1 2 9 0

UNIVERSIDADE D
COIMBRA

Ngombo Armando

# AN APPROACH TO THE UNIFIED MANAGEMENT OF HETEROGENEOUS IOT ENVIRONMENTS

Doctoral thesis submitted to the Doctoral Program in Information Science and Technology, supervised by Professors Jorge Sá Silva and Fernando Boavida, and presented to the Department of Informatics Engineering of the Faculty of Sciences and Technology of the University of Coimbra.

July 2020

Faculdade de Ciências e Tecnologia
da Universidade de Coimbra

# ABORDAGEM PARA GESTÃO UNIFICADA DE AMBIENTES HETEROGÊNEOS IOT

Ngombo Armando

Tese no âmbito do Programa de Doutoramento em Ciências e Tecnologias da Informação, orientada pelo Professor Doutor Jorge Sá Silva e pelo Professor Doutor Fernando Boavida, e apresentada ao Departamento de Engenharia Informática da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Julho de 2020

The research work presented in this thesis was partially carried out in the scope of the Social-Oriented IoT Architecture, Solutions and Environment (SOCIALITE) Project (PTDC/EEI-SCR/2072/2014) and the Mobilizador 5G Project (POCI/01/0247/FEDER/024539), co-financed by COMPETE 2020, Portugal 2020 - Operational Program for Competitiveness and Internationalization (POCI), by European Union's European Regional Development Fund (ERDF), by national funds through the Portuguese Foundation for Science and Technology (FCT), I.P., within the scope of the project Centre for Informatics and Systems of the University of Coimbra (CISUC) - UID/CEC/00326/2020, and by European Social Fund, through the Regional Operational Program Centro 2020.

Partial support was also given by the *Universidade Kimpa Vita (UNIKIVI)* and the *Escola Superior Politécnica do Uíge (ESPU)*, and co-financed by the Angolan National Grant Institute (INAGBE) - *Guia de apresentação número:* 077/2018, within the scope of the Angolan National Staff Training Plan (*Plano Nacional de Formação de Quadros (PNFQ))*, under the supervision of the Angolan Ministry of Higher Education, Science, Technology and Innovation.

Cofinanciado por:

*To uncle* ***Luís Benjamin Nkosi***

**"Be not silent, but beware of interruption and of answering words with heat."**

*Maxim 25.* Ptahhotep, XXV BCE

# Acknowledgements

I would like to thank my supervisors, who besides sharing their professional skills with me towards innovative research, supported me in my journey. I would like to thank Prof. Jorge Sá Silva for having hosted me in his team from the early stage of the doctoral program and Prof. Fernando Boavida, for his dedication and rigorous approach during the revision of the texts that we produced as both research papers and project deliverables.

I received support from the faculty members, the researchers, and the staff in both the ESPU/UNIKIVI and the *Departamento de Engenharia Informática (DEI)*/CISUC. Therefore, I am so thankful to all of them, with attention to Prof. Maria de Fátima for being such a citizen role model and a professional of the highest possible levels. To Cândido Kiluando and Francisco dos Santos, for having insured the continuity of my tasks as a lecturer, successfully.

I had the chance to be part of a research team that provided me with the required synergy for my thesis project. Hence, my gratitude goes also to Marcelo Fernandes, Prof. Soraya Sinche, Dr Duarte Raposo, Jorge Eduardo Rivadeneira, Prof. André Rodrigues, and Rui Almeida.

The journey to this thesis counted with the phases before coming to Coimbra. Among them, the candidate process where I had to provide some letters of recommendation. Such letters were promptly endorsed by people that I met both in academia and in the industry, over the last 13, 10 , 8 years ago. That is why, to Prof. Dóra Máros, Kurt Andorfer, and Prof. Makiese Mavakala, I address my first special thanks.

Many of the colleges from the research center have been crucial to my integration in the Portuguese way of living and shared with me both methodology and hands-on tools for advancing in several stages of the doctoral program. I am though grateful to Stanley Lima, Prof. Antonio Martinez-Millana, Dr Rahul Sharma, Jaime Correia, Ricardo Mendes, Paulo Silva, Ronaldo Júnior, Joaquim Leitão, Vitor Graveto, Karima Velasquez, David Perez Abreu, and Rui Queiroz.

I am beholden to my *extended* parents and tireless fighters Elizabeth Malungo Bemba, Armando João Bemba, Maria Joana Mana, and Luís Benjamin Nkosi (national hero). They have unconditionally invested time, faith, energy, and money on my academic training 15, 20, 25, 30 years ago.

I am also beholden to my sisters Isabel Armando and Antonica Armando, for financially supporting those days when I happened to have no incomes.

My final special thanks go to the following ones, both old and new friends, who have contributed to this journey weather with material or immaterial means. The non-exhaustive list includes *"les très chers" Prof. Moniz Tunga, Mauro Ferreira, Axcelle Inguweneza and Yuri Pedro*, Teresa Clara & Paulo, Pedro Duarte, Joaquim Madeira, Prof. Daniel NTotila, Prof. Mampuya Fidèle, Maya Bocheva (what a dreamer and incredible mind skills!), and Tales Sabará (what a will power and talented mind!).

There is not enough space in a document to mention the full list of the great people who made it all happen the less painful possible way. Hence, to you, anonymous companion and selfless angel, I did not forget to send you the recognition you deserve.

The wise people once claimed, *"foundations are placed at the basement because they provide the structure's stability from the ground"*. Thus, finally, I would like to address a great thank to my "built" family. For having insured the structure of the home during my absence... I am indebted to Formosa Makaya and Lwana Makaya-Armando.

# Abstract

The Internet of Things (IoT) can be defined as an extension of computer networks and the Internet, to a myriad of both smart and connected devices labelled "things". Traditionally, wireless sensors/actuators form the founding block of the IoT's perception layer, which core functions are sensing the environment and actuate on it. Thus, IoT systems were initially developed upon hardware electronic components for data acquisition and interaction with the target environment. Later, virtual sensors (also known as Software-Based Sensors (SBSs)) entered into play, consisting of software modules that offer high-level Application Programming Interfaces (APIs) that could be easily integrated into IoT products and services. A more recent trend in literature for massive data acquisition considers "social sensing", a type of sensing in which people are regarded as data producers, or Human-Based Sensors (HBSs), via their activities in Online Social Networks (OSNs).

Having a variety of new ways of collecting data may not be effective if we cannot ensure essential and specific management capabilities of the heterogeneity, in types and forms, of the entities in the perception layer. Our definition of Device Management (DM) relies on the Y.2060 recommendation by the International Telecommunication Union - Telecommunication Standardization Sector (ITU-T). DM covers all the set of operations that an Information Technologies (IT) system needs to monitor and interact with the managed entities. The aim of DM is to make sure that the applications running on a given device operate well. Hence, DM must also be extended to all types of "resources" that can be explored in the perception layer, either physical, virtual or social.

It is, therefore, crucial to develop management platforms that consider such heterogeneity in "devices", in a unified way and, additionally, that are based on open standards. On the one hand, such platforms have the potential to explore the breadth in the definition of the devices that interact with both physical and virtual worlds. On the other hand, such a solution guarantees interoperability of the management solutions. Typically, the management of IoT devices mainly focuses on well-established electronic-based entities. Well, one may want to change the algorithm of a virtual sensor, on-the-fly, or prevent a social sensor from feeding the backend for a while.

In our research, we tried to overcome the challenge of a unified managed for both traditional and new "devices" in the IoT. We leveraged an IoT middleware as the central component to provide the flexibility and scalability of our proposed architecture. Based on a comprehensive survey, we adopted the Future Internet - Ware (FIWARE) and Lightweight Machine-To-Machine (LwM2M), as the middleware and management protocol, respectively, for the applied case studies of the generic architecture we proposed.

We also developed two use-cases, namely *IoT Student Advisor and Best Lifestyle Analyzer (ISABELA)* and *5GOpenclasses*, through which we have shown that the management of the referred three types of sensing is feasible from both functional and performance points of view.

With the help of our research team, I actively conducted the design and implementation phases of both use-cases to test the models proposed in this thesis. Thus, I will present the degree of success in achieving such a goal and will mention the research opportunities that we opened for future work.

**Keywords:** IoT, Management, Heterogeneity, FIWARE, LwM2M, ISABELA, 5GOpenclasses.

# Resumo

Podemos definir a Internet das Coisas (IoT) como sendo uma extensão da Internet em geral, a um conjunto de objetos inteligentes e conectados, conhecidos genericamente por "coisas".

A IoT apoia-se fundamentalmente numa camada dita de deteção/perceção, cujas funções centrais são: detetar os fenómenos no ambiente em que são aplicados e atuar no mesmo. Tradicionalmente, esta camada consiste em um conjunto de redes de sensores/atuadores sem fios. Assim sendo, os sistemas IoT foram inicialmente desenvolvidos sobre componentes eletrónicos físicos para aquisição de dados e interação com o ambiente alvo. Mais tarde, surgiram sensores virtuais (também conhecidos como Sensores Baseados em Software (SBSs)). Estes, são programas informáticos, compostos por módulos que ofereciam interfaces de programação de aplicações (API) de alto nível, e de fácil integração em produtos e serviços IoT. A tendência mais recente para aquisição massiva de dados em ambientes IoT considera o paradigma dos "sensores sociais". Trata-se de um tipo de deteção em que as seres humanos, ou Sensores Sociais Humanos (HBSs), são considerados como produtores de dados contextualizados, através das suas atividades em Redes Sociais (OSN).

O aproveitamento desta variedade de formas de se recolher dados em massa não pode ser eficiente, sem a garantia de uma gestão ampla de todas as entidades que compõem esta camada de deteção. O objetivo final da gestão dos dispositivos IoT (DM) é garantir que a execução dos aplicativos e productos funcionem conforme ela foi projetada. Baseando-se na recomendação Y.2060 da União Internacional das Telecomunicações (ITU-T), a DM refere-se ao conjunto de técnicas para a monitorização tanto das atividades como do estado das entidades sob controlo, e para a interação com as mesmas. Por conseguinte, a DM deve ser alargada a todos os "recursos" da camada de perceção na "nova" IoT, sejam eles físicos, virtuais ou sociais.

Neste contexto, é crucial desenvolverem-se soluções de gestão, que considerem a heterogeneidade de "recursos", de forma unificada e, adicionalmente, baseadas em padrões abertos. Estas plataformas teriam o potencial de explorar a nova definição dos dispositivos que interagem com os ambientes físicos e virtuais na IoT. Por outro lado, uma tal abordagem garante uma interoperabilidade entre as soluções de gestão. Tipicamente, a gestão de dispositivos em IoT foca-se principalmente numa visão eletrónica dos "recursos" na camada de perceção. Ora, pode haver necessidade de se alterar o algoritmo de um sensor virtual ou de se decidir, momentaneamente, que um sensor social cesse de enviar dados ao sistema de informação.

Na nossa pesquisa, tentámos ultrapassar estes desafios de gestão unificada dos dispositivos na nova IoT, propondo uma arquitetura aberta e flexível, centrada em um *middleware*. Com base numa revisão exaustiva do estado da arte, adotámos o FIWARE e o LwM2M, como *middleware* e protocolo de gestão, respetivamente, nos casos de usos aplicados a nossa arquitetura genérica. O conjunto de implementações em provas de conceito, com realce as que serviram para os casos de estudos por nós denominados *ISABELA* e *5GOpenclasses*, demonstrou que a gestão unificada dos três tipos de entidades na camada de perceção IoT, é viável, tanto do ponto de vista funcional como do ponto de vista do desempenho.

Com a ajuda da nossa equipa de investigação, conduzi ativamente as fases de concepção e de implementação dos casos de uso, onde testámos os modelos propostos nesta tese. Na presente monografia, apresentarei o grau de sucesso em alcançar este objetivo de gestão unificada, e por fim, mencionarei as oportunidades de investigação que identificámos para trabalhos futuros.

**Palavras-chave:** IoT, Gestão, Heterogeneidade, FIWARE, LwM2M, ISABELA, 5GOpenclasses.

# Foreword

The research activities for this thesis were developed within the Laboratory of Communications and Telematics (LCT) of the CISUC. The thesis project resulted in several publications and technical reports for two national projects, namely, SOCIALITE (2016-2019) and *Mobilizador 5G* (2017-2020).

The main objectives of the SOCIALITE project were to define a version of an IoT people-centric architecture by developing and exploring a general Cyber-Physical System architecture to support both, People2People and People2Thing interaction. SOCIALITE also aimed to explore the developed middleware and services in people-centric context, with the aim of demonstrating their use in enhancing the autonomy and quality of life of citizens.

`https://www.cisuc.uc.pt/projects/show/251`.

The *Mobilizador 5G* is a Portuguese national project that involved fourteen (14) companies, research and innovation centres, under the leadership of Altice Labs. The strategic goal of this project was to prepare Portugal as a producer of Processes, Product and Services (PPS) for the Fifth Generation of Mobile Telecommunications (5G) networks. The solutions developed by the consortium should be part of a Technology Readiness Level (TRL) 7/8 integrated demonstration, providing services in the framework of future 5G networks.

`https://5go.pt/en/home/`, `https://www.cisuc.uc.pt/projects/show/215`.

The SOCIALITE project introduced me to the funding concepts, frameworks, technologies, applied tools and the development of innovative use-case in IoT. As for the *Mobilizador 5G* project, I was the correspondent researcher between the CISUC and the consortium for a development unit labelled "PPS4". The challenge in PPS4 was the development of innovative 5G-aligned products and services for human-to-human communication. I designed and co-developed a solution that leveraged IoT capabilities in the framework of future 5G networks. Hence, our solution was designed to explore the use-cases described in 5G vision, namely massive Machine-Type Communication (mMTC), enhanced Mobile Broadband (eMBB) and ultra-Reliable and Low Latency Communications (uRLLC).

All the articles produced during the thesis have been submitted to refereed and mainly indexed venues[1,2]. We have survey papers reporting on both new trends and developments in the literature. We also have papers of application, experimentation and performance evaluation type, reporting on our use-cases.

[1] `https://www.scimagojr.com/aboutus.php`, accessed on 2020-06-23

[2] `https://www.core.edu.au/conference-portal/About-the-db`, accessed on 2020-06-23

The resulted publications are listed below:

- **Journal articles (4+3)**

  - **Armando N.**, Sá Silva J., Boavida F., "An Approach to the Unified Management of Heterogeneous IoT Environments", IEEE Internet of Things Journal. Publisher: IEEE. *Ranking: Q1 (SCimago - Scientific Journal Rankings (SJR) 2019).* [**SUBMITTED, 2020-06-26**].
  - **Armando N.**, Almeida R., Fernandes J.M., Sá Silva J., Boavida F., "End-to-end Experimentation of a 5G Vertical within the Scope of Blended Learning", IEEE Access. Publisher: IEEE. *Ranking: Q1 (SJR 2019).* [**SUBMITTED, 2020-06-19**].
  - Rivadeneira Muñoz J.E., **Armando N.**, Sá Silva J., Rodrigues A., Boavida F., "Privacy-Aware Frameworks in the Era of the Internet of Things - Review and Proposal", IEEE Communications Surveys and Tutorials. Publisher: IEEE. *Ranking: Q1 (SJR 2019).* [**SUBMITTED, 2020-05-11**].
  - Sinche S., Hidalgo P., Fernandes J.M.., Raposo D., Sá Silva J., Rodrigues A., **Armando N.**, Boavida F., "Analysis of Student Academic Performance Using Human-in-The-Loop Cyber-Physical Systems". Telecom. 2020; 1(2):18-31. Publisher: MDPI. https://www.mdpi.com/2673-4001/1/1/3. *First issue of the journal.*
  - Fernandes J.M., Raposo D., **Armando N.**, Sinche S., Sá Silva J., Rodrigues, A., Pereira V., Gonçalo Oliveira H., Macedo L., Boavida F., "ISABELA  A Socially-Aware Human-in-the-Loop Advisor System", Online Social Networks and Media, vol. 16, pp. 100060-100060, 2020. Publisher: Elsevier. https://doi.org/10.1016/j.osnem.2020.100060. *Ranking: Q2 (SJR 2019).*
  - Sinche S., Raposo D., **Armando N.**, Rodrigues, A., Boavida F., Pereira V., Sá Silva J., "A Survey of IoT Management Protocols and Frameworks", IEEE Communications Surveys and Tutorials, vol. 22, no. 2, pp. 1168-1190, 2019. Publisher: IEEE. https://doi.org/10.1109/COMST.2019.2943087. *Ranking: Q1 (SJR 2019).*
  - **Armando N.**, Rodrigues A. , Pereira V., Sá Silva J., Boavida F. , "An Outlook on Physical and Virtual Sensors for a Socially Interactive Internet," Sensors, vol. 18, no. 8, p. 2578, Aug. 2018. Publisher: MDPI, https://doi.org/10.3390/s18082578. *Ranking: Q2 (SJR 2019).*

- **International Conference Articles (5+1)**

  - **Armando N.**, Sinche S., Rodrigues A., Sá Silva J., Boavida F., "A Comparative Assessment of Ultralight 2.0 and LwM2M 1.0", in Proceedings of IEEE Globecom 2020: Selected Areas in Communications: Internet of Things and Smart Connected Communities, Taipei, Taiwan, 7-11 December 2020. Publisher: IEEE. *Ranking: B (Computing Research and Education (CORE) 2018).* [**SUBMITTED, 2020-03-28**].
  - **Armando N.**, Fernandes J.M., Rodrigues A., Sá Silva J., Boavida F., "Exploring Approaches to the Management of Physical, Virtual, and Social Sensors", in Proceedings of the IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Virtual Conference (Due to Covid19 pandemic situation), 6-9 July 2020. Publisher: IEEE. *Ranking: A* (CORE 2018).* [**IN PRESS**].
  - **Armando N.**, Fernandes J.M., Sinche S., Raposo D., Sá Silva J., Boavida F., "A Unified Solution for IoT Device Management", in Proceedings of the 22nd International Symposium on Wireless Personal Multimedia Communications, 2019, pp. 1-6, Lisbon, Portugal, 2019. Publisher: IEEE. *Ranking: C (Excellence in Research for Australia (ERA) 2010).* `https://doi.org/10.1109/WPMC48795.2019.9096203`.
  - Fernandes J.M., Raposo D., Sinche S., **Armando N.**, Sá Silva J., Rodrigues A., Macedo L., Gonçalo Oliveira H., Boavida F., "A Human-in-the-Loop Cyber-Physical Approach for Students Performance Assessment," in Proceedings of the 4th International Workshop on Social Sensing (Socialsens), In conjunction with the ACM/IEEE CPS and IoT Week, Montreal, Canada, 2019, pp. 36-42. Publisher: ACM. `https://doi.org/10.1145/3313294.3313387`.
  - Fernandes J.M., Raposo D., **Armando N.**, Sinche S., Sá Silva J., Rodrigues A., Pereira V., Boavida F., "An Integrated Approach to Human-in-the-Loop Systems and Online Social Sensing," in Proceedings of IEEE INFOCOM WKSHPS, Paris, France, 2019, pp. 478-483. Publisher: IEEE. `https://doi.org/10.1109/INFCOMW.2019.8845278`. *Ranking: A* (CORE 2018).*
  - **Armando N.**, Raposo D., Fernandes J.M., Rodrigues A., Sá Silva J., Boavida F., WSNs in FIWARE Towards the Development of People-Centric Applications. in Proceedings of PAAMS, Communications in Computer and Information Science, vol 722, pp. 445-456, Porto, Portugal. Publisher: Springer. `https://doi.org/10.1007/978-3-319-60285-1_38`. *Ranking: C (CORE 2018).*

## ▪ National Conference Articles (1)

Sharma R., Ribeiro B., Pinto A.M., Cardoso F.A., Raposo D., **Armando N.**, Rodrigues A., Sá Silva J., Gonçalo Oliveira H., Macedo L., Boavida F., "Unveiling Markers of Stress Via Smartphone Usage", in Proceedings of RecPad2018 - The 24th Portuguese Conference on Pattern Recognition, Coimbra, Portugal, 2018, pp. 129-131. Publisher: APRP `http://www.aprp.pt/wp-content/uploads/proceedings_recpad2018.pdf`.

# Contents


**Acknowledgements** **xi**

**Abstract** **xiii**

**Resumo** **xv**

**Foreword** **xvii**

**List of Figures** **xxiv**

**List of Tables** **xxv**

**Listings** **xxvii**

**Acronyms** **xxix**

**1 Introduction** **1**
1.1 Motivation . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
1.2 Research Context . . . . . . . . . . . . . . . . . . . . . . . . . . 4
1.2.1 Problem Statement . . . . . . . . . . . . . . . . . . . . . . 4
1.2.2 Research Hypothesis . . . . . . . . . . . . . . . . . . . . . 4
1.3 Research Objectives and Contributions . . . . . . . . . . . . . . . 4
1.3.1 Concepts and Taxonomies . . . . . . . . . . . . . . . . . . 4
1.3.2 Architectures and Frameworks . . . . . . . . . . . . . . . . 5
1.3.3 Products and Services . . . . . . . . . . . . . . . . . . . . 5
1.4 Research Methodology . . . . . . . . . . . . . . . . . . . . . . . 5
1.5 Outline of the Dissertation . . . . . . . . . . . . . . . . . . . . . 6

**2 Literature Review** **7**
2.1 Methodology . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
2.2 State of the Art . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
2.2.1 Dealing with Heterogeneity in IoT . . . . . . . . . . . . . . 10
2.2.2 Social Sensing Paradigm . . . . . . . . . . . . . . . . . . . 15
2.3 Open Issues and Research Opportunities . . . . . . . . . . . . . . 17
2.3.1 Electronic-Based Sensing . . . . . . . . . . . . . . . . . . . 17
2.3.2 Software-Based Sensing . . . . . . . . . . . . . . . . . . . . 18
2.3.3 Human-Based Sensing . . . . . . . . . . . . . . . . . . . . 19
2.3.4 Cross-domain Challenges . . . . . . . . . . . . . . . . . . . 19
2.4 Summary of the Chapter . . . . . . . . . . . . . . . . . . . . . . 21

**3 Unified Management of Traditional and "New Devices" in IoT 25**
3.1 Generic Design . . . . . . . . . . . . . . . . . . . . . . . . . . 26
3.1.1 IEEE 1451 Standards . . . . . . . . . . . . . . . . . . . . . 26
3.1.2 Extending the Basic Sensing Loop in IoT . . . . . . . . 28
3.1.3 Related Work . . . . . . . . . . . . . . . . . . . . . . . . . 32
3.1.4 Proposed Architecture . . . . . . . . . . . . . . . . . . . . 35
3.2 Off-the-Shelf Technologies Testbed . . . . . . . . . . . . . . . 36
3.3 Standard-Based Frameworks Scenario . . . . . . . . . . . . . . . 37
3.4 Enabler for 5G Product and Services . . . . . . . . . . . . . . . 39
3.5 Summary of the Chapter . . . . . . . . . . . . . . . . . . . . . . 43

**4 Engineering of the Testbed and Case Studies 45**
4.1 Off-the-Shelf Testbed . . . . . . . . . . . . . . . . . . . . . . . 46
4.1.1 Protocols and Technologies . . . . . . . . . . . . . . . . . 46
4.1.2 Development and Configurations . . . . . . . . . . . . . 47
4.2 ISABELA . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.2.1 Ecosystem . . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.2.2 Device Models, Configurations and Code adaptations . . 63
4.2.3 Ultralight versus LwM2M . . . . . . . . . . . . . . . . . . 69
4.3 5GOpenclasses . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
4.3.1 Backend . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
4.3.2 Frontend . . . . . . . . . . . . . . . . . . . . . . . . . . . 77
4.4 Summary of the Chapter . . . . . . . . . . . . . . . . . . . . . . 78

**5 Obtained Results and Discussion 79**
5.1 Management Functionalities . . . . . . . . . . . . . . . . . . . . 80
5.1.1 Off-the-Shelf Testbed . . . . . . . . . . . . . . . . . . . . 80
5.1.2 ISABELA Case Study . . . . . . . . . . . . . . . . . . . . 82
5.2 Ultralight versus LwM2M . . . . . . . . . . . . . . . . . . . . . 87
5.2.1 Used Bandwidth . . . . . . . . . . . . . . . . . . . . . . . 87
5.2.2 Delays . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
5.2.3 Loss Ratio . . . . . . . . . . . . . . . . . . . . . . . . . . 91
5.2.4 Summarizing the Comparison . . . . . . . . . . . . . . . 93
5.3 Summary of the Chapter . . . . . . . . . . . . . . . . . . . . . . 95

**6 Conclusions and Future Work 97**
6.1 Synthesis . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98
6.2 Research Opportunities and Future Work . . . . . . . . . . . . . 100
6.2.1 A Model for Privacy Management . . . . . . . . . . . . . 100
6.2.2 Opportunistic Selection of the IoTagents . . . . . . . . . 100

**Bibliography 101**

**Appendices 111**

# List of Figures


1.1 Generic sensing loop in IoT. *Adapted from [Steve Liang, 2014]* . 3

2.1 Results of the PRISMA data flow . . . . . . . . . . . . . . . . 9
2.2 Technical overview of IoT enablers. *Adapted from [ITU-T Study Group 20, 2012]* . . . . . . . . . . . . . . . . . . . . . . . . . . 11
2.3 Heterogeneity in IoT and typical usage scenarios . . . . . . . . 13
2.4 Main open issues per sensing approach and crossed open issues . 18

3.1 Standardized smart transducer scheme. *Adapted from [Lee, 2002]* 27
3.2 Electronic-based sensing loop . . . . . . . . . . . . . . . . . . . 30
3.3 Software-based sensing loop . . . . . . . . . . . . . . . . . . . . 30
3.4 Human-based sensing loop . . . . . . . . . . . . . . . . . . . . . 30
3.5 IoT management reference model. *Adapted from [ITU-T Study Group 20, 2012]* . . . . . . . . . . . . . . . . . . . . . . . . . . 31
3.6 LwM2M architecture. *Adapted from [OMA, 2018]* . . . . . . . 34
3.7 Generic architecture for unified management . . . . . . . . . . . 35
3.8 Setup for the off-the-shelf test-bed . . . . . . . . . . . . . . . . . 36
3.9 Setup with standard-based frameworks . . . . . . . . . . . . . . 38
3.10 4G capabilities versus 5G. *Adapted from [ITU-T Study Group 5, 2015]* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
3.11 Setup for 5G Product and Services . . . . . . . . . . . . . . . . 41

4.1 Technologies in ISABELA. *Adapted from [Fernandes et al., 2019, 2020]* . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
4.2 ISABELA data flow . . . . . . . . . . . . . . . . . . . . . . . . . 56
4.3 Setup for the integration of legacy technologies . . . . . . . . . . 58
4.4 NLP generic architecture . . . . . . . . . . . . . . . . . . . . . . 60
4.5 Sentiment Analysis module . . . . . . . . . . . . . . . . . . . . . 62
4.6 Ultralight versus LwM2M. LAN performance setup . . . . . . . 69
4.7 Ultralight versus LwM2M. Laptop performance setup . . . . . . 71
4.8 *5GOpenclasses* test-cases scenario . . . . . . . . . . . . . . . . 75
4.9 Capabilities of the Kurento media server. Adapted from [Revision, 2020] . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76

