M. Tosic, "SMADA-Fog: Semantic model driven approach to deployment and adaptivity in Fog Computing," p. 102033. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1569190X19301649

[27] C. M. Shimizu, "Rendering OWL in LaTeX for Improved Readability: Extensions to the OWLAPI," p. 107.

[28] T. Godquin, M. Barbier, C. Gaber, J.-L. Grimault, and J.-M. Le Bars, "Placement optimization of IoT security solutions for edge computing based on graph theory," in *38th IEEE International Performance Computing and Communications Conference (IPCCC 2019)*. [Online]. Available: https://hal.archives-ouvertes.fr/hal-02314892

[29] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," vol. 11, no. 2, pp. 93–136. [Online]. Available: https://www.cambridge.org/core/product/identifier/S0269888900007797/type/journal_article

[30] Semantic Sensor Network Ontology. [Online]. Available: https://www.w3.org/TR/vocab-ssn/

[31] M. Horridge, "A Practical Guide To Building OWL Ontologies Using Prot´eg´e 4 and CO-ODE Tools Edition 1.3," p. 108.