5.1 Interface and results of unified management testbed . . . . . . . 81
5.2 Retrieved tweet objects saved in FIWARE . . . . . . . . . . . . 82
5.3 Sentiment Analysis visualizations . . . . . . . . . . . . . . . . . 83
5.4 Scenario for unified management use-cases . . . . . . . . . . . . 84
5.5 Results of the "Sample Frequency" management . . . . . . . . . 85

5.6 Addressable parameters in Software-based entities . . . . . . . . 86
5.7 Blocking notifications . . . . . . . . . . . . . . . . . . . . . . . . . 86
5.8 Occupied bandwidth . . . . . . . . . . . . . . . . . . . . . . . . . 87
5.9 Delay distributions for mobile scenarios . . . . . . . . . . . . . . . 88
5.10 Delay distributions for stationary scenarios . . . . . . . . . . . . 89
5.11 E2E packet loss . . . . . . . . . . . . . . . . . . . . . . . . . . . 91
5.12 Regular activities and downtimes. Stationary scenario . . . . . . 92
5.13 Regular activities and downtimes. Mobile scenario . . . . . . . . 93
5.14 Ratios Ultralight to LwM2M. . . . . . . . . . . . . . . . . . . . . 94

# List of Tables


1.1 Simplified view of the IoT layered model. *Adapted from [ITU-T Study Group 20, 2012]* . . . 3

2.1 Taxonomy of sensors and actuators . . . 12
2.2 Relevant work according to the interacting world . . . 14
2.3 Relevant work according to the built-in nature . . . 14
2.4 Relevant work according to the assigned relationship capabilities 14
2.5 Main open challenges identified in the literature . . . 22

3.1 Capabilities of LwM2M. *Adapted from [Zach Shelby, 2014]* . . . 34

4.1 Structure of static documents in mdb_ebs . . . 48
4.2 Structure of dynamic documents in mdb_ebs . . . 48
4.3 Structure of static documents in mdb_sbs . . . 50
4.4 Structure of dynamic documents in mdb_sbs . . . 50
4.5 Structure of dynamic documents in mdb_hbs . . . 52
4.6 Specifications of the MicaZ - MPR2400 . . . 57
4.7 Device management and application support. *Adapted from M. Koster, 2015* . . . 69
4.8 Timestamps of the captured frames . . . 74

5.1 Delays delta between Ultralight and LwM2M. Mobile scenario . 90
5.2 Delays delta between Ultralight and LwM2M. Stationary scenario 90
5.3 Dataset of the ratios Ultralight to LwM2M . . . 93

# Listings


4.1 Natural Language Processing (NLP) pipeline sample case . . . 61
4.2 Generic device model . . . . . . . . . . . . . . . . . . . . . . . . 64
4.3 Device model for SBSs . . . . . . . . . . . . . . . . . . . . . . . 65
4.4 Device model for HBSs . . . . . . . . . . . . . . . . . . . . . . . 65
4.5 Device model for actuators . . . . . . . . . . . . . . . . . . . . 66
4.6 Configuration of the payload formats in lwm2m-node-lib . . . . 67
4.7 Wireshark packet dissection of an entity creation process . . . . 72

# Acronyms

**3GPP** Third Generation Partnership Project. 42

**4G** Fourth Generation of Mobile Telecommunications. xxiii, 40

**5G** Fifth Generation of Mobile Telecommunications. v, xiv, xvi–xviii, xxii, xxiii, 5, 25, 36, 39–43, 45, 74, 75, 99, 100

**6LowPAN** IPv6 over Low -Power Wireless Personal Area Networks. 28

**6TiSCH** IPv6 over Time Slotted Channel Hopping. 8

**6lo** IPv6 over Networks of Resource-constrained Nodes. 8

**AES** Advanced Encryption Standard. 57

**AI** Artificial Intelligence. 10, 13, 17

**ANEW** Affective Norms for English Words. 61

**API** Application Programming Interface. xiii, xv, 29, 32, 33, 39, 51, 54, 55, 57, 69, 82, 83

**ASCII** American Standard Code for Information Interchange. 67

**AWS** Amazon Web Services. 32, 34

**CISUC** Centre for Informatics and Systems of the University of Coimbra. v, xi, xvii

**CKAN** Comprehensive Knowledge Archive Network. 55

**CoAP** Constrained Application Protocol. 33, 37, 66, 69, 70

**CoMI** CoAP Management Interface. 8, 34

**CORE** Computing Research and Education. xix

**CPS** Cyber-Physical Systems. xix, 13, 20, 27

**CRUDN** Create, Read, Update, Delete, Notify. 35, 39

**DEI** Departamento de Engenharia Informática. xi

**DM** Device Management. xiii, xv, 26

**E2E** End-to-End. xxiv, 70, 72, 90, 91, 93, 94

**EBS** Electronic-Based Sensor. 11, 17, 19, 29, 37, 47, 49–51, 57, 59, 68, 80, 82, 84

**EEPROM** Electrically Erasable Programmable Read-Only Memory. 28, 57

**eMBB** enhanced Mobile Broadband. xvii, 42

**ERA** Excellence in Research for Australia. xix

**ERDF** European Regional Development Fund. v

**ESPU** Escola Superior Politécnica do Uíge. v, xi

**ETSI** European Telecommunications Standards Institute. 32

**FCT** Portuguese Foundation for Science and Technology. v

**FI-PPP** Future Internet Public-Private Partnership Programme. 54

**FIWARE** Future Internet - Ware. xiii, xiv, xvi, xix, xxiii, 5, 32, 33, 37, 39, 42, 43, 54, 55, 57–59, 62, 63, 66, 68–71, 74, 76, 77, 81, 82, 87, 89, 91, 92, 95, 99, 100

**FP7** Seventh Framework Programme. 54

**GDPR** General Data Protection Regulation. 82, 100

**GE** Generic Enabler. 55, 57, 63, 70, 71

**GNSS** Global Navigation Satellite System. 54

**GPIO** General-Purpose Input/Output. 50

**HBS** Human-Based Sensor. xiii, xv, xxvii, 29, 31, 37, 51, 52, 65, 80, 82–84, 86

**HiTLCPS** Human-in-The-Loop Cyber-Physical Systems. xviii, 54, 55, 57, 59

**HTTP** HyperText Transfer Protocol. 33, 39, 59, 69, 70, 76, 92

**I2C** Inter-Integrated Circuit. 28, 57

**IDE** Integrated Development Environment. 46, 48, 59

**IEEE** Institute of Electrical and Electronics Engineers. xviii, xix, xxii, 8, 10, 25–28, 33, 57, 63, 69

**IETF** Internet Engineering Task Force. 32, 34

**INAGBE** Angolan National Grant Institute. v

**IoRT** Internet of Robotic Things. 13, 16, 17

**IoT** Internet of Things. v, xiii–xix, xxi–xxiii, xxv, xxxii, 2–13, 15, 19–22, 25–29, 31–37, 39, 42, 43, 47–50, 52–55, 57, 60, 63–66, 68–70, 74, 80, 81, 94, 97–100

**IP** Internet Protocol. 33, 50, 69

**IPSO** IP for Smart Objects. 49, 63, 64, 66, 73

**ISABELA** IoT Student Advisor and Best Lifestyle Analyzer. xiv, xvi, xxii, xxiii, 5, 37, 39, 42, 43, 45, 52–57, 59, 60, 62, 63, 65, 66, 68, 69, 79, 81–85, 87, 100

**ISO** International Organization for Standardization. 32, 77

**IT** Information Technologies. xiii, 19, 34, 100

**ITU-T** International Telecommunication Union - Telecommunication Standardization Sector. xiii, xv, 2, 3, 10, 12, 31–33, 80

**JSON** JavaScript Object Notation. 33, 34, 39, 47, 55, 62, 63, 69

**LAN** Local Area Network. xxiii, 69, 95

**LCT** Laboratory of Communications and Telematics. xvii

**LED** Light-Emitting Diode. 12, 59, 65

**LEED** Lisbon Emoji and Emoticon Database. 61, 84

**LoRaWAN** Low Range Wide Area Network. 33

**LTE** Long-Term Evolution. 33

**LwM2M** Lightweight Machine-To-Machine. xiii, xiv, xvi, xix, xxii–xxv, 5, 33–37, 39, 45, 49, 51, 63, 66–72, 74, 80, 89–95, 99, 121

**M2M** Machine-to-Machine. 32, 33, 35, 40

**MCU** Microcontroller unit. 57

**MCUn** Multipoint Control Unit. 75

**MEC** Multi-access Edge Computing. 20, 100

**MEMS** MicroElectroMechanical Systems. 2, 11

**MIPS** Millions of Instructions Per Second. 57

**mMTC** massive Machine-Type Communication. xvii, 42

**MPSS** Mobile Phones Social Sensors. 19, 23

**MQTT** Message Queueing Telemetry Transport. 33

**NCAP** Network Capable Application Processor. 27–29, 35, 74

**NGSI** Next-Generation Service Interface. 37, 39, 55

**NLP** Natural Language Processing. xxiii, xxvii, 11, 37, 39, 42, 60–62, 81, 99

**OCF** Open Connectivity Foundation. 33

**OGC** Open Geospatial Consortium. 10, 32, 33

**OMA** Open Mobile Alliance. 32, 33, 49, 55, 66

**OS** Operating System. 46, 57–59, 66, 77, 92

**OSI** Open Systems Interconnection Model. 91

**OSN** Online Social Network. xiii, xv, 2, 12, 15–17, 19, 23, 29, 37, 39, 54, 59, 60, 62, 65, 81–84, 87, 98

**P2P** Peer-to-Peer. 42

**PNFQ** Plano Nacional de Formação de Quadros. v

**PoC** Proof-of-Concept. 36, 37, 43, 52, 80, 99

**POCI** Operational Program for Competitiveness and Internationalization. v

**POS** Part-of-Speech. 60

**PPS** Processes, Product and Services. xvii, 39, 40

**PPUAP** Privacy-Preserving User Authentication Protocol. 43

**PRISMA** Preferred Reporting Items for Systematic Reviews and Meta-Analyses. xxiii, 5, 9

**QoE** Quality of Experience. 42, 77

**QoS** Quality of Service. 16, 70, 91, 95, 99

**RAN** Radio Access Network. 42

**REST** Representational State Transfer. 32, 55

**RS** Recommended Standard. 28

**SA** Sentiment Analysis. xxiii, 12, 46, 51, 52, 59, 60, 62, 63, 81–83

**SaaS** Sensing-as-a-Service. 11

**SBS** Software-Based Sensor. xiii, xxvii, 29, 37, 50, 51, 59, 64, 65, 80, 82

**SDK** Software Development Kit. 64, 65, 69

**SIoT** Social IoT. 15

**SJR** SCimago - Scientific Journal Rankings. xviii

**SMS** Short Message Service. 54

**SOCIALITE** Social-Oriented IoT Architecture, Solutions and Environment. v, xvii

**SPI** Serial Peripheral Interface. 28, 57

**SSH** Secure Shell. 58, 59

**STA** SensorThings API. 33

**STH** Short-Term History. 55, 71, 82, 85

**SVM** Support-Vector Machine. 12, 17

**TCP** Transmission Control Protocol. 33, 69, 70, 91

**TEDS** Transducer Electronic Data Sheets. 28, 29, 48, 63

**TIA** Telecommunications Industry Association. 32

**TIM** Transducer Interface Module. 27–29

**TLV** Type-Length-Value. 67

**TRL** Technology Readiness Level. xvii, 39, 74, 100

**UART** Universal Asynchronous Receiver Transmitter. 28, 57

**UAV** Unmanned Aerial Vehicles. 43, 75, 78

**UDP** User Datagram Protocol. 33, 69, 70, 91

**UE** User Equipment. 42, 75

**UNIKIVI** Universidade Kimpa Vita. v, xi

**URL** Uniform Resource Locator. 48, 50, 68

**uRLLC** ultra-Reliable and Low Latency Communications. xvii, 42, 100

**USB** Universal Serial Bus. 57, 59

**VADER** Valence Aware Dictionary and Sentiment Reasoner. 63

**VM** Virtual Machine. 42, 57–59, 70

**VNF** Virtual Network Functions. 100

**WebRTC** Web Real-Time Communication. 75–78

**WSAN** Wireless Sensor and Actuator Networks. 2, 11, 18, 24, 58, 59

**XML** eXtensible Markup Language. 33, 49

# Chapter 1

# Introduction

## Contents


**1.1 Motivation** . . . . . . . . . . . . . . . . . . . . . . . . . **2**

**1.2 Research Context** . . . . . . . . . . . . . . . . . . . . . **4**

1.2.1 Problem Statement . . . . . . . . . . . . . . . . . . . 4

1.2.2 Research Hypothesis . . . . . . . . . . . . . . . . . . 4

**1.3 Research Objectives and Contributions** . . . . . . . **4**

1.3.1 Concepts and Taxonomies . . . . . . . . . . . . . . . 4

1.3.2 Architectures and Frameworks . . . . . . . . . . . . 5

1.3.3 Products and Services . . . . . . . . . . . . . . . . . 5

**1.4 Research Methodology** . . . . . . . . . . . . . . . . . . **5**

**1.5 Outline of the Dissertation** . . . . . . . . . . . . . . . **6**

The advances in MicroElectroMechanical Systems (MEMS), wireless communications, and digital electronics have enabled the development of sensor nodes that communicate in short distances, triggering the advent of distributed systems called "Wireless Sensor and Actuator Networks (WSAN)" [I. F. Akyildiz et al., 2002]. By connecting WSAN to the Internet, it was created a broader distributed system, empowering the Internet span beyond the connected electronic devices, to the physical world. Nowadays, the end-devices on the Internet are myriad of electronic computing and connected devices labelled "things" [Sousa Nunes et al., 2015]. Thus, the concept of IoT can be defined as an extension of computer networks and the Internet, to both smart and connected "things".

While in 2018, there were more than 22 billion connected devices, this figure may reach over 50 billion by 2030[1]. IoT applications cover daily activities for both monitoring and tracking purposes, such as environmental monitoring, health-care, critical infrastructure protection, automated diagnostics, and military help [Buttyan et al., 2010; Durisic et al., 2012; Giorgetti et al., 2016; O'donovan et al., 2013].

## 1.1 Motivation

In traditional IoT infrastructures, the sensor nodes are typically stationary or mobile devices, wired or wirelessly connected. The sensor nodes collect the features of the environments and report them to a central unit for further processing, storage, and display [I. F. Akyildiz et al., 2002]. The concept of sensing has evolved since then and offered a broader set of possibilities to collect data towards the development of smart applications and services. For instance, the IoT extends the computer's network, conveying any sensed features of the physical world [Sousa Nunes et al., 2015]. Thus, what is generically called "the Internet of *things*" is already the networked connection of physical things and beyond. Indeed, the "new" IoT includes people, processes, data and entities embedded with sensing/actuating capabilities to enable the representation of and to interact with entities [Melcherts, 2017].

As displayed in Figure 1.1, the generic sensing loop in today's IoT involves a variety of sensors, comprising not only physical, electronic-based devices, but also virtual sensors (i.e., software agents that abstract one or more physical sensors), and even human sensors, such as human-originated data collected from OSN. Such heterogeneity of "devices" has an impact on the ITU-T reference model in Table 1.1. Thus, it shall be ensured a redefinition of concepts already established in the device layer.

[1] `https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/`, accessed on 2020-02-19

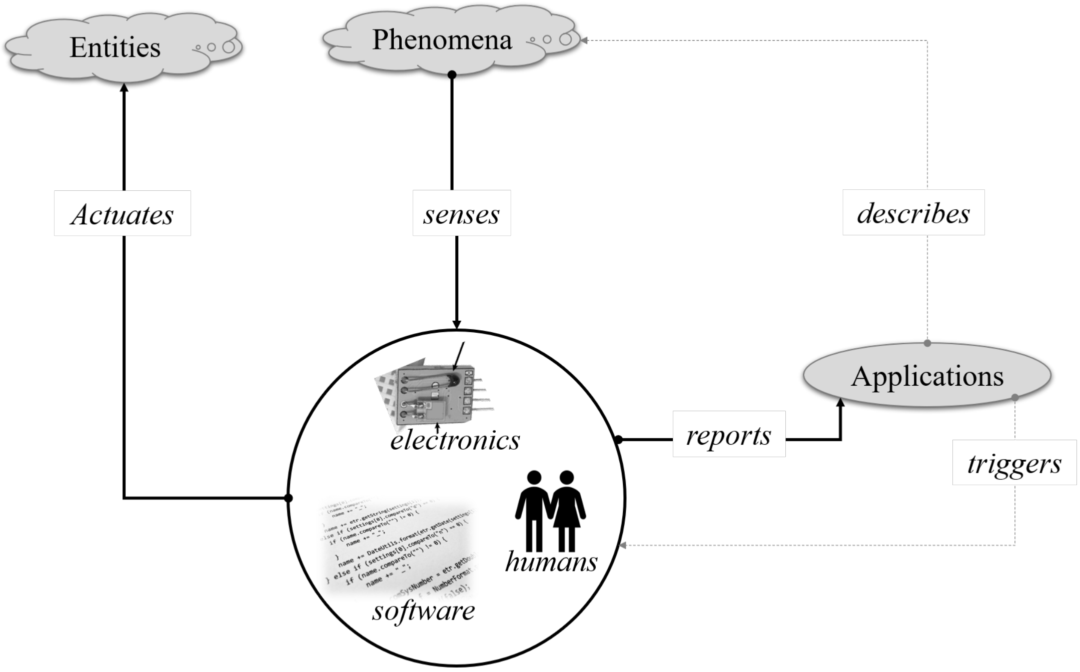



Figure 1.1: Generic sensing loop in IoT. *Adapted from [Steve Liang, 2014]*

Table 1.1: Simplified view of the IoT layered model. *Adapted from [ITU-T Study Group 20, 2012]*

| **Layers and Sub-layers** | | **Some Enablers** |
|---|---|---|
| Application | | The user interface, services |
| Middleware | | Modelling languages, data models |
| Network | | Nodes, communication supports |
| Device | Networking capability | Gateways proxies |
| | Sensing/actuating capabilities | Smart sensors/actuators |

The natural heterogeneity of hardware, software, technologies and protocols in IoT is a crucial factor in enhancing both resilience and lifetime of the network [Giorgetti et al., 2016; Oteafy and Hassanein, 2016]. To this end, heterogeneity must be driven by common specifications so that all sensor nodes will be seen as a single computing platform at the disposal of applications and services developers. Therefore, the integration of such heterogeneity both in type and forms demands a broader consensus through management protocols toward efficient management capabilities. As for the ITU-T management reference model [ITU-T Study Group 20, 2012], the management capabilities are cross-layer components that cover fault, configuration, accounting, performance and security purposes.

## 1.2 Research Context

### 1.2.1 Problem Statement

In this context, the main goal of our research was to provide answers to the following questions:

- How can we efficiently and effectively manage different types of sensors, including physical, virtual and social sensors?
- Can existing IoT management protocols be used for management in such a broad view of entities on the device layer?

### 1.2.2 Research Hypothesis

As we have surveyed in [Armando et al., 2018], there is not *an approach to the unified management of heterogeneous IoT environments* that derives from having physical, virtual, and human-based entities in the perception layer.

We knew that an IoT middleware provides a connectivity layer so that the components can communicate regardless of the specific modelling languages and data models of each one. A middleware is a software framework that enables the abstraction of both technologies and protocols from different components of a communication architecture. Therefore, we believed that the combination of middleware's flexibility, with the robustness of well-established management standards, was crucial in designing an approach to unified solution. This solution could enable the management of both data and the broad view of IoT "devices".

## 1.3 Research Objectives and Contributions

Our research objective is to propose and validate an architecture for unified management of the broad IoT device layer. Our contributions span into three areas, described in the following subsections.

### 1.3.1 Concepts and Taxonomies

We provided a survey on the sensing/actuating approaches for the IoT at large, and we proposed a taxonomy to deal with the heterogeneity of sensing/actuating forms in today's IoT. Our survey was conducted with relevant tools for a systematic review of the literature, covering studies from late 2015 to 2017.

We also provided a comprehensive survey on management protocols and frameworks for IoT, from late 1980 to 2019. We identified remaining challenges and solutions offered by new management protocols, which have not been covered by previous studies.

### 1.3.2 Architectures and Frameworks

We wanted to explore paradigms that leverage new models of sensing under normalized protocols for networking and management in IoT.

Therefore, we proposed an extension of the device layer in [ITU-T Study Group 20, 2012], which includes traditional electronic and both software and human-based entities.

We proposed and validated a generic architecture for unified management of the entities in the extended IoT device layer. Our architecture adopted relevant open standards for both data and device management.

We extensively assessed the LwM2M 1.0 and Ultralight 2.0 services in FIWARE middleware, as both are relevant support protocols for management purposes in IoT.

### 1.3.3 Products and Services

We designed, developed and validated a series of products and services under the umbrella of the generic architecture of a unified management. Theses solutions are the foundations of two case studies, namely ISABELA and *5GOpenclasses*.

## 1.4 Research Methodology

We started our research by analysing the literature, and we studied its trends in a systematic way that can be reproduced by our peers. Our methodology for the review process was based on the practices described in [Faculty Librarian et al., 2009; Liberati et al., 2009; Tranfield et al., 2003], including the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) Statement. To this end, the framework of the review process was specified in the protocol document.

Once we have identified the gaps in the literature, we filtered them to select the research opportunity that was most relevant for our team. With the research topic in mind, we designed the set of research objectives which achievements have been based on an experimental approach [Knight, 2012]. We explored the experimental approach to validate the solutions that we proposed in an environment of real-case studies.

As for the evaluation methodology, we used both qualitative and quantitative approaches. The former approach enabled us to know how well the solution we propose worked in terms of functionalities for the end-users. The latter approach enabled us to measure the performance of the frameworks that supported the solution we propose.

Finally, we leveraged as much as possible the existing state-of-the-art frameworks. To summarize, our research methodology was aligned in the following sequential list of steps:

- Review of the literature to identify open issues and challenges;
- Selection of one or two open issues identified in the literature according to the needs in our research group;
- A suggestion of approaches to address the identified issues;
- Design of a generic prototype to address the research problems;
- Integration, assessment, and validation of the prototype in both test-beds and real scenario case studies.

## 1.5 Outline of the Dissertation

Besides the introduction, this thesis is organized into five other chapters.

In Chapter 2, we describe the literature review we made to identify the challenges that we tackled during the thesis. In chapter 3, we present an outlook on the fundamental concepts in what we define as "extended IoT". Then, we explain our proposal of a generic architecture for a unified management approach beyond traditional devices. We finally present the prototype testbed and applied case studies that served for assessing the proposed architecture. In Chapter 4, we describe the frameworks, technologies and tools that we exploited for the development of both the testbed and case studies. In chapter 5, we present the main results obtained in terms of functionalities, performances and their limitations. Finally, in Chapter 6, we summarize our thesis and present the research opportunities that we have identified as future work.

# Chapter 2

# Literature Review

## Contents


**2.1 Methodology** . . . . . . . . . . . . . . . . . . . . . . . . **8**

**2.2 State of the Art** . . . . . . . . . . . . . . . . . . . . . . . **10**

2.2.1 Dealing with Heterogeneity in IoT . . . . . . . . . . 10

2.2.2 Social Sensing Paradigm . . . . . . . . . . . . . . . . . 15

**2.3 Open Issues and Research Opportunities** . . . . . . **17**

2.3.1 Electronic-Based Sensing . . . . . . . . . . . . . . . 17

2.3.2 Software-Based Sensing . . . . . . . . . . . . . . . . 18

2.3.3 Human-Based Sensing . . . . . . . . . . . . . . . . . 19

2.3.4 Cross-domain Challenges . . . . . . . . . . . . . . . 19

**2.4 Summary of the Chapter** . . . . . . . . . . . . . . . . **21**

Considerable work has been done on the sensing/actuating IoT sublayer (Table 1.1), either separately or exploring synergies between approaches. In this chapter, we present the survey we made of the concerned literature, with emphasis on the 2015 to 2017 period. Due to the amount and relevance of the work underway, it was essential to, on the one hand, survey existing initiatives and ongoing research and, on the other hand, identify the main challenges and research guidelines. These are the objectives of the current chapter, which main contributions are:

- a taxonomy for sensing/actuating entities in IoT, to establish a clear relationship between them;
- an analysis of the state-of-the-art on social sensing;
- identification of the leading open issues in what concerns the use of sensing in a socially interactive Internet, as well as the resulting research opportunities.

We kept a concise description of our methodology, presenting the most critical points specified in *Appendix A - The Protocol of the Systematic Review.* The detailed data-extraction forms of the review process are available online at the following URL: ***https://goo.gl/pVvlqt***.

## 2.1 Methodology

We started browsing the most relevant outcomes concerning the new sensing models from two research conferences, namely *IoT Design and Implementation*, and *SocialSens*. After browsing the main topics of both venues, we carried on searches in the database where their proceedings are published, namely the *Institute of Electrical and Electronics Engineers (IEEE) Xplore* and *ACM Digital Library*. Complementary searches to retrieve additional materials were also carried on using *Google Scholar*. For the review process, we have considered studies that proposed, evaluated, and described approaches in the IoT field, to enable the contextualization of the physical environments. The initial time scope was limited to materials from 2013 to 2017.

Both published and unpublished documents studies were considered, with priority to the former. Concerning the support of the materials, we mostly explored electronic sources online via the *Biblioteca do Conhecimento Online* signature of the University of Coimbra (`https://www.b-on.pt/`). We used the following primary search terms: "Virtual Sensor", "Virtual Sensing", "Social Sensor", "Social Sensing", "Fog computing", "Virtual Sensor Networks", "IPv6 over Time Slotted Channel Hopping (6TiSCH)", "IPv6 over Networks of Resource-constrained Nodes (6lo)", and "CoAP Management Interface (CoMI)". A list of search strings was also used to refine as much as possible, the studies that potentially tackled the objectives of our review.

The details of the PRISMA data flow in Figure 2.1 are reported in data-extraction forms available online at: `https://goo.gl/W5ATRz`. As depicted in Figure 2.1 The strategy for the selection of the articles included in this survey was conducted following the 4-phased PRISMA flow diagram. The first is the Identification phase, which consists in retrieving the citations from the data sources, based on both the search terms and strings. The last phase in the PRISMA flow diagram is called "Inclusion". The *unpublished* report for our survey included 106 materials divided into sixty-eight (68) items from the *Eligibility* Phase, fifteen (15) items from temporarily discarded ones during the *Eligibility* Phase, nine (9) items from our previous pre-review searches on IoT fields, and fourteen (14) on-the-fly items, according to the topic developed in the sections of the final report. To summarize the findings of the review, the results have been pooled, then subjected to a meta-synthesis process from where we produced one set of synthesized findings. A "Meta-synthesis provides a means of considering all significant similarities and differences in language, concepts, images, and other ideas around a target experience" [Tranfield et al., 2003].

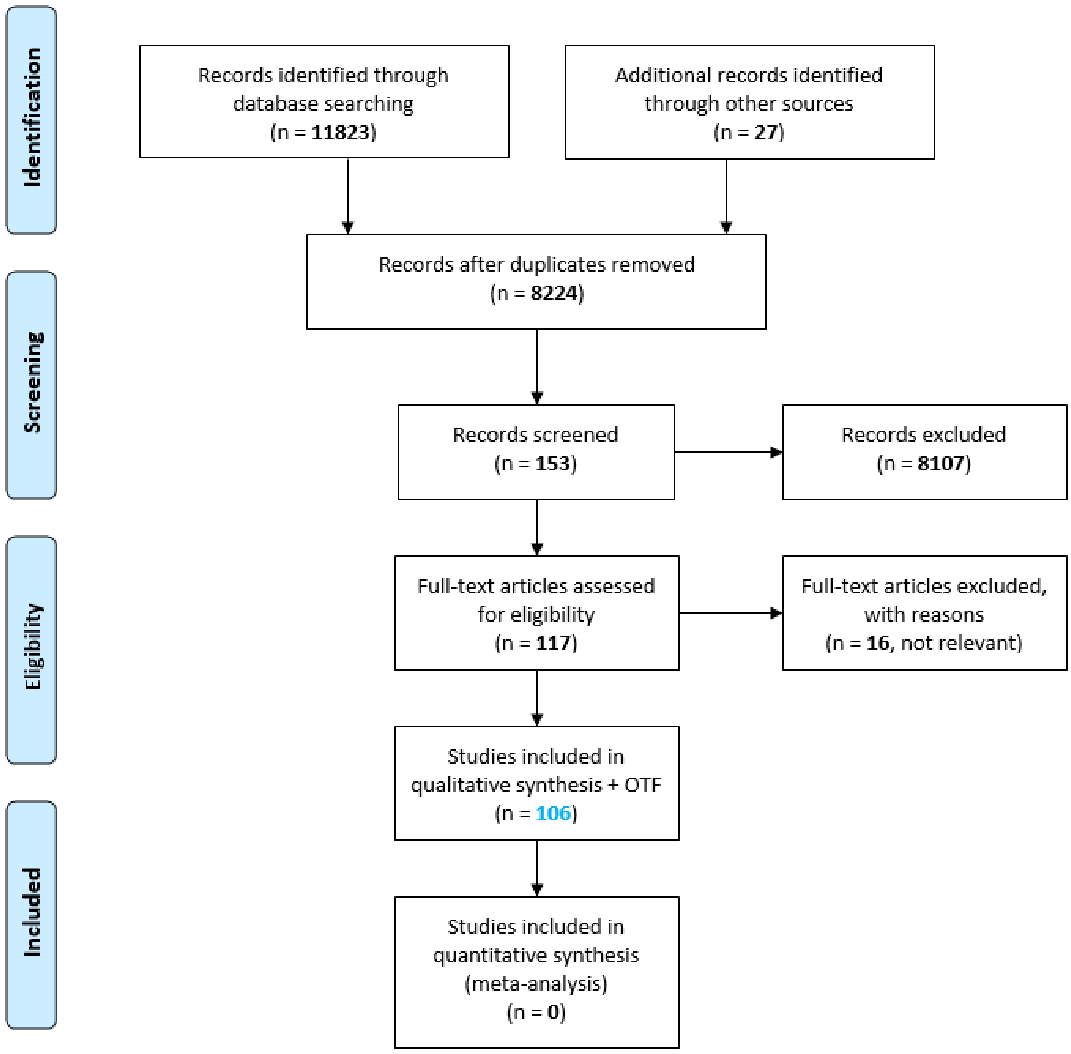



Figure 2.1: Results of the PRISMA data flow

## 2.2 State of the Art

Smart sensors, hereafter only referred to as sensors, are the IoT's founding blocks, which core function is to gather data on the target entities. The collected data is sent to processing units where simple to sophisticated approaches such as Rule-based Decision Trees, Fuzzy Logic and Artificial Intelligence (AI) algorithms, will enable the representation of the target entities in an information system. These approaches can also enable the inference of the context and, eventually, trigger some actuating devices.

In the next subsection, we address the heterogeneity of sensing/actuating devices in an Internet made of things, computer resources, and people. We also present a taxonomy for IoT devices that derives from a thorough analysis of the existing literature.

### 2.2.1 Dealing with Heterogeneity in IoT

#### 2.2.1.1 Sensors and Actuators

With reference to IoT's communication model in Table 1.1 and the IEEE 1451 Standards Family [Kumar et al., 2015; Lee, 2002], our study is focused on the sensing/actuating sublayer. Therefore, we tackle both conceptual and implementation approaches to describe the heterogeneity of entities in this sublayer. Also, by leveraging the potential of socialisation between entities as defined in [Atzori et al., 2012], we propose a taxonomy for the heterogeneity of sensing/actuating entities. For the proposed taxonomy, we firstly organize the entities according to the target world where their activities are intended to be performed, i.e., into physical or virtual worlds. Secondly, we regroup entities according to their built-in nature, i.e., into electronic-based, human-based or software-based. We finally classify entities according to their interaction ability, i.e., stand-alone or social entities.

In what concerns the main entities in the sensing/actuating sublayer, we use the definitions in the Open Geospatial Consortium (OGC) standards [Botts, Michael E. and Robin, Alexandre and Greenwood, Jim and Wesloh, David, 2014]. Thus, a sensor is an entity that retrieves the state of the sensed object and then pushes the collected data to one central processing and/or storage unit. An actuator is an entity that receives commands from a processing unit and executes an action on physical or virtual/information objects. As seen in Figure 2.2, the technical overview of IoT enablers recommended by the ITU-T already identifies networked "devices" whose activities can be carried out in both physical and virtual worlds [ITU-T Study Group 20, 2012].

In Figure 2.2, virtual means *"in the realm of the pure information world"*. A web of physical and virtual sensing devices enables us to have a virtual representation of the physical world. For instance, consider the use of sensors to build a representation of air temperature variation in a given zone or the use of

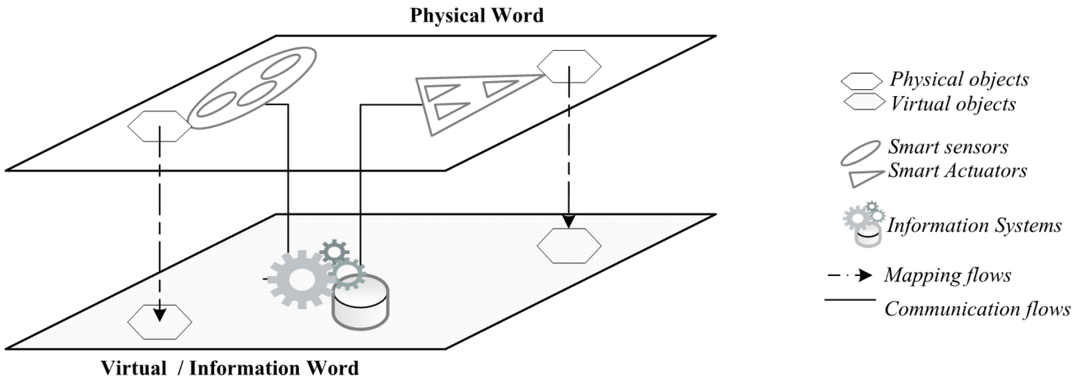



Figure 2.2: Technical overview of IoT enablers. *Adapted from [ITU-T Study Group 20, 2012]*

NLP techniques on written text in order to build a representation of someone's traits of personality. In these scenarios, we may use weather electronics, software or humans as front-end "IoT devices" to interact with the target world. The activities in the physical world are mainly associated with traditional electronic devices in WSAN [I. F. Akyildiz et al., 2002]. These devices are purely based on MEMS to interact with the sensed object. However, humans are also able to objectively report some phenomena of the physical world, like *how pleasant the weather is*, or *how empty/full a bin really is.*

Data from Electronic-Based Sensors (EBSs) can be used as it is by the processing unit or can be fed into what is known in the literature as "virtual sensors". Indeed, contrary to both electronic and human entities, the activities in the virtual world are associated with software applications, since they are capable of reporting events in the information world. [Nitti et al., 2016] surveyed the virtual objects in IoT from the perspective of the device, and even if the complexity of proposing a "standardized" one is still recognized, they give an interesting definition of "Virtual Objects". To the authors, virtual objects are context-aware representations of the physical-electronic nodes in the real world, which augment the potentialities of the associated services. "Virtual Objects" are strategic in the development of complex applications, addressing both heterogeneity and scalability issues in IoT.

There are, mainly, three types of representation models for virtual sensors. The first one is the Sensing-as-a-Service (SaaS) model [Abdelwahab et al., 2016; Gupta and Mukherjee, 2016; Khan et al., 2016], where both electronic nodes and their sensed data are put at the disposal of remote subscribers and operators. The second one is the estimation model [Cardell-Oliver and Sarkar, 2016; Li et al., 2016]. Here, the virtual state of the environment is the result of a function that estimates its status based on a variety of third-party sensor inputs. The last virtual sensor model is the prediction model [Sarkar et al., 2016; Sousa Nunes et al., 2015]. In this case, analytic techniques are used for predicting insights upon the third-party sensors historical data.

Software programs cannot be used to fetch phenomena in humans' minds directly. Therefore leveraging humans themselves arose as an exciting approach to fetch raw data for IoT purposes. In this context, humans can be considered as sensors in their increasingly global societal environment, and are regarded as such in the literature [Psomakelis et al., 2016; Sousa Nunes et al., 2015]. Indeed, by analysing human activities in OSN, for instance, it is possible to infer the emotions and mood of people while they consume sensed data and services [Bachiller et al., 2016]. To this end, tools such as Sentiment Analysis (SA) algorithms [Daniel et al., 2017] and Support-Vector Machine (SVM) techniques [Nakashima et al., 2016], enable us to give a numerical translation to the sensed data, that is, the text or any multimedia contents in a post.

Conversely, the IoT design in Figure 2.2 presents a web of actuating devices that can be used in both worlds. In fact, mechanical and electronic actuators can be looked at as the counterparts of electronic sensors. Also, computational actuators can be looked at as the counterparts of software sensors. Finally, human action can be looked at as the counterpart of human sensors. It is essential to consider the actuator concept in its broad sense by adopting a variety of ways for its implementation. In the traditional IoT context, an actuator may be implemented by mechanical action or even by human action in response to a notification. We can also see an actuator as a pure electronic signal via a Light-Emitting Diode (LED) state or a picture on a screen. Finally, we can consider an actuator as a computational operation to, for instance, fill a database or send notifications to change the value of variables in a remote computer program.

Given the concepts described in this subsection next, we will present our proposed taxonomy for Sensors and Actuators in the "new" IoT.

#### 2.2.1.2 Proposed Taxonomy

The proposed sensors and actuators taxonomy is presented in Table 2.1, considering both the ITU-T overview from Figure 2.2 and the heterogeneity of devices described in section 2.2.1.1.

Table 2.1: Taxonomy of sensors and actuators

<table>
<tr><th>Nature of the Interacting World</th><th>Built-in Nature</th><th>Relationship Capabilities</th></tr>
<tr><td rowspan="2">Physical</td><td>Electronic-based</td><td rowspan="4">Standalone and Social</td></tr>
<tr><td rowspan="2">Human-based</td></tr>
<tr><td rowspan="2">Virtual</td></tr>
<tr><td>Software-based</td></tr>
</table>

Given the mentioned heterogeneity in the IoT device layer, we started by classifying the sensing/actuating devices according to the nature of the entities they are expected to interact with, namely, physical and virtual. As a consequence of considering both physical and virtual worlds, we then observe that sensing/actuating tasks can be performed by electronic-based, software-based, or human-based entities. In future, one may imagine human entities as a subset of a more extensive set. Indeed, we may label them as living-based nature, which will include any living source of information from the surrounding environment integrated into the IoT information system.

Finally, we propose a classification of the IoT device entities according to the relationships they can establish among them. In fact, "things" connected to the Internet are different from the "things" participating in the Internet of social networks [Atzori et al., 2012]. Based on this observation, we can classify things as Stand-alone or Social. Stand-alone entities are those that only establish a direct upstream link to a node in the IoT system. As for social sensors, a relevant feature is that they are both data producers and data consumers. There are two broad classes of social sensors in the literature, which we labelled as "Human social sensors" and "Non-human social sensors". Here, we use human operators to fetch the state of the environment and feed their observations into social networks.

Figure 2.3 illustrates various types of sensor and actuator entities, including some typical usages. On the right, we can see a different type of social entities, in addition to human sensors. Considering the development of AI, we can have robots and computer programs emulating the human's role in such communities. For instance, in bot user systems (we have humans and agents interacting, Cyber-Physical Systems (CPS) cooperating in an Internet of Robotic Things (IoRT) for the Industry 4.0 purposes [Razafimandimby et al., 2016], pure software-based agents for multi-agent system purposes [El Mhouti et al., 2016], and high-frequency trading [Brogaard et al., 2017].

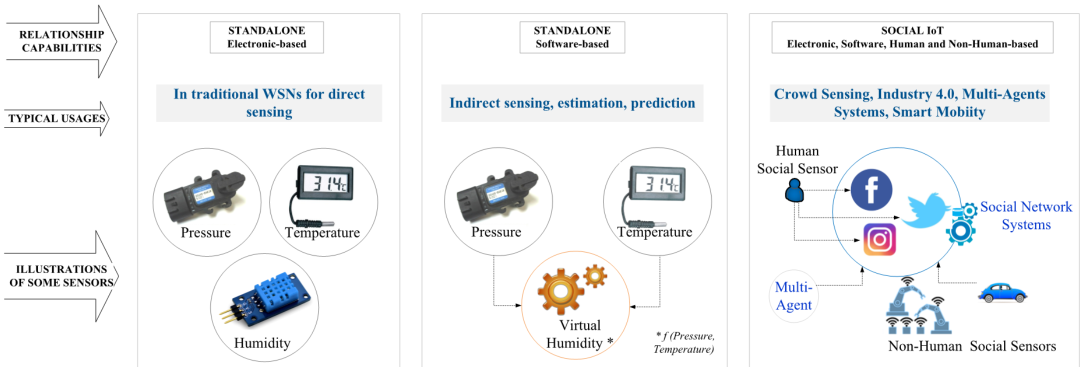



Figure 2.3: Heterogeneity in IoT and typical usage scenarios

Tables 2.2, 2.3 and 2.4 provide some references along the various axes of our proposed taxonomy. Thus, Table 2.2 refers to the target activity object, which can be physical or virtual, while Table 2.3 refers to the built-in nature of the entities, namely, hardware-based, software-based or human-based devices. Finally, Table 2.4 refers to the relationship capabilities of the entities, namely, stand-alone or social devices. Based on these tables, we can see that research is increasingly considering sensing in its broader sense, i.e., looking well beyond the plain IoT, into a socially interactive Internet. In the next section, we will focus our attention on the social sensing paradigm as a way of exploring synergies between the sensing/actuating approaches described in our taxonomy.

Table 2.2: Relevant work according to the interacting world

| **Physical** | **Virtual** |
|---|---|
| [Abdelwahab et al., 2016; Atzori et al., 2012; Bachiller et al., 2016; I. F. Akyildiz et al., 2002; ITU-T Study Group 20, 2012; Jia et al., 2017; Khan et al., 2016; Lee et al., 2015; Melcherts, 2017; Botts, Michael E. and Robin, Alexandre and Greenwood, Jim and Wesloh, David, 2014; Miori and Russo, 2017; Nakashima et al., 2016; Razafimandimby et al., 2016; Saleem et al., 2017; Wang et al., 2015] | [Cardell-Oliver and Sarkar, 2016; Gupta and Mukherjee, 2016; ITU-T Study Group 20, 2012; Li et al., 2016; Sarkar et al., 2016] |

Table 2.3: Relevant work according to the built-in nature

<table>
<tr><td>Electronic-based</td><td colspan="3">[Abdelwahab et al., 2016; Atzori et al., 2012; I. F. Akyildiz et al., 2002; ITU-T Study Group 20, 2012; Jia et al., 2017; Khan et al., 2016; Lee et al., 2015; Melcherts, 2017; Miori and Russo, 2017; Oteafy and Hassanein, 2016; Razafimandimby et al., 2016; Saleem et al., 2017; Wang et al., 2015]</td></tr>
<tr><td>Human-based</td><td colspan="3">[Bachiller et al., 2016; Giridhar et al., 2017; Lee et al., 2015; Nakashima et al., 2016; Psomakelis et al., 2016; Sousa Nunes et al., 2015]</td></tr>
<tr><td rowspan="3">Software-based</td><td>Repository</td><td colspan="2">[Abdelwahab et al., 2016; Gupta and Mukherjee, 2016; Khan et al., 2016; Nitti et al., 2016]</td></tr>
<tr><td rowspan="2">Analytical Results</td><td>Estimation</td><td>[Cardell-Oliver and Sarkar, 2016; Gupta and Mukherjee, 2016; Razafimandimby et al., 2016; Sousa Nunes et al., 2015]</td></tr>
<tr><td>Prediction</td><td>[Gupta and Mukherjee, 2016; Sarkar et al., 2016]</td></tr>
</table>

Table 2.4: Relevant work according to the assigned relationship capabilities

| **Stand-alone** | **Social** |
|---|---|
| [Abdelwahab et al., 2016; Atzori et al., 2012; Gupta and Mukherjee, 2016; I. F. Akyildiz et al., 2002; ITU-T Study Group 20, 2012; Khan et al., 2016; Rui et al., 2016; Yang et al., 2015] | [Bachiller et al., 2016; Giridhar et al., 2017; Jia et al., 2017; Lee et al., 2015; Miori and Russo, 2017; Nakashima et al., 2016; Psomakelis et al., 2016; Razafimandimby et al., 2016; Saleem et al., 2017; Wang et al., 2015] |

### 2.2.2 Social Sensing Paradigm

This subsection provides a focused yet comprehensive review of the Social Sensing paradigm. We start by addressing the definition of the Social IoT (SIoT) concept and then proceed to identify and explain the main pieces of literature that deal with it.

Inspired by human activity in OSNs, social sensors in IoT were proposed at the beginning of the 2010 decade and had, since then, been gaining popularity [Jia et al., 2017; Miori and Russo, 2017; Saleem et al., 2017; Wang et al., 2015]. The SIoT concept is the result of applying the principles of IoT communication and networking to social media. Thus, social sensors are defined in literature considering the members of a community as both data producers and consumers. The Social Sensing paradigm is based on the increasing usage of social media to report the statuses of people and their environment.

Saleem et al. [2017] present a comprehensive state-of-the-art for SIoT and identify the main limitations of existing approaches. In an attempt to overcome current drawbacks, the authors propose the concept of recommendation services among various IoT applications. However, in their sample application scenario, humans are not considered to be both SIoT beneficiaries and enablers at the designed Perception Layer. We reiterate that what is generically called "IoT" is already a networked environment of things and beyond since it includes people, processes, data, and things [Melcherts, 2017]. Thus, nowadays, we believe that the Social Sensing paradigm must involve both non-human entities and humans, as members of a social community.

Human social sensors are members of a community that shares contextualised data via a dedicated application, a web application, or an OSN. Software-based entities are to be included in SIoT since they can have social behaviour, as we saw in section 2.2.1. In SIoT, the trustful entities community is established according to five criteria:

- The *parental relationship*, in which the homogeneous entities originated in the same period by the same manufacturer, can be considered as part of the same community;
- The *co-location relationship*, in which the entities of a delimited environment can be considered as part of the same community;
- The *co-work relationship*, in which the entities that collaborate in the same IoT application can be considered as part of the same community;
- The *ownership relationship*, in which the entities belonging to the same user can be considered as part of the same community;
- The *social relationship*, in which the entities belonging to owners that have a social affinity, e.g., friends, are considered as part of the same community.

Bachiller et al. [2016] propose a middleware solution to model and represent users in social sensing applications. The approach is based on two elements called "User Component" and "User Bindings". The former models the user as a software component, allowing developers to deal with mobility between contexts. The latter enables the middleware to support multiple communication channels, including the association of users with sensing applications, irrespectively of the user network and/or device. According to the authors, humans may contribute to values (as sensors) and actions (as actuators), while also consuming sensing services. Thus, the integration of people into sensing systems has the potential to increase scale and reduce costs. Since an increasing number of people simultaneously use multiple communication devices (e.g., computer, smartphones, tablet), they argue that using OSNs as higher-level communication channels significantly improves the availability of the users in a participatory sensing scenario. In their study, participatory sensing is defined as a way of including users in distributed applications, enabling data to be acquired from both users and mobile phones' sensors.

An IoRT-based neural network control scheme is proposed by Razafimandimby et al. [2016]. The scheme is claimed to maintain the global connectivity efficiently among mobile robots and guarantee the desired Quality of Service (QoS) level. They define IoRT as a set of devices that can monitor events, fuse sensor data, use local and distributed intelligence to determine the best course of action, and then actuate in the physical world. The robots need to establish ad hoc communication with each other, to carry out cooperative tasks. Thus, maintaining communication among multiple mobile IoRT robots is a crucial issue. To this end, the authors use the graph connectivity metric to maintain the global connectivity of IoRT robots' team, when they are in mobility. The authors claim the proposed algorithms allow the whole IoRT robot network to converge to the desired communication quality.

Psomakelis et al. [2016] introduce a platform named "RADICAL" that combines citizens' posts retrieved through smartphone applications and OSNs for smart city services. *RADICAL* enables to collect, combine, analyse, process, visualise, and provide uniform access to big data sets of OSNs content, such as tweets, sensor measurements, or citizens' smartphone reports. In the authors' view, an OSN is a network of recommendations among a circle of "virtual" friends via an online platform.

Lee et al. [2015] combine big data and social sensors to create a novel early warning system for dengue outbreaks. Their starting assumption was that environmental sensors were not as ubiquitously deployed as needed for situational awareness. Thus, to deal with the problem, messages of social sensors (i.e., people) and real-time web information (e.g., tweets) were used to detect the outbreaks. Their approach also leveraged to investigate ways of understanding how the temporal trend of collected data correlates with the incidence of dengue, as identified by national health authorities.

What the authors label as "social messages" are considered to contain information which is valuable for understanding the development of real-world events, thus contributing to solutions for event awareness and crisis management. In the study, the information obtained by social sensors (humans) was used for monitoring outbreak events, in combination with the data collected by EBSs.

Nakashima et al. [2016] suggest that a range of information related to the real world can be retrieved through analysis of texts posted on OSN. They consider the acquisition of such information as "social sensor usage" and refer to the generated data as "social sensor data". The authors designed and implemented a system which generates and shares such social sensor data along with related analytical programs. Here, tweets are classified into various classes using SVM techniques, and additional analytical programs for generating social sensor data were also considered useful in developing new types of social sensors. Their database stores both texts and images, which can be analyzed by AI programs.

Finally, Giridhar et al. [2017] developed an algorithm to identify and geo-locate real-world events that may be presented as social activity signals on OSNs. They focus on content shared by users on both *Twitter* and *Instagram* to design a system capable of fusing data across multiple networks. Their fusion algorithm enables the detection of events from various OSNs to improve the accuracy of results. Here, Twitter users are considered as the sensor nodes that report witnessed events according to some probabilistic distribution. Thus, every tweet is a sensor reading, and the probability of event occurrence is expressed with the help of an exponential distribution. The authors solve the problem of corroboration by trying to map description of the events across the two different OSNs with the support of an unsupervised AI approach. By combining data from multiple social media, they can detect events that may not have enough corroboration in one network or be indistinguishable "irrelevant news" in another.

Given the comprehensive review of the techniques and approaches presented in this section, in the next section, we identify the main pending challenges.

## 2.3 Open Issues and Research Opportunities

As depicted in Figure 2.4, there are challenges for each sensing approach and challenges that pertain to the confluence of the three sensing paradigms.

### 2.3.1 Electronic-Based Sensing

Razafimandimby et al. [2016] proposed a graph connectivity metric to maintain global robot connectivity while mobile. However, the authors admit that maintaining communication among multiple mobile IoRT robots is still a crucial issue. Recent studies point to the use of Cognitive Radio frameworks in the future [Khan et al., 2017], to overcome many challenges concerning the inefficient bandwidth allocation to large numbers of devices.

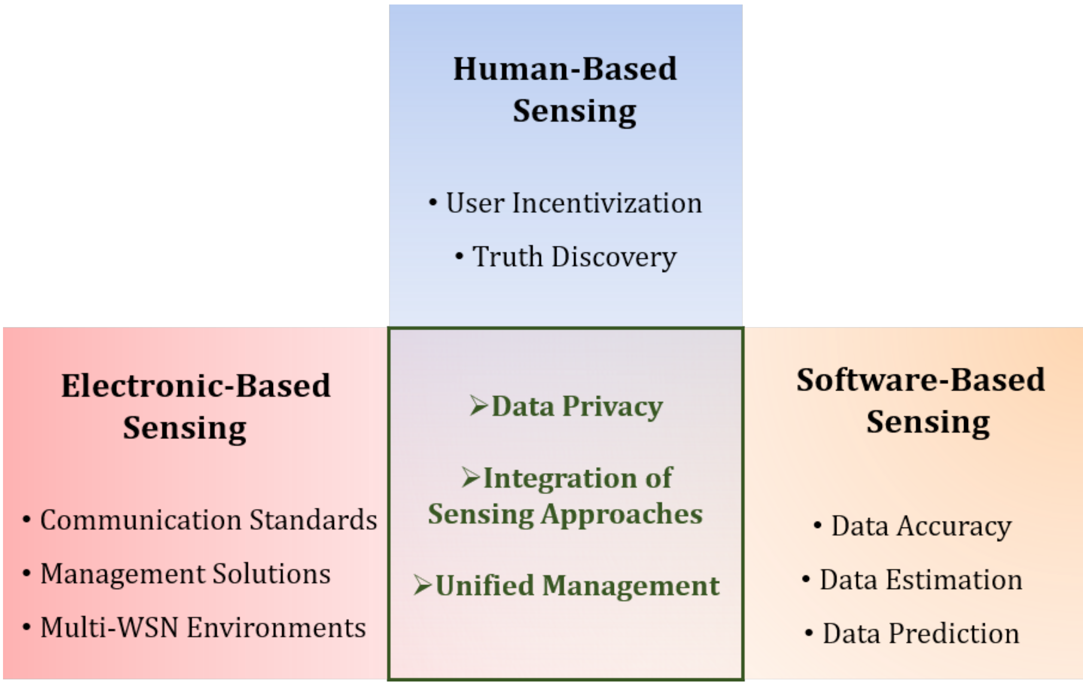



Figure 2.4: Main open issues per sensing approach and crossed open issues

Finally, the use of multiple technologies, either standardised or not, in multi-WSAN environments is one of the more critical challenges of current WSAN applications and systems, from a management point of view. Architectures and solutions that can integrate both multiple standards and multiple networks are essential for the full acceptance and effective deployment of WSAN [Raposo et al., 2017].

### 2.3.2 Software-Based Sensing

Virtual sensing may need to resort to algorithms and techniques for data estimation and/or data prediction. These pose the challenge of data accuracy. Cardell-Oliver and Sarkar [2016] tested a temperature virtual sensor framework in a real scenario to estimate values for days and months. With their solution, up to 95% of the sensor readings achieve an estimation error up to 0.5° Celcius.

According to the authors, their solution achieves significantly higher accuracy compared to state-of-the-art methods, which are suitable for estimating sensor data for a limited time-frame only, ranging from a few minutes to a few hours. Even though the results are useful for indirect sensing, there is room for improvement. Other virtual sensing solutions, e.g., Englert et al. [2015], have similar limitations. It is to be concluded that better approaches to estimation and prediction are needed.

## 2.3.3 Human-Based Sensing

### 2.3.3.1 User Incentivization

One of the significant challenges concerning social sensing and, more generally, crowdsensing techniques is how to provide incentives for user participation. The increasing popularity of Mobile Phones Social Sensors (MPSS) brings new problems to the OSN solution that can affect the users' motivation to join a community [Bachiller et al., 2016; Liu et al., 2018]. Among such problems, we have significant battery consumption, mobile network financial costs to the phone users, and substantial traffic load, in particular for applications that require fine-grained continuous sensing. Most of the current MPSS transmit mobile sensor data to servers through cellular networks [Vallati et al., 2016; Yang et al., 2015].

The approach of using OSNs as higher-level communication channels must consider conservative behaviour from the users regarding both battery and processing consumption, as well as communication costs. Yang et al. [2015] defend rewarding phone users to cover sensing and transmission costs to encourage them to participate in the social system. Finding effective ways for user incentivization is a critical issue for the success of social sensing, especially when it is MPSS-based.

### 2.3.3.2 Truth Discovery

The main challenge in social sensing applications is the determination of the correctness of the observations from unknown and potentially untrusted data sources [Huang and Wang, 2015]. This point is also shared by Marshall and Wang [2016] and Amin et al. [2015], to whom a critical challenge in Social Sensing paradigm is how to ascertain the credibility of claims and the reliability of sources without knowing them a priori. These challenges are labelled as "truth discovery" or "fact-finding".

## 2.3.4 Cross-domain Challenges

### 2.3.4.1 Data Privacy

Data privacy is another primary concern and challenge in a socially interactive Internet. Data privacy is a common issue in computing systems since they all deal with operating on data that are sensible. Yang et al. [2017b] concluded that all IoT devices could be vulnerable to certain types of attacks, just like any IT system is vulnerable at some point.

One of the approaches nominated to mitigate the risk of attacks is the development of security solutions with lightweight computing needs due to the limitations of the EBSs. The second approach is the development of open, standardised security policies for IoT products.

He et al. [2016] analysed the security challenges in IoT-enabled CPS, namely, critical infrastructures and Industry 4.0. Among the opportunities for future works, they identified investigations into the security architecture for IoT. Indeed, an appropriate security architecture would lead to an implementation of the security by Design, that is, before any installation of IoT-enabled CPS.

Even if the management of crowdsourcing platforms is a centralised one, this does not mean that they are willing to give up on their privacy. To Rui et al. [2016], there may be concerns regarding reliability and privacy-preservation when using crowdsensing platforms to deliver services. The studies by Abu Alsheikh et al. [2017]; Alsheikh et al. [2017] present solutions for the preservation of user privacy in a crowdsensing environment. Like related studies in the literature, the presented solutions rely on a reward-based system, which often has a monetary cost for the service provider.

#### 2.3.4.2 Integration of Sensing Approaches

The combination of sensing approaches - electronics, virtual, and human -, with the social capabilities that may be configured for each of them is one of the more prominent challenges for an effective, socially interactive Internet. Large-scale sensing tasks are more likely to get reliable results if sensed data is collected by many sensing approaches [Rui et al., 2016]. Thus, combining all available sensing approaches results in more efficient sensing, with enhanced contextualised data from a broader set of possibilities.

The proper combination of sensor inputs, while collecting and processing micro-sensors data, is still far from being ordinary[1]. In this regard, the special issue of a publication underlines the synergy among the collected data as still a challenging task in IoT [Yang et al., 2017a]. On the other hand, the platforms presented by [Lee et al., 2015; Psomakelis et al., 2016] are the first approach to the integration of the three sensing approaches.

However, the proposed integration is quite limited and cannot offer an appropriate response to IoT latency-sensitive applications [Fonseca et al., 2016; Taneja and Davy, 2017], besides being entirely dependent on Internet connectivity. In this regard, levering the Multi-access Edge Computing (MEC) paradigm [Abbas et al., 2018] in the integration of sensing approaches can be a new solution design to provide such an appropriate response. More generally, flexible integration approaches are needed, which, in addition to providing some integration, can adequately address other challenges, such as knowledge extraction, context awareness and cognitive systems.

[1] `http://cis.eecs.qmul.ac.uk/2015SummerSchool/LaurissaTokarchuk_CIS_SummerSchool2015.pdf`, accessed on 2017-05-20

#### 2.3.4.3 Unified Management

The heterogeneity in sensing and actuating approaches leads to the challenging research opportunity of finding a unified management solution for IoT devices in the broadest sense. Indeed, providing a management solution that deals with the IoT device layer from the perspective of the confluence of electronic-based, software-based, and human-based sensors and actuators is still an open issue. Nevertheless, proper management of IoT systems is crucial to their operation [ITU-T Study Group 20, 2012]. Many proposals addressing IoT management issues can be found in the literature but, to the best of our knowledge, they are focused on well-established physical-electronic IoT devices. While many such solutions are proprietary, some standards-based approaches can also be found, aiming at facilitating overall management of IoT systems.

Table 2.5 provides references to the key challenges that we identified in this section, for the three sensing approaches. While the first three columns from the left present state of the art, in the column *Limitations*, we show what is still a gap in the literature and thus the main open challenges. In the last column are listed some relevant works from which we can find the claimed limitations in each reference criteria.

## 2.4 Summary of the Chapter

Sensing is changing not only the Internet but also the way people are interacting with it and one another. What started as environmental sensing in a single, restricted-scope network has quickly grown into an interconnected web of "things". Currently, IoT is giving way to clouds of virtual sensors and to the Social Sensing paradigm, in which people and processes are both data producers and consumers. In this context, the current chapter provided a survey of sensing approaches in the literature, with emphasis on the 2015 - 2017 period.

Going beyond physical sensing, we addressed both social and virtual sensing approaches, and the respective actuation counterparts. The subsequent identification of research challenges points to critical open issues concerning the identified three sensing approaches: electronics, virtual, human-based. The challenges in each of the mentioned areas are stimulating. Nevertheless, unified management for the three built-in natures of the IoT devices, with the associated measurement results from practical implementation, is probably the most significant challenges, as they are crucial to an operational socially interactive Internet.

Table 2.5: Main open challenges identified in the literature

| **Reference Criteria** | **Research Directions** | **Potential Solutions** | **Limitations** | **Relevant Work** |
|---|---|---|---|---|
| **Data Accuracy** | • Data estimation<br>• Data prediction<br>• Truth discovery<br>• Fact-finding | • Analytical cognitive systems<br>• Integration of multiple sensing approaches | • Noisy Environments<br>• Management of data sources<br>• Users privacy | [Abu Alsheikh et al., 2017; Alsheikh et al., 2017; Cardell-Oliver and Sarkar, 2016; Englert et al., 2015; Huang and Wang, 2015; Liu et al., 2018; Marshall and Wang, 2016; Wang et al., 2015; Yao et al., 2016] |
| **Data Privacy** | Full-stack analysis | • Static analysis (stored data)<br>• Communication analysis (data circulating between communication systems)<br>• Design-driven security | • Energy and computational capabilities in IoT devices for running more efficient security algorithms<br>• General security policy and standards for IoT products<br>• Adaptation of advanced security in traditional systems to IoT systems | [He et al., 2016; Yang et al., 2017a] |

*Continued on next page*

Table 2.5 – *Continued from the previous page*

| Reference Criteria | Research Directions | Potential Solutions | Limitations | Relevant Work |
|---|---|---|---|---|
| **User Incentivization** | MPSS | • OSN<br>• Users rewarding system | • Traffic load and network financial costs<br>• Users privacy<br>• A reward-based system which often has a monetary cost for the service provider<br>• Data privacy issues | [Abu Alsheikh et al., 2017; Alsheikh et al., 2017; Bachiller et al., 2016; Liu et al., 2018; Vallati et al., 2016; Yang et al., 2015] |
| | Crowdsensing platforms | Reward-based systems | • Limited deployed case studies<br>• Quality of the data collected due to heavy traffic load and high-power consumptions<br>• Lack of universal Method for crowdsensing strategies<br>• A reward-based system which often has a monetary cost for the service provider | [Abu Alsheikh et al., 2017; Alsheikh et al., 2017; Liu et al., 2018] |

*Continued on next page*

Table 2.5 – *Continued from the previous page*

| Reference Criteria | Research Directions | Potential Solutions | Limitations | Relevant Work |
|---|---|---|---|---|
| **WSAN** | • Bandwidth allocation<br>• Connectivity while in mobility | • Cognitive radio<br>• Network virtualisation | • Limited deployed case studies<br>• Inefficient utilization of the spectrum | [Khan et al., 2017, 2016; Razafimandimby et al., 2016] |
| | Network management | Management standardisation | • Limited deployed case studies<br>• Multi-WSAN environments | [Raposo et al., 2017] |

# Chapter 3

# Unified Management of Traditional and "New Devices" in IoT

## Contents


**3.1 Generic Design . . . . . . . . . . . . . . . . . . . . . . 26**

3.1.1 IEEE 1451 Standards . . . . . . . . . . . . . . . . . 26

3.1.2 Extending the Basic Sensing Loop in IoT . . . . . . 28

3.1.3 Related Work . . . . . . . . . . . . . . . . . . . . . 32

3.1.4 Proposed Architecture . . . . . . . . . . . . . . . . . 35

**3.2 Off-the-Shelf Technologies Testbed . . . . . . . . . . 36**

**3.3 Standard-Based Frameworks Scenario . . . . . . . . 37**

**3.4 Enabler for 5G Product and Services . . . . . . . . 39**

**3.5 Summary of the Chapter . . . . . . . . . . . . . . . 43**

In section 2.2.1, we saw that sensing and actuating tasks involve a variety of sensors, comprising not only physical, electronic-based devices, but also virtual sensors, and even human-originated data collected from social networks. This heterogeneity in mechanisms for sensing and actuating leads to challenging questions in what concerns IoT solutions. For our thesis, we were interested in tackling the management of sensors and actuators beyond traditional electronic-based devices. We remind that DM is a concept to monitor and interact with both state and activities of the managed entities so that we can make sure that the applications running on a given device operate well [ITU-T Study Group 20, 2012, 2016].

Many proposals addressing IoT management issues can be found in the literature but, to the best of our knowledge, they are focused on well-established physical sensing [Sinche et al., 2018, 2019]. Well, one may want to change the algorithm of a virtual sensor on-the-fly or prevent a social sensor from feeding the IoT system for a while. While many such solutions are proprietary, some standards-based approaches can also be found, aiming at facilitating overall management of IoT systems. In this context, the objective of this chapter is to:

- present an outlook on the fundamental concepts in what we define as "extended IoT";
- propose a generic architecture for a management solution beyond traditional devices;
- present the prototype testbed and applied case studies that we set for the proposed generic architecture.

## 3.1 Generic Design

In our thesis, "extended-IoT" refers to the IoT in which operations are carried out by software and/or human agents, in addition to electronic-based, physical devices [Armando et al., 2018]. This section starts with an overview of the IEEE 1451 standards family [Lee, 2002; Kumar et al., 2015], which will be used as the basis for extending IoT to software and human sensors and their respective management.

### 3.1.1 IEEE 1451 Standards

Transducers (sensors, actuators, filters) are the primary electronic interfaces to the real world. They are entities that receive a signal as input and generate a modified signal as output [Botts, Michael E. and Robin, Alexandre and Greenwood, Jim and Wesloh, David, 2014]. In this regard, smart transducers are analog or digital sensing/actuating units combined with both a processing unit and a communication interface.

The family of smart transducer standards, under the designation of IEEE 1451, has been adopted by the industry, largely [Lee, 2002; Kumar et al., 2015]. It was created in 2007, with the objective of facilitating device and data interoperability in the realm of IoT and CPS. These standards define a set of common communication interfaces, network services, metadata concerning transducer connectivity to instruments, instrumentation systems, and control/field networks, in order to enable access, management, and control of networked transducers.

As shown in Figure 3.1, the IEEE 1451 standards defines two main blocks:

- Network Capable Application Processor (NCAP), widely known in the literature as "network gateway";
- Transducer Interface Module (TIM), widely known in the literature as "sensor node".

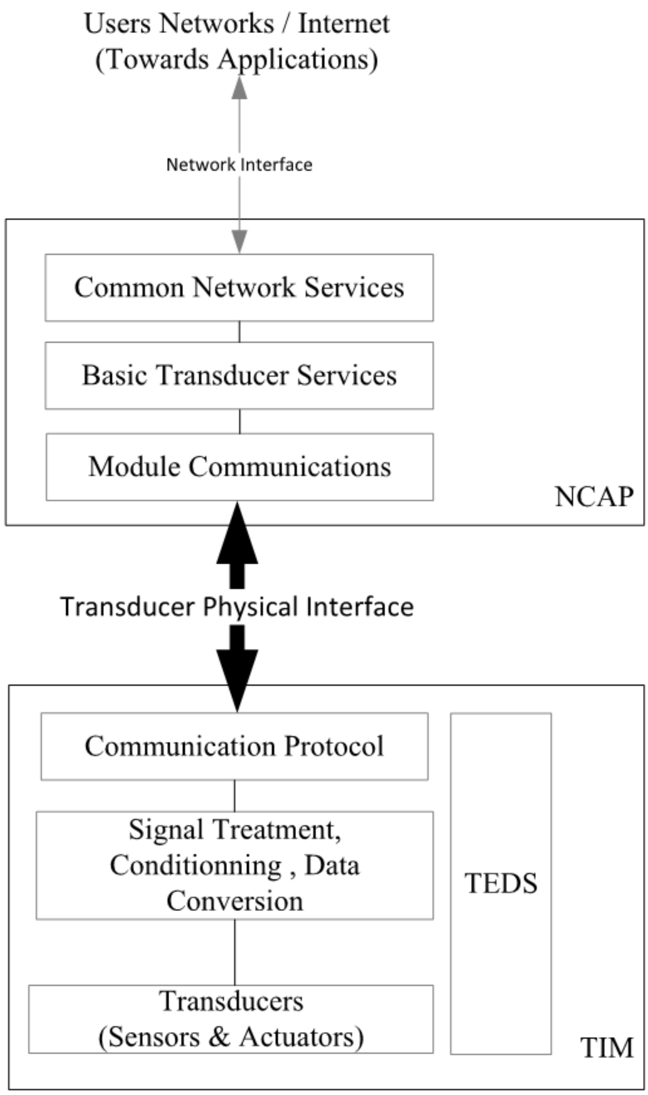



Figure 3.1: Standardized smart transducer scheme. *Adapted from [Lee, 2002]*

The NCAP block comprises communication modules to drive the Transducer Physical Interface and the essential functions required to control and manage transducers, communication protocols, and media-independent Transducer Electronic Data Sheets (TEDS) formats. It also comprises standard network services, for connectivity with user applications. The interface between the NCAP block and the TIM block includes both serial and wireless links, such as Serial Peripheral Interface (SPI), Inter-Integrated Circuit (I2C) and Recommended Standard (RS)-232/Universal Asynchronous Receiver Transmitter (UART) serial interfaces, defined in IEEE 1451.2, or wireless interfaces such as IEEE 802.11x, IEEE 802.15.1, IEEE 802.15.4, and IPv6 over Low -Power Wireless Personal Area Networks (6LowPAN), defined in IEEE 1451.5. 6LowPAN is intended to allow direct TIM access from the Internet.

The TIM comprises hard transducer end-units, metadata TEDS, signal processing units, and the necessary communication protocols to deal with the NCAP driver. TEDS contain manufacture-related information that allows both the self-identification and self-description of transducers to the system or network. For instance, in TEDS we may find the transducers' manufacturer identification number, serial number, measurement ranges, calibration data, and location information. TEDS usually reside in embedded memory, typically Electrically Erasable Programmable Read-Only Memory (EEPROM), within the transducer. The concept of virtual TEDS extends the benefits of the standardised TEDS to legacy sensors and applications where embedded memory is not available. That means virtual TEDS can exist as a separate file downloadable from the Internet [National Instruments, 2006], that is, without passing by an NCAP.

### 3.1.2 Extending the Basic Sensing Loop in IoT

For the sake of brevity, we will focus our analysis and discussion on sensing transducer units, not on actuators. Nevertheless, solutions for the management of sensing units can be easily adapted to cover actuating transducers as well, since the management protocols we will be considering have been designed so that the same programming functions serve for both actuation and sensing purposes. In a generic IoT sensing loop presented in Figure 1.1, neither end-users nor applications are interested in the sensors per se, instead of in the phenomena they monitor and for which the sensors provide an abstract view. In the following paragraphs, we revisit some definitions proposed by [Botts, Michael E. and Robin, Alexandre and Greenwood, Jim and Wesloh, David, 2014] that are relevant in the context of this thesis.

- Phenomenon: a physical or virtual state that can be observed and its properties measured. E.g., the temperature and other characteristics of an object is a typical real-world phenomenon. Regarding the phenomena in the information world, we have, for instance, the emotional state of individuals.

- Smart Sensor: an entity capable of observing a phenomenon to produce data which can then be interpreted, conducting information and knowledge. In the rest of the thesis, "smart sensors" will merely be referred to as "sensors".
- Process: an operation that takes one or more inputs, based on a set of parameters and a methodology to generate one or more outputs.

With the generic sensing loop in mind, we extend the basic sensing concept in order to include not only physical, electronic sensors, but also virtual and human sensors, taking as an example the measurement of the overall ambience of a house. To this end, let us take a scenario as an example, where we will set three metrics for the measurement of the phenomenon under consideration:

- room temperature;
- perceived comfort and
- the mood expressed in the (electronic) messages sent by the house dwellers.

To quantify each of these metrics, we will be using dedicated sensors, namely EBS, SBS, and HBS, to provide the inputs for the measurements of the temperature, comfort, and mood, respectively.

The traditional approach, illustrated in Figure 3.2, allows the gathering of temperature values for this sample IoT application. It uses a DHT11 sensor as TIM, connected to a Raspberry Pi micro computer in the role of NCAP. On the other hand, Figure 3.3 illustrates the collection of data that will provide information on perceived comfort. More specifically, a process is fed with measured values from a cloud of virtual devices, providing air temperature and relative humidity of the house. The comfort values are estimated upon a set of defined rules, according to both temperature and humidity [Sharanya and John, 2017].

The computer program that runs such a process is an SBS [Abdelwahab et al., 2016; Li et al., 2016; Sarkar et al., 2016]. Thus, the hardware platform that runs the process contains both the virtual TIM and the NCAP. Lastly, in Figure 3.4, we have a sensing loop approach that enables assessing the mood of the house dwellers. From this application's perspective, humans are sensors. Indeed, the text they post in the OSN will provide useful information to infer the overall ambience of the house, as previously mentioned. We use Twitter OSN for our stated scenario. With a web API, the content of tweets and the associated metadata can be retrieved, and subsequently submitted to analytic techniques that produce scores for the target inference. As in the case of TEDS for EBSs, tweets metadata carries relevant information for the IoT application.

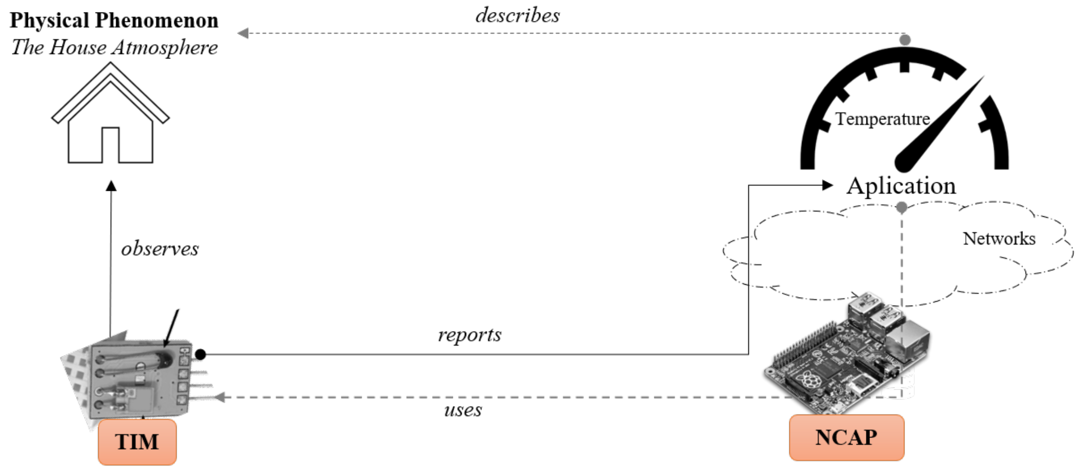



Figure 3.2: Electronic-based sensing loop

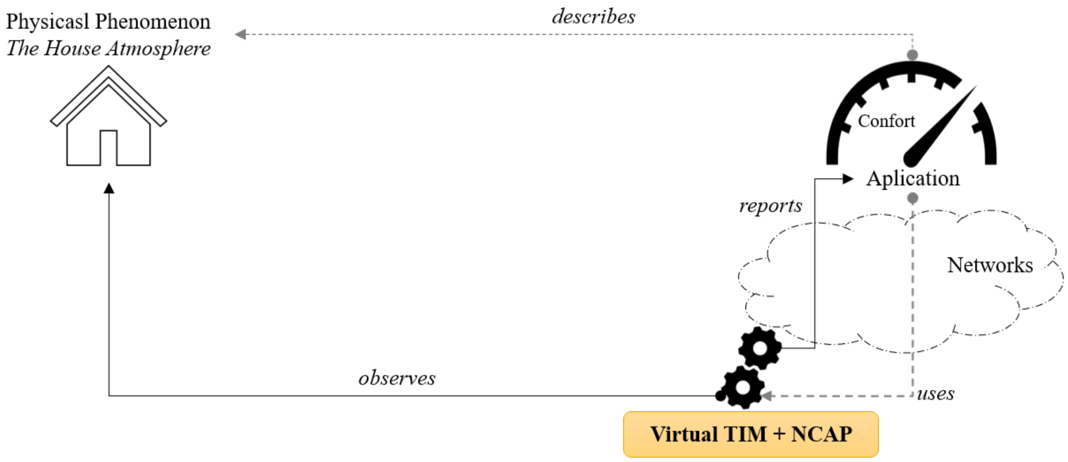



Figure 3.3: Software-based sensing loop

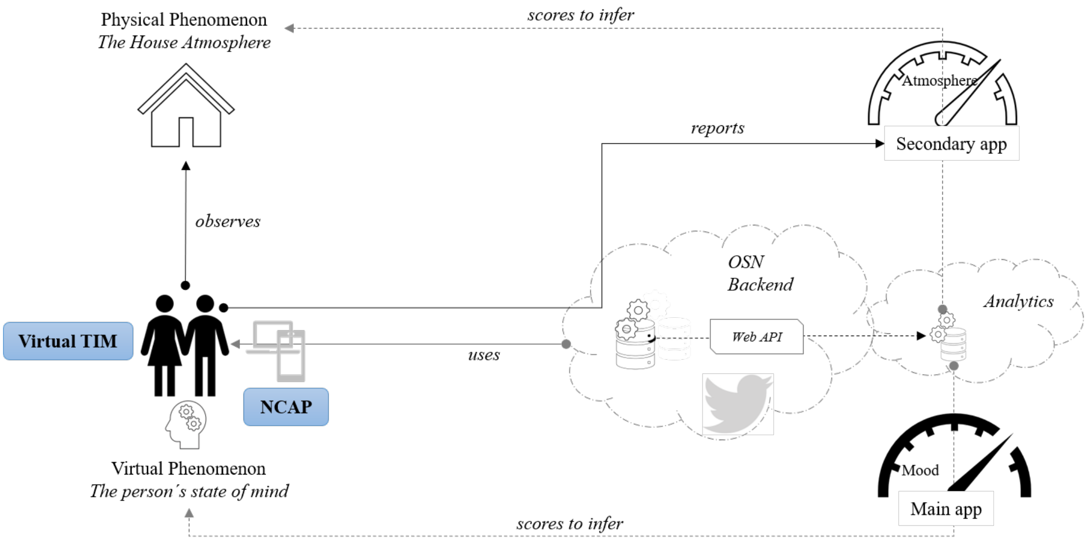



Figure 3.4: Human-based sensing loop

From the extended IoT perspective, sensors use a combination of means to estimate the observed underlying properties [Botts, Michael E. and Robin, Alexandre and Greenwood, Jim and Wesloh, David, 2014]. As we have seen in the provided example, this may involve physical, electronic means to enable the representation of things that are part of the target scenario, as well as software-based and Human-Based Sensor. We also note that in the presented example we did not consider the communication device (a personal computer, a mobile device) as a sensor, although in many cases it can have such role as well [Khan et al., 2013]. As a final remark, we highlight that the various types of sensors that make up the extended IoT are complementary. In that sense, they enrich the IoT data that will be used by applications to provide the agreed services levels.

In what concerns management, the ITU-T defined essential and specific management capabilities from the early years of the IoT [ITU-T Study Group 20, 2012]. Essential capabilities include device management, such as remote device activation and de-activation, diagnostics, firmware and software updating, device working status management, and traffic and congestion management. On the other hand, specific management capabilities are those tightly coupled with each application-specific requirement. Management capabilities are a cross-layer component associated with the four layers of the reference model. In each layer, IoT management covers fault, configuration, accounting, performance and security management capabilities.

Proper management of IoT systems is crucial to their operation, with emphasis when dealing with the extended IoT. The heterogeneity of the connected sensors adds much complexity in the management platforms. Hence, based on the content presented so far in this thesis, we proposed in Figure 3.5, an enhanced Device Layer from the ITU-T IoT reference model.

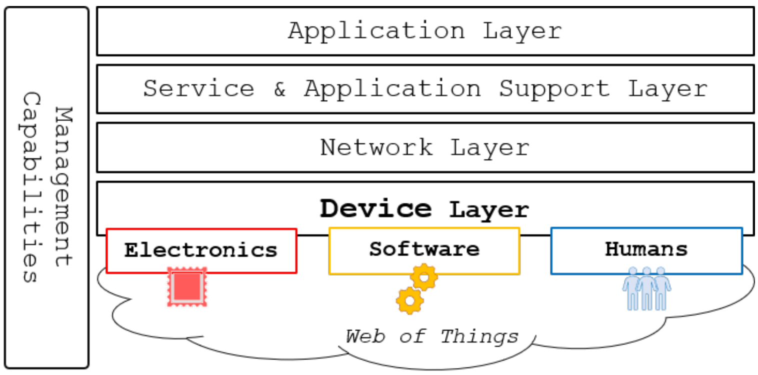



Figure 3.5: IoT management reference model. *Adapted from [ITU-T Study Group 20, 2012]*

Following, we will present the related work on management protocols and frameworks in IoT systems.

### 3.1.3 Related Work

#### 3.1.3.1 Management Platforms

The design of IoT management platforms must take into account a myriad of aspects that includes both the heterogeneity of resources and the networks protocols they require. The leading standardization institutions namely Internet Engineering Task Force (IETF), Open Mobile Alliance (OMA), OGC, European Telecommunications Standards Institute (ETSI), International Organization for Standardization (ISO) and the ITU-T work to ensure the integration and interoperation of the growing complexity in IoT systems.

While on the one edge of the management concept we have functionalities to serve a management center, we also have protocols to support the data exchange between the managed entities and the management center. Such a concept can be bundled by entities of codes in the form of services that are exposed more and more via RESTful APIs. APIs offer the flexibility of interactions, and they are tailored for the heterogeneity of components in the IoT system. In fact, it is expected that a management system handles seamlessly a large variety of technologies over which they operate. A middleware enables the abstraction of both technologies and protocols from different components of a communication architecture [Razzaque et al., 2016]. Therefore, it must be able to provide a connectivity layer so that the components can communicate regardless of the specific modelling languages and data models of each one.

Guth et al. [2018] made a comparison of open-source and proprietary IoT platforms. The authors analysed components of IoT systems, such as FIWARE [Fazio et al., 2015], Amazon Web Services (AWS) IoT[1], Azure IoT Microsoft[2], IBM WATSON IoT Platform[3] and Samsung's SmartThings[4]. Even they considered IoT devices, gateway, integration middleware and applications, the study does not include IoT management frameworks that are standardized, such as those available in the literature. The ISO/IEC 30141 [ISO, 2018] is one of these reference architectures defined in a general way. It does not provide specification about IoT management as the ITU-T does it in its Recommendation Y.4702 [ITU-T Study Group 20, 2016]. OneM2M is a global initiative targeting Machine-to-Machine (M2M) communication systems and IoT. Among the organizations that support the OneM2M initiative, we have the ETSI, the Telecommunications Industry Association (TIA), and OMA. The OneM2M proposes an IoT management architecture in [OneM2M, 2018].

[1] https://docs.aws.amazon.com/iot/latest/developerguide/iot-dg.pdf, accessed on 2019-02-29

[2] https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/iot/, accessed on 2019-02-29

[3] https://www.ibm.com/ru-ru/marketplace/internet-of-things-cloud, accessed on 2019-02-29

[4] https://smartthings.developer.samsung.com/docs/index.html, accessed on 2019-02-29

FIWARE is a European set of specifications to provide APIs that ease the development of "smart" applications in multiple vertical sectors. FIWARE is open, public and royalty-free. IoT agents in FIWARE are responsible for handling the management of the diversity of communication protocols from the IoT resources. While a model of an agent named "Ultralight" is the historical one for the tests in FIWARE, currently, there are agents for the main protocols used by the IoT resources. Among them, we have JavaScript Object Notation (JSON), LwM2M, that can run over different application protocols such as HyperText Transfer Protocol (HTTP), Message Queueing Telemetry Transport (MQTT), and Low Range Wide Area Network (LoRaWAN).

The Open Connectivity Foundation (OCF) developed an open-source framework for the interoperability of resources in IoT, which is described in their Core Specification V2.0-2018. They defined a component for IoT management and control with functional client-server interactions that include "resource discovery", notifications and management. The OCF framework is supported over communication technologies such as Bluetooth, ZigBee, WiFi, and mobile Long-Term Evolution (LTE). At higher layers, the framework operates over User Datagram Protocol (UDP)/Internet Protocol (IP) and Transmission Control Protocol (TCP)/IP, and support both IPv4 and IPv6. The adopted application protocol is Constrained Application Protocol (CoAP). Since 2017, OMA and OCF agreed to joint work on IoT device management.

Finally, the OGC also provides open-source standards that allow heterogeneous of devices in IoT to communicate. The OGC worked with IEEE to harmonize a set of their standards with the IEEE 1451 specifications [Kumar et al., 2015]. *SensorML* is among these standards from OGC that provide models and eXtensible Markup Language (XML) schema for describing the processes within sensor and observation processing systems. The OGC SensorThings API (STA)[5] provides an open, geospatial-enabled and unified way to interconnect the IoT devices, data, and applications over the Web. STA is designed to support CoAP, HTTP and MQTT messaging application layer protocols.

The frameworks and architectures in the literature emphasize the support of IoT resources, regardless of their technology or manufacturer. Thus, they can be managed as part of an IoT ecosystem. Some management frameworks and architectures can support different management protocols using gateways, as is the case of ITU-T gateway manager, the IoT Agent manager in FIWARE, and the management adapters in the OneM2M architecture.

#### 3.1.3.2 LwM2M Protocol

The platforms that we presented in section 3.1.3.1 support different management protocols, with a tendency to use the LwM2M protocol. There are numerous libraries, several products, and broad community support for LwM2M.

[5] `https://www.ogc.org/standards/sensorthings`, accessed on 2017-11-17

The IoT management solutions from top leaders currently in the IT market, such as AWS, IBM Watson, and Google Cloud platform are developed for LwM2M. For the future, the new versions of CoMI by the IETF[6] seem to be a more comprehensive proposal. The new versions of CoMI will also include LwM2M as part of its IoT resources management solutions.

With the payload encoded as plain text, JSON objects, or Tag-Length-Value, LwM2M intends to provide a single, secure protocol for controlling and managing both IoT devices and applications. That is, the devices implement and use the same agent function for both the actuation and sensing purposes and management of the IoT device itself. The typical management capabilities of the LwM2M protocol are shown in Table 3.1.

Table 3.1: Capabilities of LwM2M. *Adapted from [Zach Shelby, 2014]*

| **Managed Entities** | **Management Functions** | **Management Tasks** |
|---|---|---|
| Device | Bootstrapping | Key management |
| | | Service provisioning |
| | Configuration | Changes to settings |
| | | Changes to parameters |
| | Firmware Update | Update application and system software |
| | | Bug fixes |
| | Fault Management | Report errors |
| | | Query about status |
| Application | Configuration and Control | Configure settings |
| | | Send control commands |
| | Reporting | Notify changes in values |
| | | Notify alarms and events |

The LwM2M architecture, presented in Figure 3.6, includes two main logical components, namely the LwM2M server and the LwM2M client.

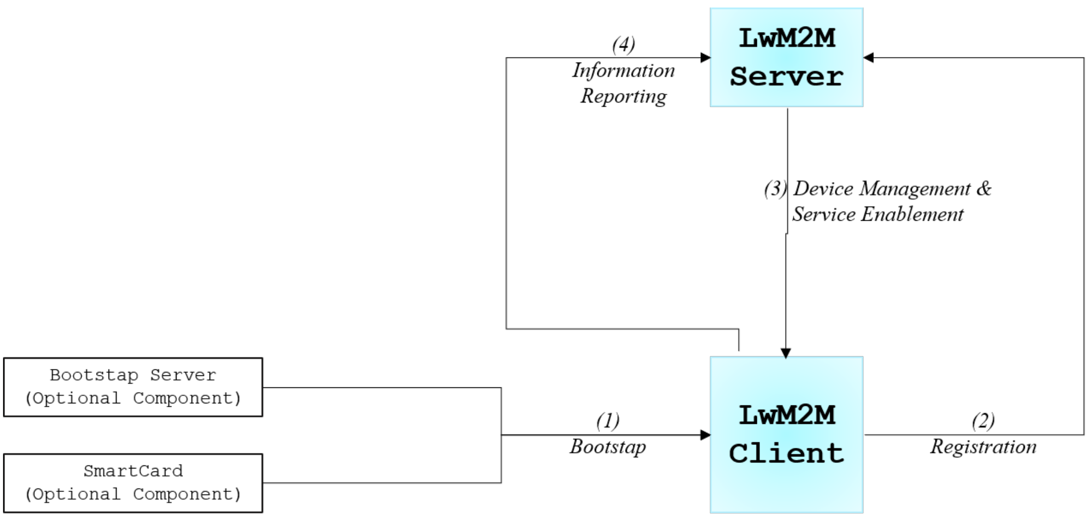



Figure 3.6: LwM2M architecture. *Adapted from [OMA, 2018]*

[6] https://core-wg.github.io/comi/, accessed on 2017-01-03

There are four interfaces between these two components: bootstrap, client registration, device management and service enablement, and information reporting. Typically, the LwM2M client resides in a network gateway, i.e., an NCAP, serving as an endpoint of the LwM2M protocol. The client communicates with a server to execute device management operations on its behalf. Contrary to the LwM2M client, the server is typically deployed within the M2M Network Service Provider.

### 3.1.4 Proposed Architecture

In Figure 3.7, we present the generic architecture over which our thesis project has been developed.

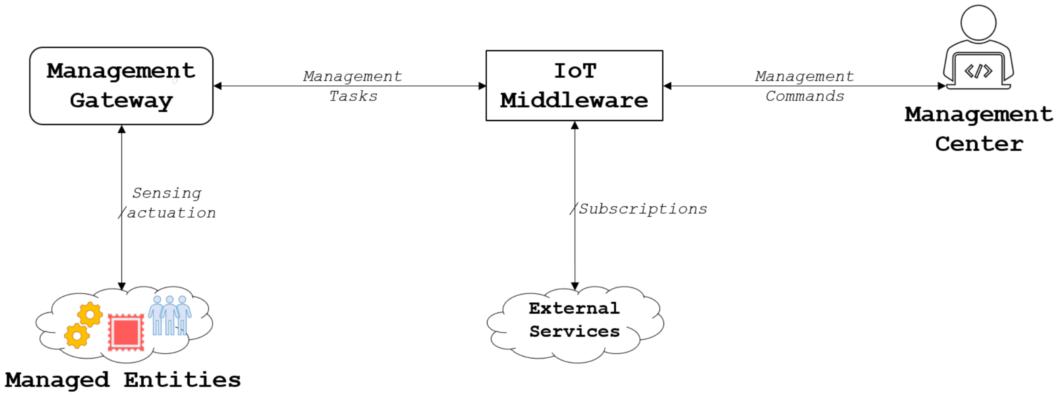



Figure 3.7: Generic architecture for unified management

The *IoT Middleware* is the central component of the generic architecture for a unified management of the "extended IoT". The middleware enables the abstraction of both technologies and protocols from different components of the architecture. Hence, it provides the flexibility and scalability of our architecture. The combination of the middleware flexibility with the robustness of well-established management standards is a fundamental approach to a unified solution for both data and device management.

The *Management Gateway* has the functionality of an NCAP defined in section 3.1.1. In IoT, they are typically mobile devices such as smartphones and stationary devices such as Raspberry Pi or Arduino-like devices. The IoT *Managed Entities* refer to the heterogeneity of elements with sensing/actuation capabilities as defined in section 3.1.2.

*External Services* include all the remote entities that can help the unified management system with additional capabilities. Such capabilities are to be typically related to parallel processing tasks.

Finally, the *Management Center* is the component of the architecture where Create, Read, Update, Delete, Notify (CRUDN) commands are sent towards the managed entities. It is typically to be interfaced by a dashboard.

The generic architecture from Figure 3.7 was exploited in applied case studies. We firstly mounted a Proof-of-Concept (PoC) testbed with off-the-shelf technologies in an attempt to test the deployment of a unified management scenario. Then, we introduced a second scenario by adopting more sophisticated tools. We developed a first case study to validate the models pertained to unified management. Finally, we leveraged the design in Figure 3.7 for the development of a second case study that was part of an integrated demonstration, providing services in the framework of future 5G networks.

## 3.2 Off-the-Shelf Technologies Testbed

In this section, we present a PoC testbed for the unified IoT device management approach. The solution is a first step towards the implementation of the generic architecture in a case study. The outcome of these implementations, which use off-the-shelf technology, shows that the proposed management approach is feasible and leads to a unified view of all types of sensing.

Figure 3.8 depicts the scenario for the test-bed. There are management databases both at server and client sides. The management server in Figure 3.8 resides in a laptop and maintains a global scope management database. The communication between the management server and management clients resorts to the LwM2M protocol.

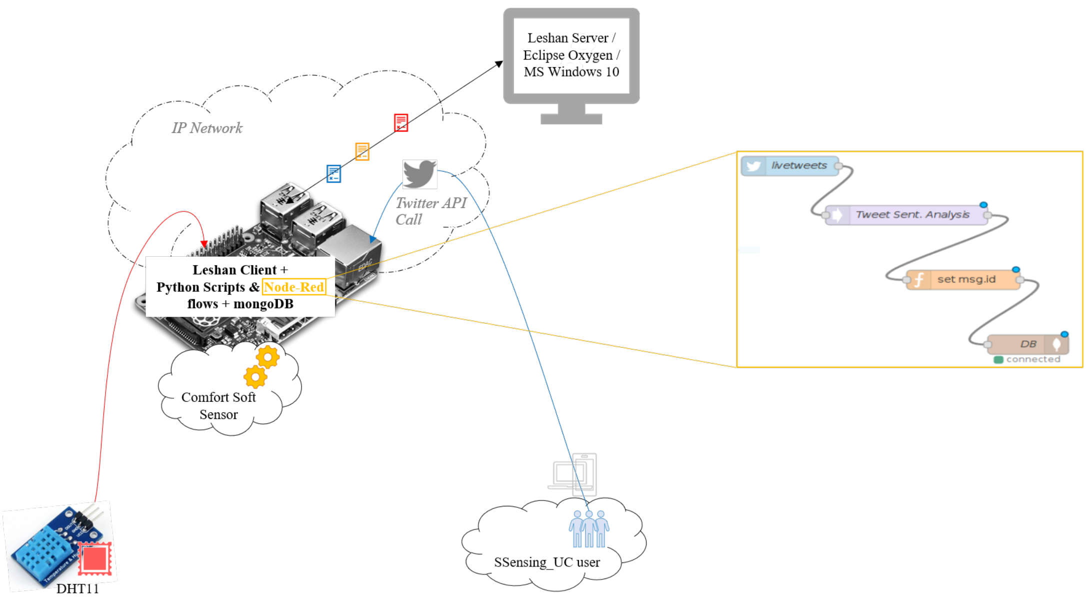



Figure 3.8: Setup for the off-the-shelf test-bed

The experiments described in this prototype testbed use a customized integration of *Leshan*[7], which is one implementation of the LwM2M protocol. Leshan provides Java libraries for developing LwM2M servers and clients. In each client, we loaded the instances of sensor objects from a local database.

[7]`https://www.eclipse.org/leshan/`, accessed on 2019-03-14

We recall that this study focuses on the management of sensing devices only. We remind that the management solution for sensing transducers can easily be adapted to cover actuating devices. Finally, our experiments included the following essential management capabilities: diagnostics (for fault management), working status and sensor reset (for configuration management), and notification of changes in values (for accounting management). To this end, we performed several read, write, and execute operations on each managed device.

The prototype's central hardware comprises a Raspberry Pi 3 running the Raspbian GNU/Linux 8.0 (Jessie) operating system[8]. This piece of hardware constitutes our management client, which runs the managed objects databases and the Leshan client logical component. The only EBS we used in the experiments was a DHT11 sensor[9]. As an SBS, we chose a comfort sensor. Finally, for the HBS, we created a Twitter account labelled "SSensing_UC".

In section 4.1, we will present the steps adopted to implement our PoC implementation, comprising the three types of sensing, namely physical sensors, virtual sensors, and human sensors. The basic functionalities resulted from this PoC testbed are presented in section 5.1.

The experiments in this prototype testbed aimed at providing management capabilities for the devices, but presented significant limitations, namely:

- Split data and device management
- No standardized data model and non-automatic device provisioning
- Non-persistent storage for both real-time and historical data

These limitations had consequences in the management architecture for scalability, flexibility and interoperability. With the stable release of a FIWARE-compliant LwM2M agent, i.e., a prototype of an IoT Agent accepting CoAP requests and redirecting to Next-Generation Service Interface (NGSI) consumer[10], in December 2018, it was possible to envisage a management solution that overcame the limitations described above.

## 3.3 Standard-Based Frameworks Scenario

The architecture in Figure 3.9 was deployed in the first case study that we developed "ISABELA". The ISABELA system will be extensively described in section 4.2. The setup in Figure 3.9 comprises six components, namely, Online Social Network, Mobile Host Device, IoT Middleware, NLP Engine, Stationary Host Device, and Management Center.

[8] `https://www.raspberrypi.org/products/raspberry-pi-3-model-b/`, accessed on 2018-01-30.
[9] `https://akizukidenshi.com/download/ds/aosong/DHT11.pdf`, accessed on 2017-11-18
[10] `https://github.com/telefonicaid/lightweightm2m-iotagent.git`, accessed on 2019-01-30

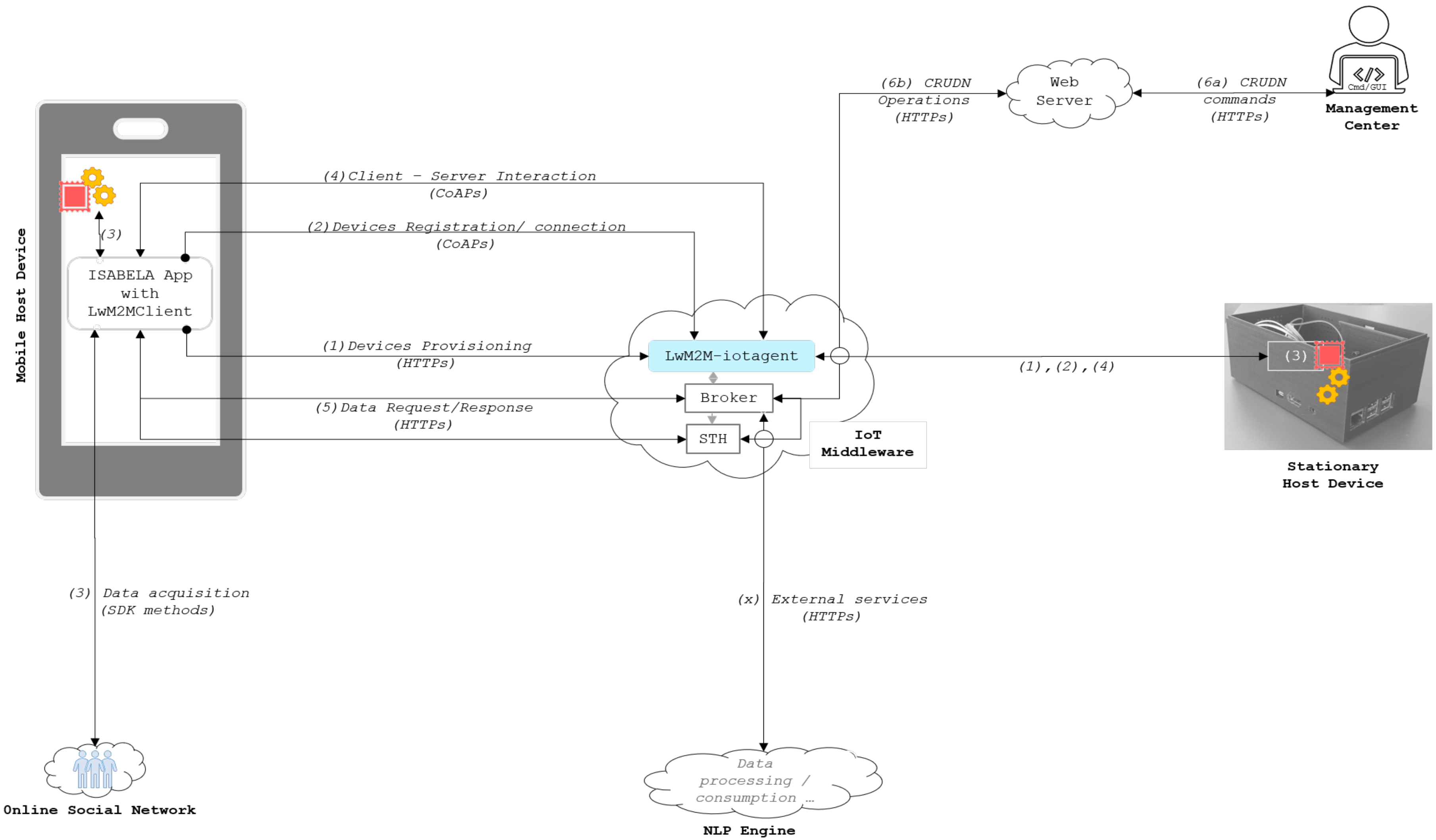



Figure 3.9: Setup with standard-based frameworks

We implemented a Leshan-based LwM2M client in the ISABELA mobile app. The IoT middleware is a FIWARE image configured for the ISABELA case study. Finally, the management center is a Postman desktop API client, from where performed the CRUDN operations on the monitored entities. The flow (1) in Figure 3.9 is an HTTP-based interaction with the administrative port of the IoT Agent. During the provisioning phase, the internal protocol in FIWARE will be in charge of creating the data structure of each monitored device in the Broker.

LwM2M clients enable the following actions:

- Connect to the IoT agent. This action corresponds to flow (2),
- Expose the clientsť resources to FIWARE and
- Enable the respective interactions (flow (4)).

The sensor values provided by the IoT resources correspond to the flow (3) in Figure 3.9. Flow (5) is used for triggering the NLP module for the text posted in OSN, and for sending the results back to the Broker.

Finally, the flow (6) is a set of requests from and towards the management center. Whenever flow (6) is triggered by the management center, the Broker will query the IoT Agent within the NGSI domain, so that the IoT Agent will communicate with the LwM2M clients over the NGSI-LwM2M bridge protocol.

The JSON-formatted payload provisioned to the LwM2M qgent is standardised in what is known as a device model. The structure of devices model will be covered in section 4.2.2.1. In section 4.2, we will present the steps adopted to implement our models in ISABELA case study. The resulted functionalities from this case study are presented in section 5.1.2.

We wanted to leverage the proposed generic architecture beyond the management of IoT devices. In the next section, we will introduce the second case study that we developed.

## 3.4 Enabler for 5G Product and Services

The next generation of cellular networks was designed to respond to the communication challenges for the year 2020 and beyond. Thus, 5G will be an important structure on which electronic communications should rely. We wanted to test the models that we proposed in our thesis on 5G networks.

For Portugal, this is one of the main projects for being a producer of 5G PPS. From a technological viewpoint, the project aimed to research, development, validation and integrated demonstration TRL 7/8 [Héder, 2017] of a set of products, capable of being part and providing services in the framework of future 5G networks.

The project also aimed at colligating and harmonising efforts of different telecommunication operators, intending to create innovative solutions for the global market, exploring bot Business-to-Business and to Customer models. The targeted products should cover all the functional domains in 5G networks, reflecting in their organization the following structure of domains: access, core and vertical sectors (differentiated by M2M communication and human communication-PPS4).

*Mobilizador 5G* was broken into five technical unities, where our solution was included in the unity for human communication. Thus, we proposed a set of products and services that would stress the supported communication network in what concerns its capabilities for broadband, latency and M2M communications. The reference comparison of the expected capabilities between 5G a Fourth Generation of Mobile Telecommunications (4G) is displayed in Figure 3.10.

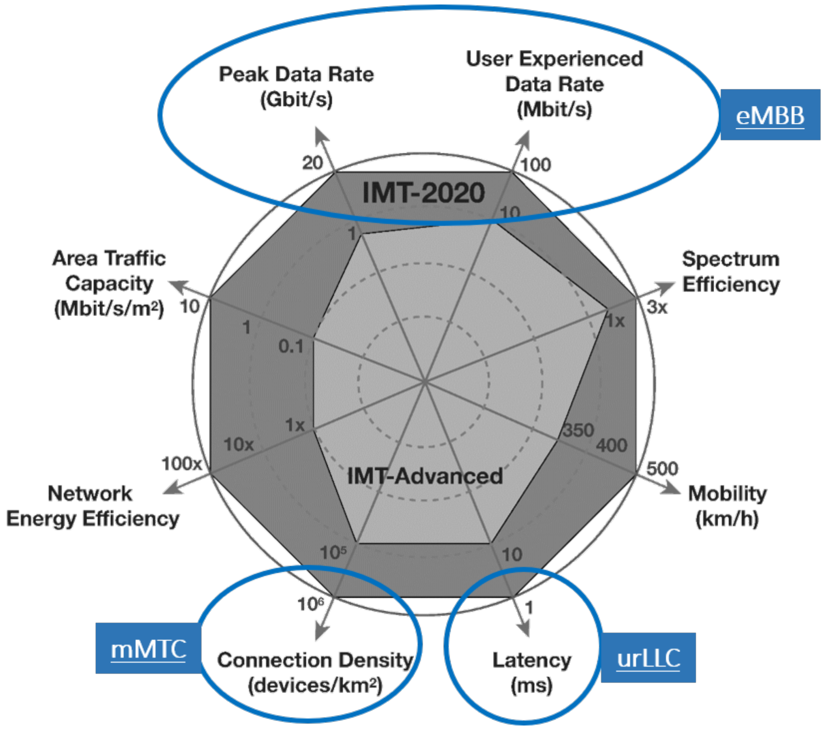



Figure 3.10: 4G capabilities versus 5G. *Adapted from [ITU-T Study Group 5, 2015]*

To this end, we designed, developed and tested a product named "5GOpenclasses", to support blended learning classes. The architecture for our solution is based on Figure 3.7, as displayed in Figure 3.11.

The Android application, which is part of the teaching facility, allows student users to view live streams of the class sessions, videos from a repository and to chat with their colleagues.

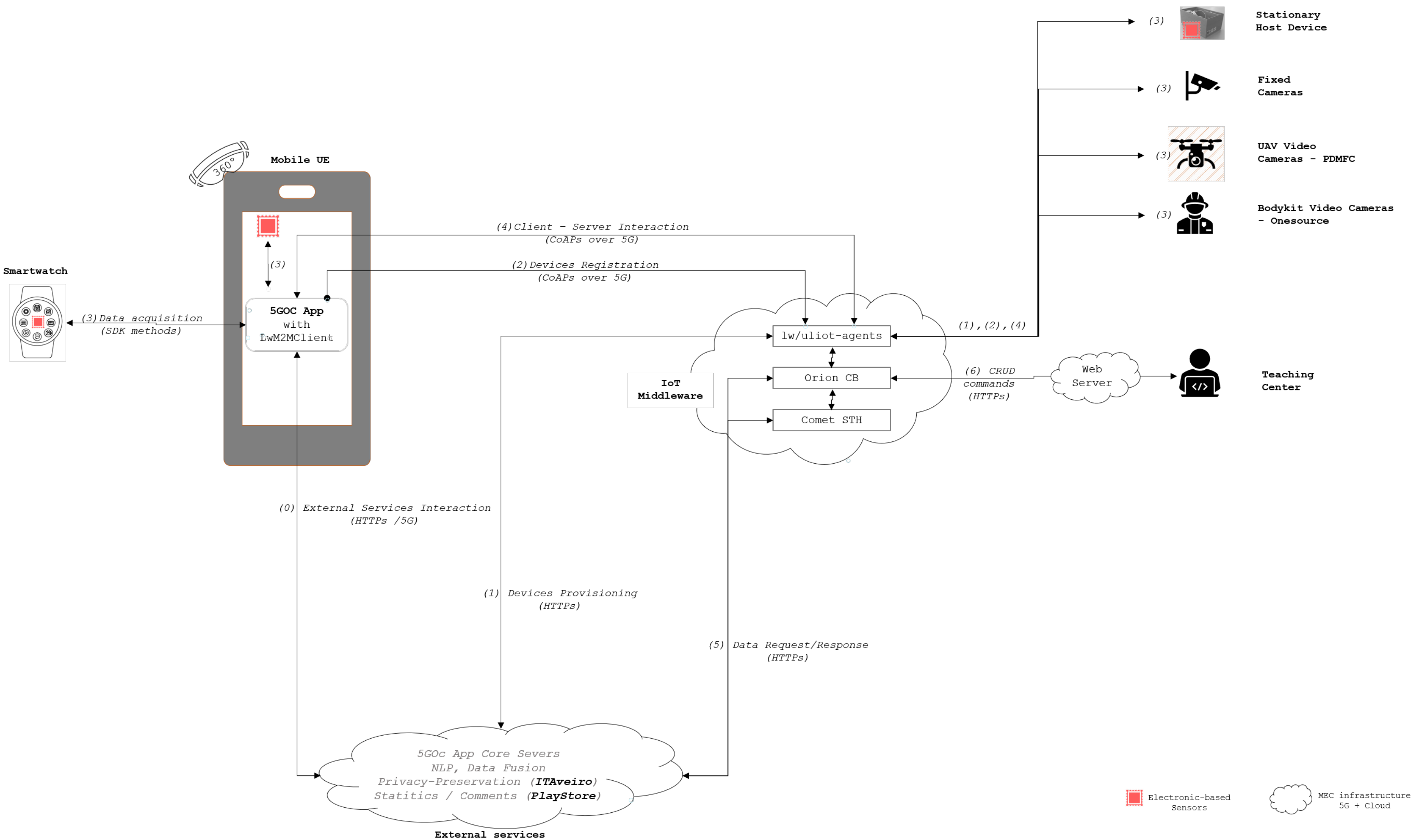



Figure 3.11: Setup for 5G Product and Services

Besides the mobile end-user application (app), we also developed a web-based app to serve the teachers to create, manage and close the class sessions. The live streams are managed as in a Peer-to-Peer (P2P) way between the cameras and the mobile apps logged into the ongoing class sessions. The app server Virtual Machine (VM) also hosts the storage for the class sessions materials.

As to experience the mMTC use-case in 5G, we collect two categories of sensor data during the class sessions, namely mobile and stationary. The former comprises gyroscope, accelerometer, proximity, sound, and light. The latter comprises light, noise, and temperature. While the mobile sensors are built-in the mobile devices, the stationary sensors will be provided by an IoT set box that we developed within the scope of ISABELA case study. It is essential to remark that mobile sensing is only performed on the User Equipments (UEs) that will be located inside the classroom premises. In the current version of the product, this detection is time-based, i.e., every 30 seconds, we toggle the value of a Boolean an attribute named "in_class" of the entity *Student*.

In an advanced version of the *5GOpenclasses*, the value for *in_class* is to be powered by the network indoor locating capability. The app server must receive the UE locations and its updates to decide toggling or not the *in_class* value for each student. This feature was thought to experience the uRLLC use-case in 5G. This feature requires further analysis to be fully implemented due to the number of interfaces to be considered from all the components, i.e., the mobile app, both the Third Generation Partnership Project (3GPP) Radio Access Network (RAN)s and the core networks and the application backend.

Overall, the collected data from sensors will serve analytical services to feed the teacher dashboard. For instance, the teacher will be able to monitor the influence of sensing values in the student's performances as tested in ISABELA use case [Fernandes et al., 2019, 2020; Sinche et al., 2020].

The core functions of the application are hosted in a core App server, where the provision of session control and media server, are handled. The app server is based on FIWARE. The optimised video streaming will rely on a eMBB slice configured by other partners of the project.

Concerning the Quality of Experience (QoE) process to the proposed case study, the data collected from the user's feedback can be summarized as early as possible in the network, before sending the associated results to a processing unit for the selected QoE metrics. The evaluation of the QoE was carried on with subjective metrics, namely the collected comments and rating from the users via a questionnaire built in the mobile app. The comments can be processed by external NLP services such the one working for ISABELA case study, presented in section 4.2.1.4.

As part of an integrated project, the authentication process are powered by the Privacy-Preserving User Authentication Protocol (PPUAP) solution from ITAveiro[11] [Sucasas et al., 2016]. Likewise, the array of live video streams from outdoors, namely Unmanned Aerial Vehicles (UAV) and a live streaming camera attached to firemen suit (Bodykit) are provided by other partners from the project, namely PDM-FC[12]and Onesource[13].

## 3.5 Summary of the Chapter

This chapter is the core of our thesis because this is where we set our proposal, according to one of the gaps that we have identified in the literature. We proposed a generic architecture, then we presented the designs of the prototyped test-bed and applied case studies to validate it.

The first served as a PoC for an evaluation of the proposed architecture by using simple, off-the-shelf components.

In the second setup, we wanted to explore more standardised tools for the frontend, the backend and the external services. Therefore, we developed an applied case study named "ISABELA" integrating the components of the unified management approach. We used FIWARE middleware for combined data and device management. We also explored the replacement of the Node-red flow from the PoC by a more advanced and open/tuneable solution to perform external services.

The last setup was also developed for a case study scenario named "5GOpenclasses", where we wanted to leverage the proposed generic architecture beyond the management of IoT devices. Such adaptability to different case studies has been possible thanks to the flexibility of the middleware, which is the central component of our generic architecture.

[11] https://www.it.pt/ITSites/Index/3
[12] https://www.pdmfc.com/
[13] https://onesource.pt/

# Chapter 4

# Engineering of the Testbed and Case Studies

## Contents


**4.1 Off-the-Shelf Testbed . . . . . . . . . . . . . . . . . . . 46**

4.1.1 Protocols and Technologies . . . . . . . . . . . . . . 46

4.1.2 Development and Configurations . . . . . . . . . . . 47

**4.2 ISABELA . . . . . . . . . . . . . . . . . . . . . . . . . 53**

4.2.1 Ecosystem . . . . . . . . . . . . . . . . . . . . . . . 53

4.2.2 Device Models, Configurations and Code adaptations 63

4.2.3 Ultralight versus LwM2M . . . . . . . . . . . . . . 69

**4.3 5GOpenclasses . . . . . . . . . . . . . . . . . . . . . . 74**

4.3.1 Backend . . . . . . . . . . . . . . . . . . . . . . . . 75

4.3.2 Frontend . . . . . . . . . . . . . . . . . . . . . . . . 77

**4.4 Summary of the Chapter . . . . . . . . . . . . . . . . 78**

The methodology to achieve the research objectives will be both exploratory and experimental, leveraging the existing frameworks in the state-of-the-art. In this chapter, we will present the different technical aspects that we have exploited and the engineering choices we made to validate the generic architecture proposed in section 3.1.

We started working on the first version of a unified management solution, leveraging off-the-shelf technologies and frameworks. The second setup to validate the generic architecture has been implemented in a real case-study scenario. The generic architecture also served as a structure for a solution that we proposed in a project involving partners from other research centers and the industry.

## 4.1 Off-the-Shelf Testbed

The source code of the experiments that we caried for this testbed are available online at `https://goo.gl/YXyA66`. The repository online includes the customised Leshan repository for both client and server projects. The repository online also comprises the python scripts for reading the temperature and the comfort values, and to store the objects in a local database. Finally, the repository online comprises the *Node-Red* flow code for running the pipeline that retrieves the tweets objects and performing English-language text SA. Both MongoDB and Node-red come by default in the Raspberry Pi 3 Operating System (OS). Following, we describe the set of tools that we used to implement the unified management testbed.

### 4.1.1 Protocols and Technologies

Eclipse and Apache Maven

We used the Eclipse oxygen[1] Integrated Development Environment (IDE) to integrate the Leshan project. Since the Leshan repository comes as a Maven project [2], we installed Apache Maven in the Microsoft Windows 10 OS.

Fuzzy Logic

This mathematical tool was used to implement the comfort sensor, taking as input the air temperature and relative humidity readings. A fuzzy system comprises four essential elements, namely fuzzifier, inference engine, fuzzy rule base, and defuzzifier [Fei et al., 2017]. The fuzzifier converts the inputs into what is called "fuzzy variables" using membership functions. Each variable represents a degree of membership to a specific fuzzy set. Fuzzy variables provide a mapping of objects to a constant membership value normalized in the range of 0 to 1. Each fuzzy set is represented by a linguistic term, such as high, low, medium, small and large.

[1] `https://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/oxygen1a`, accessed on 2018-01-30.

[2] `https://maven.apache.org/index.html`, accessed on 2018-01-30.

The inference engine is a collection of *if ... then* rules, by which the fuzzy input is mapped to a linguistic output variable according to the fuzzy rule base. The output variable has to be converted into an output score by a so-called "defuzzification process", such as centroid, averaging, root sum squared, or mean of the maximum method. The output score used in our experiments ranged from 0-100%, which corresponds to low and high comfort, respectively. The higher score is obtained when the temperature is between 20 - 25° Celcius, and relative humidity is around 50 %.

Node-Red
Node-red is a flow-based programming toolkit for IoT. It uses the JavaScript management system and installs library packages called "nodes" that run the required functions[3]. A set of nodes working for one global task is called "a flow". In our case, we configured one flow to fetch tweets, assign a sentiment score to its payload, and store it in the MongoDB database. Our testbed for unified management "v1" was developed with the Node-red v0.17.5 release.

MongoDB
MongoDB is an open-source NoSQL database that provides high performance, high availability, and automatic scaling. A record in MongoDB is a data structure composed of field and value pairs called "a document". Documents are similar to JSON objects. MongoDB provides a pluggable storage engine API that allows third parties to develop storage engines for MongoDB. It stores documents, in what are called "collections"[4] and finally, databases hold collections of documents[5]. Our work was developed with the MongoDB v2.4.10 release.

### 4.1.2 Development and Configurations

#### 4.1.2.1 Management of EBSs

Step 1 - Definition of the object resources
Tables 4.1 and 4.2 present the generic structures of the documents that hold the resources exploited in Leshan client. Of course, the list of attributes defined in these documents can be extended, depending on the use-case. The collection we created for electronic-based devices is labelled "mdb_ebs", which includes two documents corresponding to the content of Tables 4.1 and 4.2. The first document is labelled "static", which contains all the attributes/values pairs we want to be updated manually.

The second document is labelled "dynamic", which contains all the attributes/-values pairs we want to be updated automatically by the managed entity. The assignment of a static ID to each document enables the Leshan client to interact with the right resources of the managed entities.

[3] `https://nodered.org/`, accessed on 2018-01-30.
[4] The equivalent of tables in traditional relational databases
[5] `https://docs.mongodb.com/manual/introduction/`, accessed on 2018-01-30.

Table 4.1: Structure of static documents in mdb_ebs

| Attributes | Data type | Description |
|---|---|---|
| _id | double | Unique number to identify the object |
| application | string | Brief description of the sensor |
| configuration | - | Five subfields |
| 1.manufacturerID | double | Basic TEDS requirement |
| 2.model | string | Basic TEDS requirement device model |
| 3.minrange | double | Minimum range values for the target metric |
| 4.maxrange | string | Maximum range values for the target metric |
| 5.units | string | Unit of the target metric |
| type | string | Type of device in "extended IoT" perspective |
| info | string | Online Uniform Resource Locator (URL) for more information about the sensor |

Table 4.2: Structure of dynamic documents in mdb_ebs

| Attributes | Data type | Description |
|---|---|---|
| _id | double | Unique number to identify the object |
| location | object | Latitude, longitude, altitude of the sensing device |
| value | float | Value of the measured temperature |
| timestamp | date | Time of the fetched value |

Step 2 - Creation of the *temperature.py* file

A python script was used to update the dynamic values into the "mdb_ebs". The script includes the acquisition of the target metric, which in this example is the temperature value. In addition to the dynamic values, the script can also be used to update the static values. Using the MongoDB save operation in the script enables to insert a new document in the "mdb_ebs" and then keep on updating the same document because the _id is fixed. This updating approach is a deliberate choice.

We wanted to fix the objectst̕ _id so that the Leshan client will be able to fetch the right resources it is programmed for, within the "mdb_ebs". On the other hand, this approach allows saving memory space on the host machine, because we only keep the last configurations and activities of the devices. Depending on the goals of the application, we could use another approach and save the historic data. To this end, the _id value should not be a unique value, and we should replace the MongoDB *save* method with an *insert* one.

Step 3 - Download and installation of tools

After installing both the IDE and Maven, we cloned the Leshan project from the official project[6] to the Eclipse workspace, and then prepared the workspace for Maven. The *Leshan-client-demo* project available in the Leshan repository simulates, by generating random sensor values, a temperature sensor, and implements all the required fields of the *3303.XML* model.

[6] `https://github.com/eclipse/leshan`, accessed on 2018-09-08

The numbers reserved for the IoT resources in LwM2M are standardized by OMA in what is called "IP for Smart Objects (IPSO)" registry[7]. The code from the Leshan project is structured into four classes, namely *LeshanClientDemo*, *MyLocation* class, *MyDevice* class and *Temperature* class. To implement this part of our experiments, we created the following classes and .xml model, based on the Leshan repository samples: *TemperatureDaemon*, *TemperatureSensor*, adapted *LeshanClientDemo*, *LeshanServerDemo* and a new 26241.xml

We did not modify the structure of *MyDevice* and *MyLocation* classes for the whole experiment because we ran the Leshan client in the same hardware host, i.e., a Raspberry Pi computer.

Step 4 - *TemperatureDaemon* class

This class was developed to trigger the *temperature.py* script. Besides, it is responsible for fetching the values in "mdb_ebs" according to the attributes of the EBS.

Step 5 - Creation of *TemperatureSensor* class

To represent the temperature LwM2M object, we created the *TemperatureSensor* class. The class extends the *BaseInstanceEnabler* class from *leshan.client.resource* and implements the Observer to receive the updates of sensor data when a new value is collected. From the *BaseInstanceEnabler* class, the *TemperatureSensor* overrides two methods namely, *read()* and *execute()*.

The read method handles the read requests made by the Leshan server, returning the value of corresponding resource identification. The execute method receives commands from the Leshan server and executes operations. From the Observer interface, the class overrides the update method. This method is called when the *TemperatureDaemon* class collects new data. Thus, the method updates all the defined sensor variables in the object and the .xml model.

Step 6 - Further adjustments

We created a new *.xml* LwM2M device model with the IPSO number 26241, that includes the resources set in Tables 4.1 and 4.2. Both *LeshanClientDemo* and *LeshanServertDemo* classes have been modified to upload the values of the resources in 26241.xml exposed by the devices. In the server *project/models* folder, we first created the new .xml model allocating it a number within the IPSO privately reused scope, i.e., "26241 - 32768". After, we added a 26241.xml label to the list of *LeshanServertDemo* Class / modelPaths. Then, a replica of the XML has been loaded into the client project/models folder.

Finally, we adapted the content *LeshanClientDemo* class to the new setup of the Leshan client project.

[7] `http://www.openmobilealliance.org/wp/OMNA/LwM2M/LwM2MRegistry.html#resources`, accessed on 2017-03-03

Step 7 - Running the programs

We started by exporting the *leshan-client-demo* project as a runnable *.jar extension* file and saved it in the same folder where the python script was hosted. After, we wired the DHT11 sensor to the Raspberry Pi with the data pin connected to the General-Purpose Input/Output (GPIO) #25.

In Eclipse, we launched the *leshan-server-demo* project retaining the IP address of the host machine, in this case, a laptop, where it was running. Typically, the IP address is displayed at the bottom of the console space in Eclipse. Then, we went back to the Raspberry Pi and opened the shell to the folder where both the python script and the *leshan-client-demo.jar* have been stored. Finally, we ran the *leshan-client-demo.jar*, providing the server IP address and the coap port number in the option fields, e.g., *> $ java -jar leshan-client-demo.jar -u 10.3.3.232:5683*.

#### 4.1.2.2 Management of SBSs

Since we propose a unified management approach, the steps to implement the management capabilities for SBSs are equivalent to those previously described in the EBSs. For this reason, the following paragraphs only describe the features that are different from one to the other. Tables 4.3 and 4.4, structure the documents for "mdb_sbs".

Table 4.3: Structure of static documents in mdb_sbs

| **Attributes** | **Data type** | **Description** |
|---|---|---|
| _id | double | Unique number to identify the object |
| application | string | Brief description of the sensor |
| _id | - | Four subfields |
| 1.model | string | To fill with the name of the library performing the virtual sensing |
| 2.minrange | double | Minimum range values for the target metric |
| 3.maxrange | double | Maximum range values for the target metric |
| 4.units | string | Unit of the target metric |
| type | string | Type of device in "extended IoT" perspective |
| info | string | Online URL for more information about the sensor |

Table 4.4: Structure of dynamic documents in mdb_sbs

| **Attributes** | **Data type** | **Description** |
|---|---|---|
| _id | double | Unique number to identify the object |
| inputs | object | Set of input values from third-party sensors |
| location | object | Latitude, longitude, altitude of the sensing device |
| value | float | Value of the measured comfort |
| timestamp | date | Time of the fetched value |

Due to the specificity of software-based entities, as compared to electronic-based ones, the documents in "mdb_sbs" will have slight differences compared to those in "mdb_ebs". Specifically, we discarded the *manufacturerID* in static documents because we think it is not relevant to the SBSs. In dynamic documents, we added the attribute labelled "input" because, in the case of SBSs, the observed phenomena are values from third-party sensors.

After defining the object resources, we created the adapted Java classes and .xml model. We defined a new model 26242.xml with the resources of the objects following the set in Tables 4.3 and 4.4. Both *LeshanClientDemo* and *LeshanServertDemo* classes were updated accordingly, comprising a parameter for the EBS and SBS. In the server *project/models*, we first created the new .xml model. Then, we added a *26242.xml* label to the list of *LeshanServertDemo* Class / *modelPaths*. After that, a replica of the LwM2M device model (.xml file) was uploaded into the client *project/models*. Finally, we adapted the content *LeshanClientDemo* class to the current setup of the Leshan client project, comprising the parameters for both EBSs and SBSs.

As the server was already running, we went back to the Raspberry Pi shell and ran the new *leshan-client-demo.jar*.

#### 4.1.2.3 Management of HBSs

Likewise, the steps to implement the management capabilities for HBS are equivalent to those previously described for SBSs. For this reason, the following paragraphs only describe the features to be adapted accordingly.

<u>Creation of Node-Red flow</u>
The scheme on the upside right of Figure 3.8 is mainly composed of a Twitter API configured to fetch the live tweets posted by the "SSensing_UC" user. At the same spot in Figure 3.8, we have Node-red node called "Sentiment". We renamed it to "Tweet Sent. Analysis" for documentation purposes. This node is used to analyse the payload of the tweet and add a SA score based on an *AFINN-111* lexicon [Nielsen, 2011]. According to the "Tweet Sent. Analysis" node used in our experiments, a positive score will mean positive emotional content, whereas a negative score will mean harmful emotional content.

The SA scores proposed by the "Tweet Sent. Analysis" node are typically ranged from -5 to +5. Such score is the result of the sum of the polarity the lexicon allocates to each word of the text. We also configured a function node to set a unique _id to the document via the node labelled "Set msg.Id", because we wanted to use the *save* operation in mongoDB database.

We remind that the data we retrieve by calling the Twitter API includes only the public information made available by the users. The SA score is not public, and we compute it locally by configuring the "mongoDB" node, renamed "DB" in Figure 3.8, with a *save* operation. Thus, we do not store the historic data either. The last node in Figure 3.8 is a "mongoDB out", renamed as explained above, which was configured to store both the last Tweets and the affected sentiment.

We defined two documents and a collection we created in the database labelled "mdb_hbs". The static document for "mdb_hbs" has the same structure as the static document in "mdb_sbs". Of course, the values are adapted to each sensing approach, as we will see in the results. Such a choice was made because the dynamic document already contains the main attributes to be managed. Indeed, it contains all the metadata from the last tweet and the associated SA details. Table 4.5 presents the attributes that we selected, within the raw records that the Node-red flow provides to the "mdb_hbs" collection.

Table 4.5: Structure of dynamic documents in mdb_hbs

| **Attributes** | **Data type** | **Description** | **Node-red flow Resource** |
|---|---|---|---|
| _id | double | Unique number to identify the document | _id |
| inputs | object | Input values of the sensors | tweet.text |
| location | object | Latitude, longitude, altitude of the sensing device | location..place |
| value | object | Value of the measured polarity | sentiment.score |
| timestamp | date | Time of the fetched value | tweet.created_at |

Finally, we created the following classes and .xml model: *PolarityDaemon*, *PolaritySensor*, and new *26243.xml*. Contrary to other equivalent classes in both "mdb_ebs" and "mdb_sbs", here *PolarityDaemon* class ran a python script. Indeed, the Node-red flow already updates the "mdb_hbs" for every tweet sent by the "SSensing_UC". However, the optional script enables to load the static document in "mdb_ebs". The new model *26243.xml* was defined with the resources of the objects following the set in Tables 4.3 and 4.5.

Both *LeshanClientDemo* and *LeshanServertDemo* classes were updated to upload *26243.xml* values provided by the HBS. It comprises the parameters for the previous electronic and software-based "devices" within the scope of the extended IoT detailed in section 3.1.2. In the server *project/models* file, we first created the new *.xml* model; then we added a *26243.xml* label to the list of *LeshanServertDemo* Class/*modelPaths*. After that, a replica of the device was loaded into the client project/models. Finally, we adapted the content of the *LeshanClientDemo* class to the current setup of the Leshan client.

The obtained results of this PoC testbed will be presented in section 5.1.1. Following, we will describe the first case study called "ISABELA", which serves as an environment to validate the second version of the unified management.

# 4.2 ISABELA

## 4.2.1 Ecosystem

### 4.2.1.1 Technologies and Frameworks

The aim of ISABELA study case was first to infer the impact of the students' lifestyle in their academic performance, and then provide them with recommendations for a set of good practices towards the achievement of better outcomes. ISABELA is also a specialization of the IoT paradigm in which humans and their interactions can simultaneously be viewed as data sources and sinks [Wang et al., 2019]. The ISABELA system bridges the gap between IoT and the beneficiaries of technologies in a network of connected embedded devices. To this extent, people become a part of a device cloud surrounded by a highly dynamic pool of resources used to fulfil their needs.

ISABELA leverages a set of technologies that support its different components shown in Figure 4.1. First, an Android mobile application that serves as an interface between the end-users (students) and the IoT resource pool, including a chatbot. The application also enables to collect selected sensing data from mobile handheld devices. Likewise, the application serves as an entry point to retrieve objects provided by Social Sensors via their "timeline".

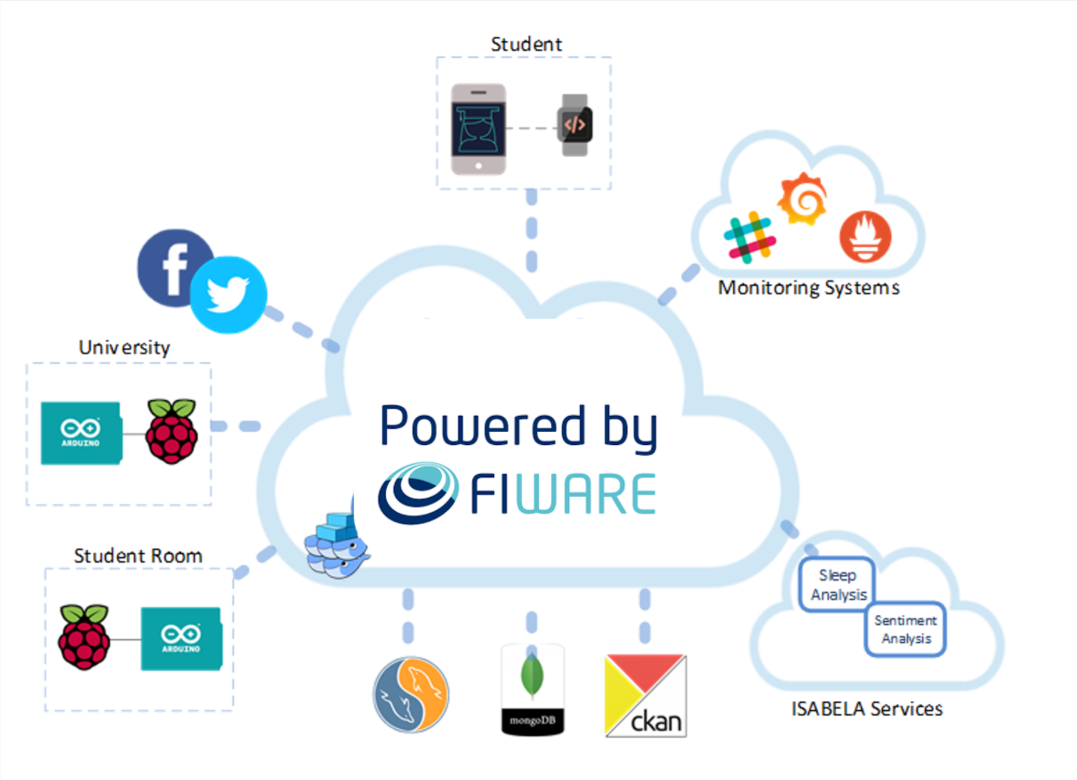



Figure 4.1: Technologies in ISABELA. *Adapted from [Fernandes et al., 2019, 2020]*

Because more powerful technologies could be exploited for data collection in stationary environments, the legacy technologies that will be presented in section 4.2.1.3 have not been adopted in the final architecture of ISABELA. Therefore, our research team developed from scratch an IoT set called "ISABELA box" as part of the components to monitor phenomena in stationary delimited environments, namely classrooms and students' rooms. The data collected in ISABELA include Global Navigation Satellite System (GNSS), Wi-Fi fingerprint, activity recognition, proximity, screen lock, phone calls and Short Message Service (SMS) statistics, alarm clock, OSNs public posts, noise, temperature, light, and humidity. This data is used for several inferences purposes such as both indoor and outdoor location, sleep detection, sentiment and emotion analysis [Fernandes et al., 2020].

ISABELA is an IoT people-centric platform that leverages the concept of Human-in-The-Loop Cyber-Physical Systems (HiTLCPS). The principle behind HiTLCPS is to infer the users' intents, psychological states, emotions, and actions, using this information then to determine the system's behaviour [Sousa Nunes et al., 2015]. This involves using a large variety of sensors and mobile devices to monitor and assess human environments.

The core element of the ISABELA system is based on FIWARE, a top reference IoT middleware[8]. In the next section will be describing the function of the main components from FIWARE that we leveraged in ISABELA.

#### 4.2.1.2 Backend

FIWARE is a European set of specifications to provide APIs that ease the development of smart applications in multiple vertical sectors. FIWARE is open, public and royalty-free. The FIWARE project started in 2011 within the Seventh Framework Programme (FP7) of the European Commission as part of the Future Internet Public-Private Partnership Programme (FI-PPP). On the one hand, the primary goal of FI-PPP was to advance a process for harmonized European technology platforms and their implementation. On the other hand, the aim was to provide integration and harmonization of relevant policies frameworks.

The process to establish FIWARE in the market was divided into three phases. The first phase was aimed at creating the technological core, while the second phase was mainly aimed at the implementation of FIWARE nodes. Finally, the last phase aimed primarily at creating a sustainable ecosystem for Small Medium Enterprises, through the selection of sixteen business accelerators. The ecosystem of developers and entrepreneurs met for the first time in December 2016, at Malaga City - Spain.

[8] `https://www.iot-survey.com/fiware`, accessed on 2019-03-13

FIWARE relies on a library of components called "Generic Enablers (GEs)", which offer reusable and commonly shared functions "As A Service". Through APIs, GEs allow developers to put into effect functionalities, making programming much more accessible by combining them. Overall, FIWARE provides an IoT system with harmonized data models and Resource Directory functionalities via standardized APIs. ISABELA is built on such a technological solution in order to demonstrate that it is possible to combine the flexibility and scalability of middleware with the robustness of well-established standards in the proposal of a unified management architecture. In the domain of ISABELA case study, we used several GEs from FIWARE. In Figure 4.2, we provide the data flow in the ISABELA system, with a focus to the FIWARE components.

The IDAS GE was used to deal with the sensing state and manage all kinds of IoT devices. IDAS supports management and interoperability capabilities between the IoT devices that communicate using a variety of protocols and the FIWARE platform. For practical reasons, we stared by leveraging the Ultralight agent to connect the IoT device from ISABELA boxes. The mobile sensors have been directly connected to their corresponding entities in FIWARE. ISABELA boxes allow collecting temperature, humidity, noise and luminosity data in the user's stationary environment (home and school).

HiTLCPS need to be scalable, and dynamic models are needed in order to represent the data of such systems. To have such a dynamic ecosystem, FIWARE adopted the NGSI9/10 information model, previously developed by the OMA [Krco et al., 2014]. The model is based on entities and attributes. Each entity has its type and is represented by attributes, using JSON format payload. All the entities in ISABELA are managed by the ORION GE. Using this API it is possible to create, delete and retrieve entities, and to update existing ones. ORION acts as a "broker" by allowing external systems (consumers) to make data subscriptions with specific rules to the entities and attributes. This approach is adopted to improve system distribution and scalability. ORION sends notifications to these systems when subscription rules are fired. The series of data sent by IoT devices are associated with the entities stored in ORION.

The ORION is not fit for storing historic data. Instead, the CYGNUS GE is the connector in charge of persisting certain sources of data towards several storages, using the Apache Flume technology and by subscribing to the ORION entities. Thus, when a new entity arrives at CYGNUS, the listener will put it in a specific channel and forward it to a third-party data storage (e.g., MySQL, MongoDB and Comprehensive Knowledge Archive Network (CKAN) channel). Databases usually do not provide APIs to retrieve the data to applications. Hence, another module of the FIWARE platform handles this issue, namely the Short-Term History (STH)-COMET GE. COMET provides a RESTful API with historic-queries capabilities and aggregated methods. Thus, each time the ISABELA application needs to access historic data, COMET will connect to the Databases and retrieve the data.

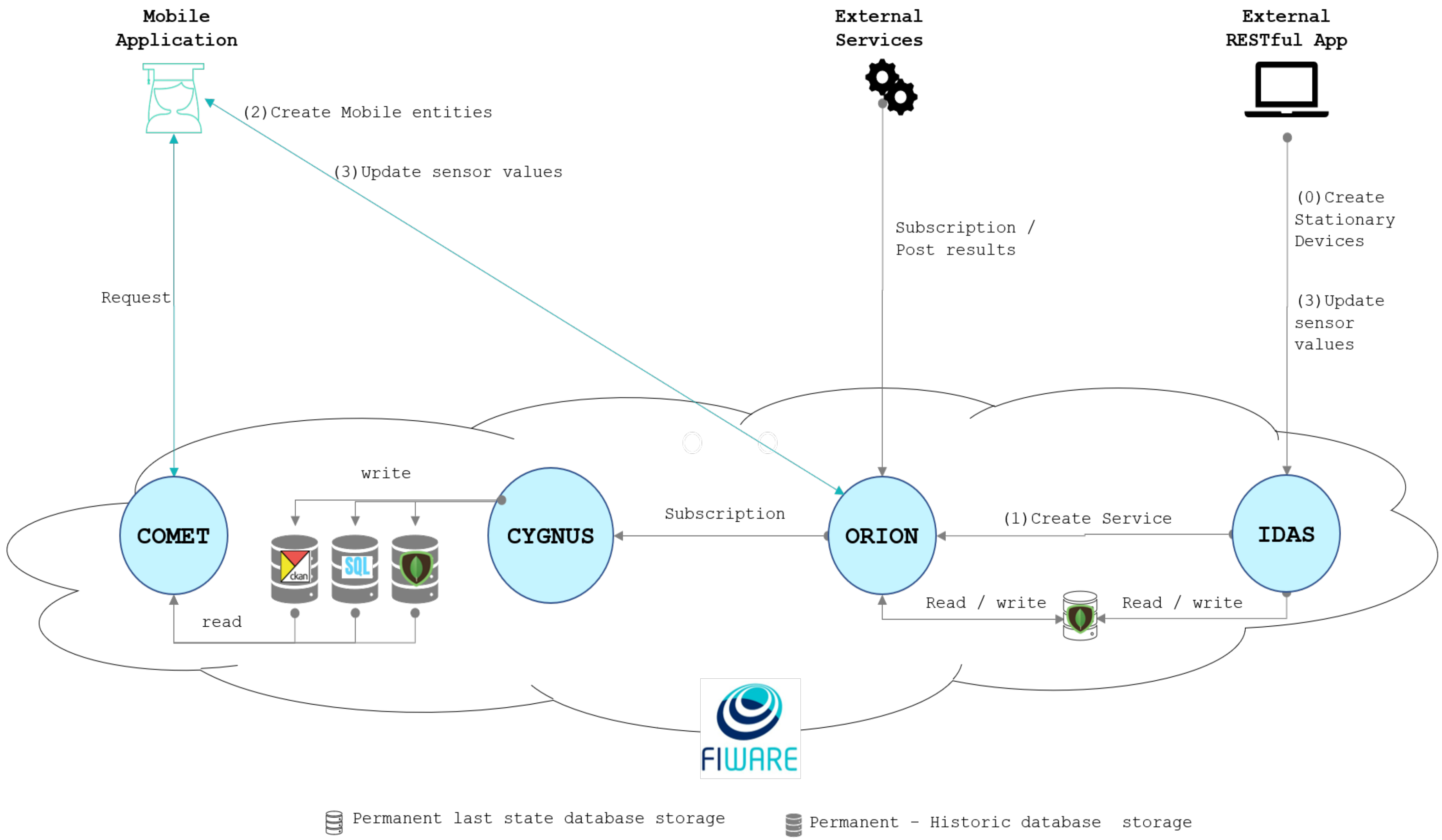



Figure 4.2: ISABELA data flow

Security and privacy are some of the most important requirements of HiTL-CPS. In order to protect communications between GEs, IoT devices, and applications, FIWARE leverages the Identity Management GE called "Keyrock". Keyrock, abstracted in Figure 4.2, allows adding authentication and security to devices, users, and applications, as well as authorization policies. Additionally, we anonymised personal data handled in ISABELA for privacy concerns.

Finally, ISABELA system provides feedback to the users through a chatbot, using a human-computer interaction technology based on natural language conversations, named "Dialog Flow" [Milton et al., 2018]. The ISABELA bot is capable of retrieving data from the several APIs available in the ISABELA ecosystem and subsequently provide user recommendations.

Following, we will see how leveraging FIWARE was a key approach to successfully integrate legacy EBSs, towards the development of people-centric applications [Armando et al., 2017].

#### 4.2.1.3 Legacy Technologies

The specification of a micaZ sensor board is shown in Table 4.6 [Sá Silva, Jorge et al., 2016]. Figure 4.3 shows the set tested. In the laptop, we installed a turnkey framework that included a Linux distribution and a packaged TinyOS, libraries, and nesC compiler [Sá Silva, Jorge et al., 2016]. We used a second VM for the FIWARE runtime.

Table 4.6: Specifications of the MicaZ - MPR2400

| | |
|---|---|
| **Microcontroller unit (MCU)** | Atmega 128L - Up to 16Millions of Instructions Per Second (MIPS) Throughput at 16MHz |
| | 4 KBytes |
| **EEPROM** | 4 KBytes |
| **Flash memory** | 128 KBytes |
| **Radio module** | CC2420 2.4 to 2.48 GHz IEEE 802.15.4, Advanced Encryption Standard (AES)-128 |
| **Transmission rate** | 250 Kbps |
| **Extensions** | Analog I, Digital I/0, I2C, SPI, UART |

The walk-through leveraged to integrate MicaZ EBSs in people-centric applications is the Following :

- Step 1 - In TinyOS, write a sensing application, then couple the MicaZ hardware to the *mda100* sensor board and the *mib520* programming board. Finally, connect the whole set to a Universal Serial Bus (USB) port of the TinyOS host machine. Install the application in the sensor node using the command */Oscilloscope$ SENSORBOARD=basicsb make micaZ install,1 mib510,/dev/ttyUSB0*. As a sensing application, we used the generic "Oscilloscope" program from TinyOS libraries (...apps/Oscilloscope). Disconnect the sensor node from the programming board and connect the Sink node to the host machine.

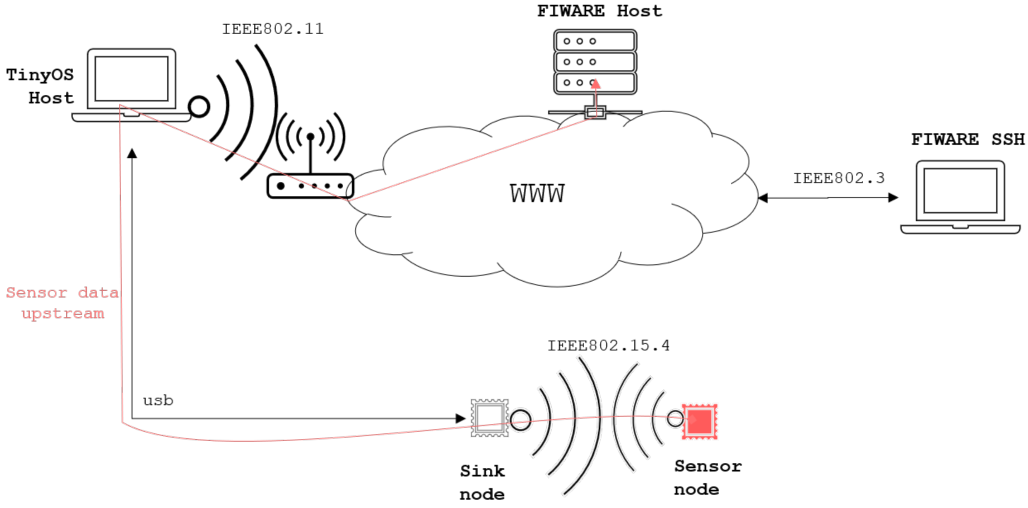



Figure 4.3: Setup for the integration of legacy technologies

- Step 2 - Staying in TinyOS, write the dispatcher program, then connect the Sink node to the *mib510* board. Finally, install the dispatcher in the Sink node using the command *make micaZ install,0 mib510,/dev/ttyUSB0*. Do not disconnect the whole set (sink and programming board) from the Laptop. Remark that we changed the TOS_ID_NODE (node identification number) to 0. By convention, the number 0 is reserved to identify the Sink node in a WSAN. As a dispatcher program, we used the generic *BaseStation* program from TinyOS libraries (...apps/BaseStation).
- Step 3 - In TinyOS, write a program to fetch sensor data from the dispatcher and store it in a text file. We used a modified "seriallisten" program from TinyOS libraries (...support/sdk/c/sf). By default, *seriallisten* enables to displays the results on the terminal screen. We edited the *seriallisten.c* file and inserted a function to store on-the-fly, the last sensor data in a text file. To simplify, we only retrieved and stored the last Byte of the sensed packets. In the testbed, this Byte represents a sensor data.
- Step 4 - Write a middleware program to upload the sensor data to FIWARE from the text file created in the previous step. The program should enable to create an entity, create a service, register the device and send the sensor data. Our solution was developed in Java language.
- Step 5 - Launch the VM that embeds the FIWARE instance, then run its services (docker-compose up -d).
- Step 6 - Open a Secure Shell (SSH) protocol session via *puTTY* terminal and connect remotely to FIWARE.
- Step 7 - Connect the laptop to the Internet. This step serves to simulate the uploading of the sensor data to FIWARE via the Internet backbone.

- Step 8 - Start the sensor node. It will run the application to collect and report the sensor data to the Sink. By using the generic application aforementioned, the green LED should start blinking in both nodes.
- Step 9 - Return to TinyOS and run the program to fetch sensor data from the dispatcher. Stop the application after retrieving at least one packet. If using the example in step 3, open a *Terminal* and run the modified seriallisten program with the command *...c/sf$ ./seriallisten /dev/ttyUSB1 micaZ*. Remark that now we use the USB Port number 1. In fact, there are two ports associated with the mib520 programming board. Port USB0 for configuration of the sensor nodes, e.g. to install applications in the nodes and port OS1 for the Application layer, e.g. to retrieve data from the sensor nodes [Sá Silva, Jorge et al., 2016].
- Step 10 - Run the middleware program. In our case, the IDE was installed in Microsoft Windows environment, and the text file produced by *seriallisten* is a shared file to both Microsoft Windows and Ubuntu over VM. If both the entity and its last attribute value have been uploaded to ORION context broker with success, an HTTP code is received in the IDE workstation added to the sensor value. Typically, a *201* means the request has been fulfilled, and the entity was created. A 20x code is a standard response for a successful request. For our testbed, we will assume that any other code translates an unsuccessful request.
- Step 11 - Ultimately, return to the SSH session and verify the existence of the entity created in ORION, with the value uploaded.

The setup for the integration of this legacy technologies is subject to several optimizations in both hardware and software environments. For the former, the laptop could be replaced by a smaller computing system like Raspberry Pis, to make the setting more practical in the deployment scenario. For the latter, one Virtual Machine is enough to run all the virtual systems in the laptop. Moreover, the WSAN runtime can be simplified by developing both the retrieving and the middleware program in only one language (Java or C). Indeed, both languages enable to develop the mechanism to fetch the sensor data from the dispatcher, then format it before uploading dynamically to FIWARE.

ISABELA comprises two external services, namely sleep analysis and Sentiment Analysis (SA). The former enables the inference of the users' sleep hours, based on data collected from both EBSs and SBSs. Comprehensive results of sleep detection can be found in literature as we explain in [Fernandes et al., 2019, 2020]. Because existing solutions had not yet successfully integrated HiTLCPS and OSNs [Sousa Nunes et al., 2015], we proposed an approach for that challenge as we will describe in the next subsection.

#### 4.2.1.4 Sentiment Analysis Service

The SA module from ISABELA leverages NLP tasks for IoT purposes. Specifically, it infers both sentiment and emotions of the textual objects collected from OSNs. The most straightforward outcome of the sentiment analysis task is: "Does a text express a positive or negative sentiment?". Usually, we assign a polarity value to a text in the [-1,1] interval, where 1 corresponds to very positive polarity and -1 corresponds to very negative polarity. What we generically call SA in our thesis, comprises sentiment and emotion analysis. Emotions analysis is a step further as compared to sentiments, which consists typically of positive, neutral and negative scores. The process for emotions analysis classifies texts into more complex scopes that involve nuances of feelings such as fear, fulfilment, frustration, boredom, excitation, happiness [Maia and Santos, 2018]. Figure 4.4 displays a generic architecture for an NLP system.

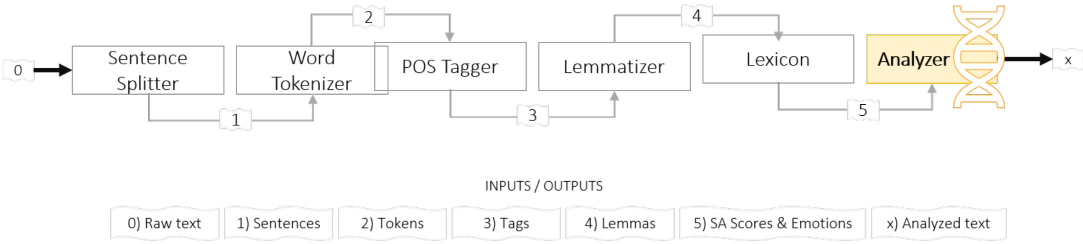



Figure 4.4: NLP generic architecture

In our case, the input to the system is a text that is pre-processed using a variety of linguistic tools/tasks such as "tokenization", "Part-of-Speech (POS) tagging", and "lemmatization". Focusing on sentence-level, we assume there is a single opinion in each sentence. This assumption can be extended by splitting the sentence into phrases that contain only one opinion each. This can be something as complex as a predictive classifier to identify sentence boundaries. A typical sentence splitter in the literature is based on a full stop in the text(.). The "tokenization" task consists of isolating each component of the text, while "POS tagging" consists of finding the grammatical classification of each of those components, e.g. adjectives, punctuation, verb, adverbs. Depending on the tag, the sentiment or emotional load may vary. In general, adjectives attract more attention from an NLP system, because they are considered to be more informative when it comes to emotions.

The "lemmatization" task is responsible for bringing all the components into their original form. For grammatical reasons, documents use different forms of a word, such as organize, organizes, and organizing. Additionally, there are families of related words with similar meanings, such as democracy, democratic, and democratization. Thanks to "lemmatization", all conjugated verbs will be turned into their infinitive form, plurals and gender will be eliminated, which reduces the complexity of analysing a text. Finally, the "classification" unit in Figure 4.4 is where different strategies are used to provide sentiment scores or emotion tags to the sentence.

Studying sentiment and emotions in a text demands complex design and computations, which is not yet perfect in the literature [Feldman, 2013]. The system must deal with complexity such as sarcasm, noisy texts, emojis, emoticons and slang. The Classification unit is where the linguistic resources are leveraged to annotate the pre-processed text with an overall mean value for the sentiment and all emotions tags in the text. These NLP results may be attached to whole documents (for document-based sentiment), to individual sentences (for sentence-based sentiment), or specific aspects of entities (for aspect-based sentiment). These annotations are the output of the NLP system, i.e., the classified text. The simplest classification unit is developed by utilising a set of lexicons as linguistic resources.

Lexicons are documents that comprise the sentiment scores, or emotions tags of the entities assumed to represent a written language, i.e. the Corpora/Corpus. As an example of lexicons, we have "Sentilex-Flex PT2" for retrieving the polarities [Silva et al., 2012], Lisbon Emoji and Emoticon Database (LEED) Appendix 1 [Rodrigues et al., 2018] and Affective Norms for English Words (ANEW)/ANEWPT [Soares et al., 2012], for retrieving the valence, arousal and dominance values in emoticons/emojis and text, respectively. Valence expresses the intrinsic "good"-ness, neutrality or "bad"-ness a token. Arousal expresses the level of excitement attached to the token.

With both valence and arousal scores of a token, proven approaches in the psychology literature, such as the "Russell Circumplex Model" [Posner et al., 2005], enable to associate its corresponding sentiment and emotion. On the other hand, proven lexicons, such as the LEED Appendix 2 and "Linguateca"[9], will enable to associate emotions to emojis and Portuguese tokens, respectively.

Leveraging the pipeline in Figure 4.4, we will show an example of essential NLP tasks in progress. In Listing 4.1, we have a sentence in the Portuguese language as input, which mean *"Wars generate unproductiveness"*.

Listing 4.1: NLP pipeline sample case

```
Raw text: "As guerras geram improdutividade"
Sentence: as guerras geram improdutividade.
Tokens: "As", "guerras", "geram", "improdutividade".
Tags: "art", "n", "v-fin", "n", "punc"
Lemmas "o", "guerra", "gerar", "improdutividade"
Sentiment classification: tokenlemmanotfound, -1,
    notfound, -1, notfound

We finally compute the average polarity, which is the
    sum of tags Polarities divided by the number of
    considered Tags and Lemmas in the text => -1
```

[9]https://www.linguateca.pt/acesso/corpos/dicemocoes.txt, accessed on 2018-07-12

Our idea was to leverage the capability of humans for providing contextualized data, which could be interpreted, leading to information and knowledge [Armando et al., 2018]. Hence, we designed and implemented the SA module depicted in Figure 4.5 on Java Spring platform. The SA module analyses texts produced by end-users in two popular platforms, namely Facebook and Twitter. On Facebook, we worked with the text written by the user in *"What's on your mind"* space. On Twitter, we worked with the text written by the user available in *"What's happening"* space, i.e. the tweets and retweets. With such text entries and applying NLP techniques, we tried to produce insights about the users' mood [Feldman, 2013].

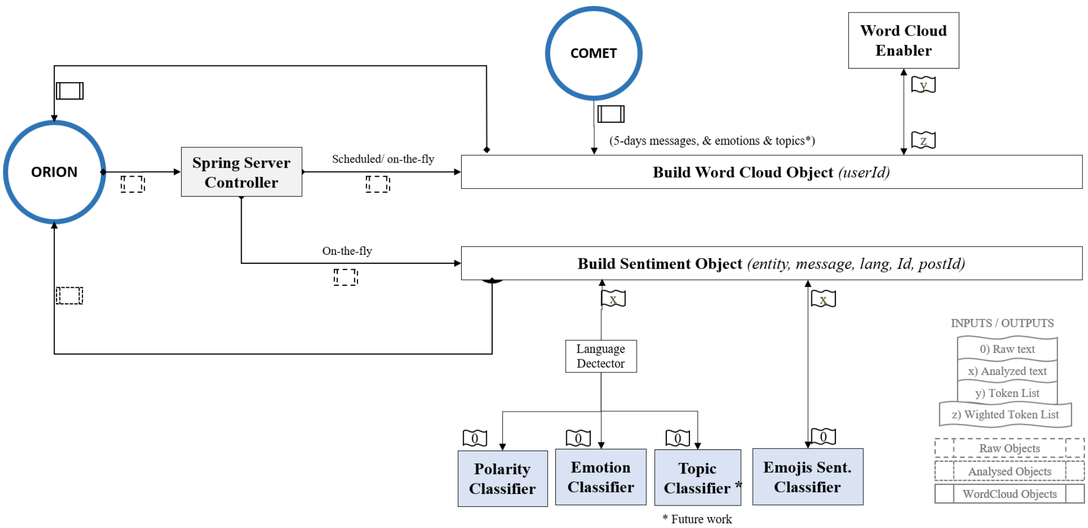



Figure 4.5: Sentiment Analysis module

The rest of the SA module works as follows. With an authorization from the end-user, the ISABELA application retrieves the posts in the OSNs and composes a JSON structure of the entity to be created at ORION. With a subscription in FIWARE, the OSN entities in ORION are echoed to the *Controller* class, which triggers the *Build Sentiment Object* class to:

- Identify the language of the message attribute;
- Send the message to the corresponding language module for performing polarity and emotion analysis, based on the text;
- Send the text to the corresponding language unit for polarity and emotion analysis, based on the contained emojis [Lou et al., 2020];
- Compose the analysed JSON entity and send it to ORION.

Given the scarcity of SA libraries for the Portuguese language, we leveraged the principles explained for Figure 4.4. We implemented a sample pipeline or the Portuguese language as a unit for the SA module. This implementation handles modifiers in tokens, that is, when a negative, a positive or a superlative adjective precede a token, we modify its Sentiment score as given in the lexicon.

Contrary to the Portuguese language, there are abundant SA implementations for the English language. We used the Valence Aware Dictionary and Sentiment Reasoner (VADER) library [Hutto, Clayton J. and Gilbert, 2014], which is both a lexicon and rule-based sentiment retrieving tool, specifically created for working with messy social media texts. The resulted polarity in VADER is based on the valence score normalized to [-1,1] interval, offering an accuracy of 60%+.

As it wil be will be presented in section 5.1.2.1, we display the SA results through charts, emojis and word clouds. Word clouds are widely used in the literature for infographics as they are very efficient tools to depict the representative keywords from a text [Cui et al., 2010].

As seen in section 3.3, the JSON payload provisioned to the LwM2M IoT agent is standardised in what is known as a "device model". Following, we will describe the device model, as well as the main configurations and code adaptations made to implement the unified management approach in ISABELA.

## 4.2.2 Device Models, Configurations and Code adaptations

### 4.2.2.1 LwM2M Device Models

A device model includes two main parts, namely the device attributes and their corresponding internal attributes mapped in FIWARE IDAS GE. Listing 4.2 is the generic device model in JSON format that we provide in the *IoTagent*.

We did not make a difference between a managed device and its corresponding entity. Hence, we have the same value to both the *device_id* and *entity_name*. In a device model, the (active) "attributes" represent both the state and the activity of the managed devices. The first three elements of the "attributes" in Listing 4.2 refers to the IEEE 1451 - TEDS information.

The attribute *Description* is to contain any relevant information attached to the managed entity. We did not adopt the IPSO resource named "Application Type" because it is a Read-Write attribute while in our case, we wanted a read-only attribute. The attribute *DateTime Stamp* is to add a contextualization to the information reported by the managed entities. Both *Sample Frequency Value* and *Blocking Status* attributes are to enable the tracing of the values reported in *Sample Frequency Value*, and the command *Set Block Status* in "lazy" (attributes), respectively. As concerning the "lazy" (attributes), we only implemented an element to enable the change of the data acquisition rate of the sensors on-the-fly. It is labelled *Sample Frequency* in the LwM2M Object and Resource Registry[10]. Sample Frequency "lazy" attribute has the IPSO objectID 6040, and it is defined as "How often, in seconds, the inputs are read/sampled".

[10] `http://www.openmobilealliance.org/wp/OMNA/\acrshort{lwm2m}/LwM2MRegistry.html#resources`, accessed on 2019-04-15

Listing 4.2: Generic device model

```
"devices": [ {
 "device_id": "IPSOObjectName_ISABELAUserId",
 "entity_name": "IPSOObjectName_ISABELAUserId",
 "entity_type": "Unifiedevice",

 "attributes": [
   Manufacturer, Model Number, Min & Max Range Values,
      Sensor Value, Sensor Units, Description, Sample
      Frequency Value, DateTime Stamp, Blocking Status
 ],
 "lazy": [Sample Frequency],
 "commands": [Set Blocking Status],
 "internal_attributes": {
 "lwm2mResourceMapping": {
 "attributeNamex": {
 "objectType": IPSOObjectId,
 "objectInstance": 0,
 "objectResource": IPSOReusableResourceIdx
},...]
```

We admit that in physics domain, the units for "Period" would have been $s^{-1}$. However, we thought it was more convenient to be aligned to an already conventional entity instead of creating another IPSO object such as the *Sample Period* or *Sampling Period*. To summarize, the "lazy" attribute is meant to indicate the throughput of the device's activities, i.e., data acquisition and actuation minimal intervals. The "commands" (attribute) in this version comprises a variable to enable blocking and unblocking the activity of the managed devices. Thus, it provides the IoT system with the capability of preventing a sensor from sending its readings to the backend. The device model structures the entities to be provisioned in the IoT agent. Typically, all the (active) "attributes" are registered with a Read-only capability, while the "lazy" (attributes) are of Read-Write types.

For the case of the virtual devices (Software-based or Human-based) in the mobile application, we created other activity files, where we import the corresponding Software Development Kit (SDK) methods, to emulate the device behaviour as it occurs in Android *sensorActivity* management class. For instance, the *get_Tweets_Service.java* file runs the *statusesService.homeTimeline()* method from *com.twitter.sdk.android.core.services*, which retrieves the maximum limit tweets/retweets in the users' "home_timeline".

Based on the structure in Listing 4.2, Listing 4.3 is the generic device model for SBSs. Thus, we chose to display an excerpt in Listing 4.3 only with the elements that are particular to SBSs as compared to Listing 4.2.

Listing 4.3: Device model for SBSs

```
...
"attributes": [
   Min & Max Range Values, Sensor Value , Sensor Units,
      Description, Sample Frequency Value, Parameters,
      Sensor Algorithm, DataTime Stamp, Blocking Status]
      ,
"lazy": [Sample Frequency, Set Parameters, Set
    Algorithm],...
```

In SBSs, it is relevant to track and choose the input variables leveraged to produce resulted in the virtual values. Besides, it is crucial to be able to configure the formula/algorithm used to compute the indirect values. For instance, in ISABELA, the "Sociability" or the "Alone Time" are SBSs that can be calculated differently, depending on the algorithm we adopt [Fernandes et al., 2019]. Also based on the structure from Listing 4.2, Listing 4.4 is tailored for HBSs.

Listing 4.4: Device model for HBSs

```
...
"attributes": [Language, Post Message, Post Id,
    Description, Sample Frequency Value, DateTime Stamp]
    ,...
```

The (active) "attributes" in Listing 4.4 comprise a *Post_Id* to identify the last retrieved tweet object in the Broker, before launching a new query to the OSN. For instance, the Twitter SDK requires a *Post_Id* from where it retrieves the objects in the "home_timeline". If the *Post_Id* is not given, the SDK will retrieve the last maximum objects, without any reference. Hence, there will be a risk to send duplicated objects to the ORION *context broker*.

Finally, Listing 4.5 shows the data model for a generic actuator, an addressable text, that we implemented in the case study. A string value from the *Management Center* defined in Figure 3.7 feeds the "lazy" (attribute) labelled "Set Parameters". In ISABELA, this attribute is meant to be updated by a teacher who will send personalized recommendations to a particular student or a group. The recommendations are received both in the form of notification and via the ISABELA chatbot. On its turn, the student can reply to the teacher, by witting a text directly in the chatbot textbox. This reply text feeds an attribute of the created entity in ORION with the same name. Since the *Reply Text* attribute is configured as an (active) "attribute", it will be observed by the IoT agent, so that whenever its value changes, it will be updated in the Broker. The *Set Parameters* attribute can contain values to configure an actuation task such as an LED toggling or a buzzer sound.

Listing 4.5: Device model for actuators

```
...
 "attributes": [
   description, Parameters, Reply Text, DataTime Stamp,
      Blocking Status],
 "lazy": [Set Parameters],
 "commands": [Set Blocking Status],...
```

Even if Haydarov [2017] proposed an effective implementation of LwM2M for Android OS, the proper communication with the FIWARE server had to be modified to make it work properly. Thus, we have successfully made it interact with FIWARE, despite some implementations incompatibility issues. The following subsection presents the main configurations and adaptation we have made in both the LwM2M clients and Server from the original codes.

#### 4.2.2.2 Main Configurations and Code Adaptations

LwM2M client for a mobile host

The configuration of the Android clients consisted firstly in creating a data model and the associated files for each object we wanted to manage. Then, we add new methods in the*MainActivity.java* file to implement the flows (1) to (4) in Figure 3.9. Every time the LwM2M client starts, the pre-provision of the configured IoT devices is performed in the Agent (CoAP Create operation). The logout from ISABELA application triggers simultaneously the deregistration and erasure of the managed objects stored at the *IoTagent* (CoAP Delete operation). We used the entities numbers (devices and attributes) both for the managed *ObjectId* and the associated *ReusableResourcesId*, according to the IPSO specifications. For the cases where we could not find the *ReusableResourcesId* for the attribute of an object, we overrode the one proposed by OMA, selecting the most similar number possible not yet registered within the institution's reusable resources range (2048 - 26240).

Another approach to overcome this issue was also to pick a number from the private resources range (26241 32768). For instance, the *Sample Frequency Value* 26040, was inspired from IPSO 6040 - "Sample Frequency".

LwM2M client for stationary host

When the stationary host is switched on, a script launches our tailored NodeJS-based client from *Telefónica I+D* project[11]. The *lwm2mclient.js* creates the managed objects, connects them to the Agent and finally runs the python script that drives the Arduinos sensors in the ISABELA Box, according to a *setInterval() scheduler()*. A function in *iotagent-lwm2m-client.js*, *childProcess.exec()*, fetches the *stdout* result from the Arduino sensor.

[11] `https://github.com/telefonicaid/lwm2m-node-lib`, accessed on 2019-07-30

For instance, the following code excerpt will enable to update the changes in the temperature sensor values provided by the Adafruit-based python script[12]:

*lwm2mClient.registry.setResource('/3303/0', 5700, stdout.trim(), handleObjectFunction).*

LwM2M dedicate NodeJS Server

Despite both Android client and the dedicate *lwm2m-iotagent* being compliant to the same protocol, i.e. LwM2M, the latter did not decode the received values properly. The client sent the payload in Type-Length-Value (TLV) format, but the read values were in plain text. Concerning the numbers, for instance, we received values in American Standard Code for Information Interchange (ASCII) instead of float or double as sent by the client. Since the imported Leshan files in the Android clients are protected codes, we had to adapt at the server-side, as shown in Listing 4.6.

Listing 4.6: Configuration of the payload formats in lwm2m-node-lib

```
formats: [
  { name: 'application-vnd-oma-lwm2m/text',
    value: 0 // changed from ex 1541
  },
  { name: 'application-vnd-oma-lwm2m/tlv',
    value: 11542 // old 1542
  },
  { name: 'application-vnd-oma-lwm2m/json',
    value: 11543 // old 1543
  },
  { name: 'application-vnd-oma-lwm2m/opaque',
    value: 11544 //old 1544
  } ,
  { name: 'application-vnd-oma-lwm2m/octet-stream',
    value: 42
  },
  { name: 'application-vnd-oma-lwm2m/link-format',
    value: 40
  }
 ],
 writeFormat: 'application-vnd-oma-lwm2m/text'
};
```

Moreover, we had to insert another layer of conversion in *lwm2m-node-lib//InformationReporting.js/ dataHandler: function (chunk) {}*, to overcome the received format of the numerical values (Float, Double). We have also added reconnection patches in both mobile and stationary hosted LwM2M clients.

[12] https://github.com/adafruit/Adafruit_Python_DHT/tree/master/examples, accessed on 2019-01-03

In fact, sometimes the former elements disconnect from the Server (iotagent.js) and do not reconnect by themselves. In the end, we created an image of the LwM2M server from *Telefónica I+D* GitHub project[13], containing mainly the following patches:

- In *commonLwm2m.js* file, a *trim()* function to delete spaces added by the Android LwM2M client, at the beginning of the Uniform Resource Identifier objects;
- In *informationRepport.js* file, we call a *config.js* file that accepts "text/-plain" in *create.observe()* function;
- In *deviceManagement.js* file, we also call a *config.js* file that accepts text/plain in *createrequest()* function.

The main codes leveraged in the ISABELA case study and the produced datasets are available online at URL: ***https://bit.ly/2OyRBTt***. The repository in the indicated URL comprises one folder named "codes" that contains the FIWARE instance. The "codes" folder also contains a shell script named "trace" used to capture the server inbound-packets in Figure 3.9, i.e., from both Ultralight and LwM2M clients towards the corresponding IoT agents in FIWARE server. There is also another shell script to generate *.csv* files from Tshark. Tshark is a Linux command-based version of Wireshark for network packet analyser[14]

Finally, "codes" folder comprises both the Android application and the components of the Stationary Clients in Figure 3.9. The other folder in the URL is named "dataset". It comprises a .xlsx file named "Charts", where we put the detailed charts and figures of the experiments. Inside the "dataset", we also have two folders, namely *.../mob and /statio* for mobile and stationary hosted clients, respectively. Each of these folders contains .cap documents and the corresponding .csv formats. *.cap* files are registries of the sensor data from the clients towards the server.

Electronic-based devices in IoT are generally hard-programmed to be compliant with a specific management protocol that communicates with the corresponding IoT Agent in the middleware. However, some IoT entities may be adapted to different communication protocols. It is, for instance, the case of the integration of legacy EBSs technologies as described in section 4.2.1.3.

In this context, the main goal of the next subsection is to provide answers to the following question:

- Which aspects are to be considered for a dynamic choice of the IoT agent from a middleware?

[13]`https://github.com/telefonicaid/lightweightm2m-iotagent.git`, accessed on 2019-07-30

[14]`https://www.wireshark.org/docs/man-pages/tshark.html`

### 4.2.3 Ultralight versus LwM2M

We wanted to assess the communication performances of the LwM2M 1.0 implementation that we leveraged in the latest version of ISABELA, by comparing it to the transmission of the sensing values over Ultralight 2.0, which is a very popular API in FIWARE.

Table 4.7 helps us situating LwM2M and Ultralight services APIs into the TCP/IP communication model. Table 4.7 only covers the list of standards and protocols that we leveraged for the assessment of both services in FIWARE.

Table 4.7: Device management and application support. *Adapted from M. Koster, 2015*

| **Standards and Protocols** | | **Communication Layers** | **Management Layers** |
|---|---|---|---|
| Management Capabilities | | - | Application Software |
| IPSO Objects | Simplified JSON | - | Data Models |
| LwM2M 1.0 | Ultralight 2.0 | - | API and Services |
| CoAP | HTTP | Application | - |
| UDP | TCP | Transport | - |
| IPV4 / IPV6 | | Network | - |
| IEEE802.11 / IEEE802.3 | | Data Link and Phy | - |

#### 4.2.3.1 LAN Performance Setup

We mounted the testbed presented in Figure 4.6 to compare the communication performance of Ultralight and LwM2M in a mobile scenario. This testbed enabled to capture both outbound and inbound packets from the mobile device and the IoT agents, respectively.

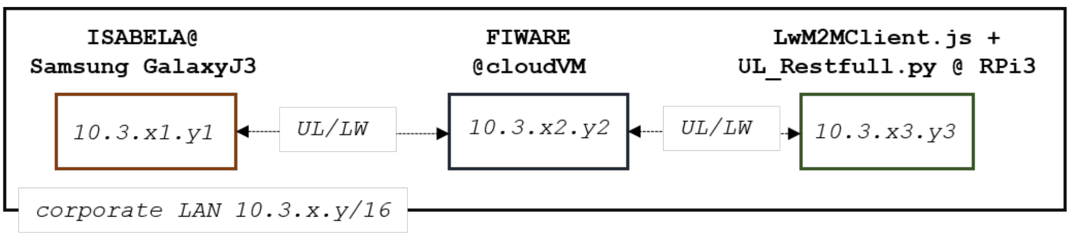



Figure 4.6: Ultralight versus LwM2M. LAN performance setup

The testbed is composed of three modules, namely mobile phone in the left side of Figure 4.6, stationary device in the center and server on the right side. The mobile phone runs the ISABELA Android application, which implements the mobile client modules of Ultralight and LwM2M . The applications imports the LwM2M library from [Haydarov, 2017]. Since Ultralight is a simplified JSON codification for the payload transmitted over HTTP, its corresponding entity in the mobile application is developed with native libraries from Android SDK.

The stationary device is a Raspberry Pi, which runs both Ultralight and LwM2M clients. The former client module also includes a python script that creates and sends a series of sensor values to the Ultralight agent in the server. The latter client of the testbed is an adaptation of the NodeJS-based client from *Telefónica I+D*[15].

Finally, the server module runs a cloud-based VM where we installed a docker environment wit the suite of containers that implement GEs from FIWARE. We ran ORION context broker and the IoT agents for both LwM2M and Ultralight. We remind that the IoT entities are created in the *Broker*. Then, external data consumers subscribe to the Broker to receive the last snapshot of the entities whenever the sensor values they observe change.

The delta ($\Delta$ ) between the timestamp of the packet at the source and when it is captured in ORION produces the End-to-End (E2E) Delay. The Network Delay is calculated as a delta between the timestamp when the packets arrive at the server interface and the timestamp that the packet leaves the clients' interfaces. The Used Bandwidth measurements included all types of packets generated during the tests such as *connection*, *retransmissions* and *posts*. Finally, the number of received packets in the Broker divided by the number of sent packets that have been sent from the clients gave us the values for the Delivery ratios.

We sent the traffic from clients to the server according to four scenarios, namely:

- (1) Mobile Client to Server over Ultralight Service
- (2) Mobile Client to Server over LwM2M
- (3) Stationary Client to Server over LwM2M
- (4) Mobile Client to Server over Ultralight

In preliminary tests, we wanted to submit the scenarios to the same network conditions. Hence, the traffics for these scenarios were sent simultaneously. The resulted have shown that the delays of traffic over LwM2M were heavily more delayed than Ultralight. We changed the testbed approach and opted in running the four scenarios separately. However, the packets delivered via LwM2M, transported over CoAP/UDP, were still considerably more delayed than packets delivered via Ultralight services, transported over HTTP/TCP. In *Appendix C - Ultralight vs LwM2M. Comparative Performances in a Corporate LAN* shows the obtained results.

Finally, we came up with a third approach by controlling the network conditions, including both the QoS and synchronization aspects. The approach consisted of running the whole performance tests inside a personal computing. Following, we will describe the third approach.

[15]`https://github.com/telefonicaid/lwm2m-node-lib/bin/iotagent-lwm2m-client.js`, accessed on 2019-07-30

#### 4.2.3.2 Controlled Network Performance Setup

Like in section 4.2.3.1, we mounted the testbed presented in Figure 4.7 to capture the packets sent from both the mobile and stationary hosted clients to the server.

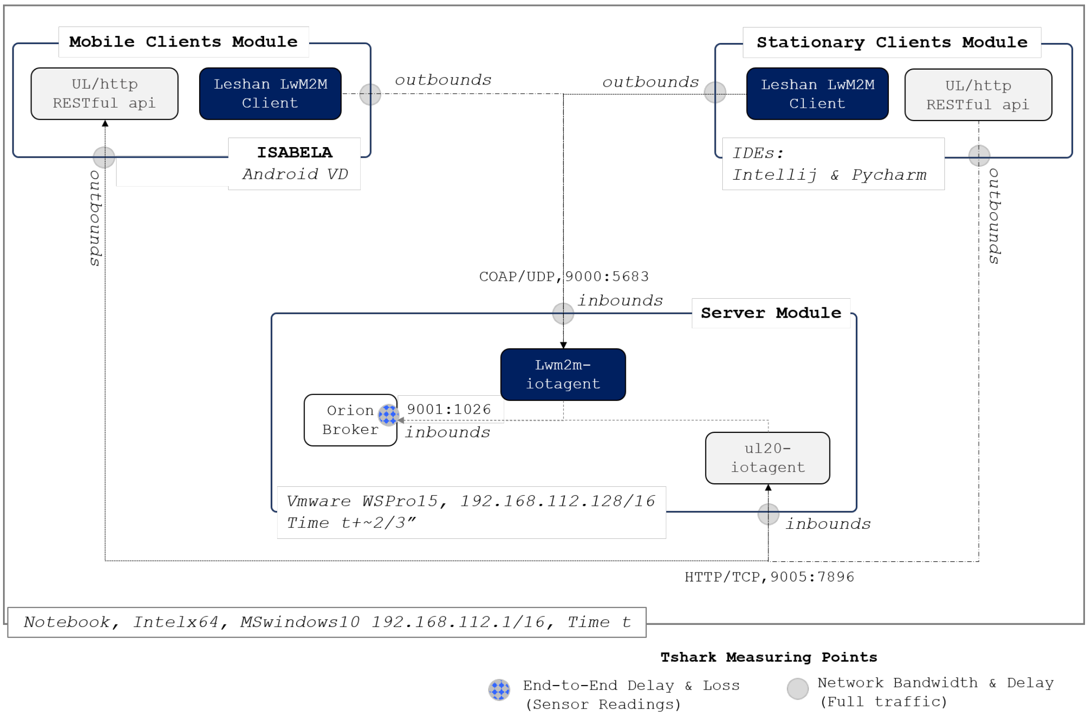



Figure 4.7: Ultralight versus LwM2M. Laptop performance setup

The testbed was also composed of three modules, namely mobile clients, stationary clients, server. An Android application implemented the mobile client module importing the same LwM2M library as in section 4.2.3.1. Likewise, the stationary client module of this testbed is an adaptation of the NodeJS-based client. Here, the stationary client module also includes a python script that creates and sends a series of sensor values to the Ultralight agent in the server. The server is an instance of FIWARE where we have leveraged the STH-COMET, abstracted in section 4.2.3.1, as a third GE to register the historic data from the Broker.

Because the synchronization between both clocks is not stable [Vmware, 2019], we set them manually. We advanced the reference to the server's clock of 2-3 seconds as compared to the reference of the clients'. This choice enables us to avoid negative values in the delays that we calculated.

We chose the same relevant indicators to measure the performance of the selected scenarios, namely packet loss, delays and bandwidth occupancy.

The sensor values sent by the clients were fictitious. In fact, the sensor values were composed of an integer that is the packet ID and the Unix-Epoch timestamp, in milliseconds, that indicates when the sensor value was produced.

A dot splits the two numbers. E.g., *1.1xx* is the 1$^{st}$ sensor value produced by the client module at the epoch stamp of *1xx* (unity of time). Listing 4.7 depicts an example of one packet-capture received in the Broker[16] for the LwM2M entities from the stationary client. As we can see in the *String value* from the Listing 4.7, this packet was produced at epoch time 1573351711953999872 nanoseconds and was received at epoch time 1573351711.996911019 seconds[17]. In the example from Listing 4.7, the E2E delay would be 1573351711996 *minus* 1573351711953 = 43 milliseconds (ms) .

Listing 4.7: Wireshark packet dissection of an entity creation process

```
v Frame 4: 478 bytes on wire (3824 bits), 478 bytes captured (3824 bits) on interface 0
  > Interface id: 0 (br-df66889607e2)
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov 10, 2019 02:08:31.996911019 GMT Standard Time
    [Time shift for this packet: 0.000000000 seconds]
    Epoch Time: 1573351711.996911019 seconds
    [Time delta from previous captured frame: 0.000476498 seconds]
    [Time delta from previous displayed frame: 0.000476498 seconds]
    [Time since reference or first frame: 0.000617140 seconds]
    Frame Number: 4
    Frame Length: 478 bytes (3824 bits)
    Capture Length: 478 bytes (3824 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp:http:json]
    [Coloring Rule Name: HTTP]
    [Coloring Rule String: http || tcp.port == 80 || http2]
> Ethernet II, Src: 02:42:ac:12:00:07 (02:42:ac:12:00:07), Dst: 02:42:ac:12:00:05 (02:42:ac:12:00:05)
> Internet Protocol Version 4, Src: 172.18.0.7, Dst: 172.18.0.5
> Transmission Control Protocol, Src Port: 40588, Dst Port: 1026, Seq: 1, Ack: 1, Len: 412
> Hypertext Transfer Protocol
v JavaScript Object Notation: application/json
 v Object
  v Member Key: contextElements
   v Array
```

[16]interface ID *br-df66889607e2*

[17]Equals to 1573351711996911019 nanoseconds

Listing 4.7 (Cont.): Wireshark packet dissection of an entity creation process

```
    v Object
     v Member Key: type
        String value: box_ebs
        Key: type
     v Member Key: isPattern
        String value: false
        Key: isPattern
     v Member Key: id
        String value: loudness_7b7d
        Key: id
     v Member Key: attributes
       v Array
          v Object
             v Member Key: name
                String value: sensor
                Key: name
             v Member Key: type
                String value: string
                Key: type
             v Member Key: value
                String value: 1.1573351711953999872
                Key: value
          Key: attributes
    Key: contextElements
  v Member Key: updateAction
     String value: UPDATE
     Key: updateAction
```

The used bandwidth for each scenario defined in section 4.2.3.1 also included all types of packets generated during the tests such as *connection*, *acknowledgements*, *retransmissions* and *posts*. Finally, the number of received packets in the Broker divided by the 250 packets sent from the clients gave us the values for the delivery ratios. We believe that 250 packets were enough to generate the datasets that we wanted to analyse. Indeed, the structure of the sensor data is unchanged, and we assumed that the entities in Figure 4.7 process equally the structure of any sensor data they receive.

Table 4.8 summarizes the various test suites. The second column identifies the test scenarios, as described in section 4.2.3.1. In the third column, the type of client is identified, namely, mobile or stationary. The fourth column is named "Sample Frequencies."[18]. The "Sample Frequencies" are ordered as they have been launched in the trials.

[18]IPSO entity number 6040. It presents the period, in seconds, between sampling instants

Table 4.8: Timestamps of the captured frames

| **Order** | **Test Scenarios** | **Type of NCAP** | **Sample Frequencies** | **Comments** |
|---|---|---|---|---|
| $1^{\text{st}}$ | (1) | Mobile | 10 to 1 | Continuously |
| $2^{\text{nd}}$ | (2) | Mobile | 10 to 1 | Continuously |
| $3^{\text{rd}}$ | (3) | Stationary | 10 to 7 | Continuously |
| $4^{\text{th}}$ | (3) | Stationary | 6 | "Sample Frequency" 7 to 6 interrupted for $\tilde{1}$1 hours |
| $5^{\text{th}}$ | (4) | Stationary | 10 to 1 | Continuously |
| $6^{\text{th}}$ | (3) | Stationary | 5 to 1 | "Sample Frequency" 6 to 5 interrupted for $\tilde{3}$ hours |

E.g., 10 to 1 means that the first trial was run with a data acquisition rate of 10 seconds, and then we decreased the "Sample Frequency" by one until a "Sample Frequency" of 1 second was reached in the last trial. For the test scenarios 1, 2, and 4, a single test suite per test scenario have been obtained, with no interruption in the sequence of "Sample Frequencies" mentioned above. In the case of test scenario 3, we forced two interruptions to study the reaction to different networks and processing conditions.

The first interruption lasted around 11 hours between "Sample Frequencies" 7 and 6. During this time, the laptop was switched to sleep mode, with many applications having been shut down, beforehand. The second interruption lasted around 3 hours between "Sample Frequencies" 6 and 5. Contrary to the previous interruption, here the laptop was running many applications, including test scenario 4. The effects of these interruptions will be discussed in section 5.2, along with the results of the performance assessment in a controlled network.

In the next section, we will give the technical insights about the second case study that we developed to validate the architecture we proposed for unified management in "new" IoT.

## 4.3 5GOpenclasses

During the performance tests described in section 4.2.3, we registered an entropic behaviour from the implemented version of the LwM2M management service from FIWARE. Moreover, the implementation of an Ultralight agent that supported lazy/passive attributes was not available yet[19] by the time we started selecting the best IoT agent for the generic architecture in Figure 3.7. Therefore, adopting one of these services for the management pipeline in a TRL 7/8 product was not timely yet.

We left for future works the investigations of the issues in the maturity/stability of the existing implementations for both Ultralight or LwM2M services in FIWARE.

[19] `https://fiware-iotagent-ul.readthedocs.io/en/latest/usermanual/index.html`, last access on 2019-01-06

As a consequence, we also postponed the implementations of the functionalities in Figure 3.11 attached to the outcome of these investigations. We prioritized the development and testing of the functionalities that we needed to set the scenario displayed in Figure 4.8. With this scenario, we could ensure the most needed capabilities from our products to the integrated demonstration workshop scheduled for September 2020. By the time we finished the redaction of this dissertation, the *Mobilizador 5G* project was still going on for at least six months.

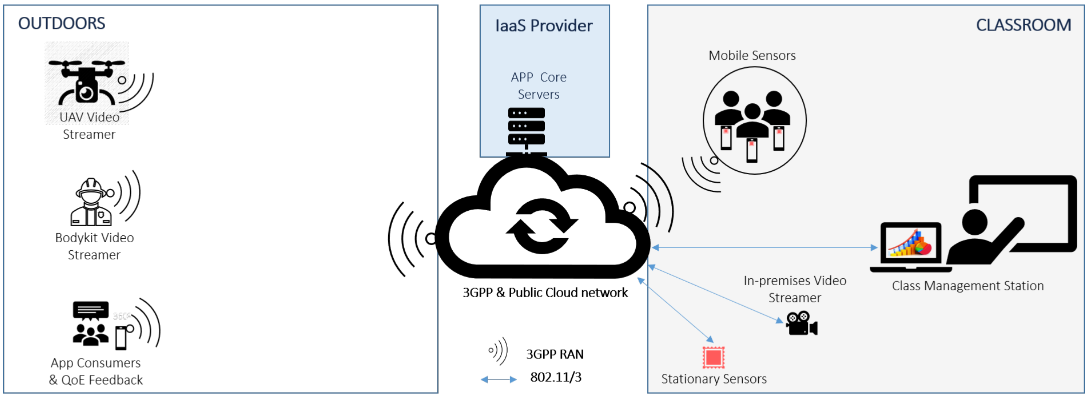



Figure 4.8: *5GOpenclasses* test-cases scenario

As introduced in section 3.4, students will be divided into two groups based on the location of their UE. When the UE is located inside the class premises, the users will be considered as a *mobile sensor data producer*. Thus, they should not be able to chat, ask questions and watch the live streams. However, they will provide sensors data from the mobile or smartwatch if necessary.

In Figure 4.8, each student is associated with a UE running an android applications. Students can chat, ask questions, watch the live streams and consume resources from a multimedia repository. Besides the mobile application, we also developed a webbased (dashboard) application to serve the teachers. Via this application, a teacher can trigger or stop the live streams, manage the class sessions and the repository. Live video streams are produced by three types of sources, namely UAV, Bodykits and in-premises stationary cameras.

## 4.3.1 Backend

### 4.3.1.1 Video Streaming

We used a Kurento Web Real-Time Communication (WebRTC) server to work as a receiver and transmitter for real-time video streams [Revision, 2020]. The typical capabilities of a Kurento are displayed in Figure 4.9. While standard WebRTC media servers' tasks include transcoding, Multipoint Control Unit (MCUn) and recording, Kurento server adds tasks such as flexible processing, augmented reality, blending, mixing and analyzing.

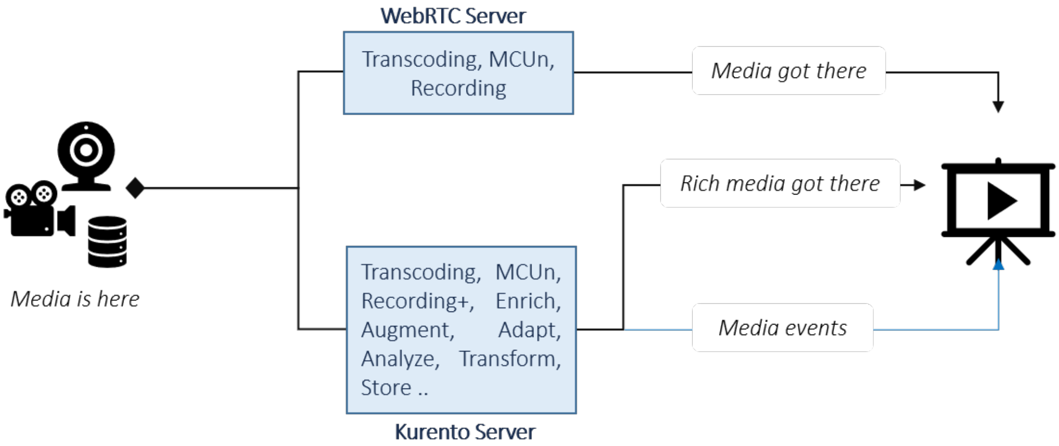



Figure 4.9: Capabilities of the Kurento media server. Adapted from [Revision, 2020]

We remind that the web-based application will provide the features to teachers for recording and transmitting video to students. These video streams will be redirected to a WebRTC multimedia server from where students will be able to visualize through the mobile application.

#### 4.3.1.2 Data Structure

HTTP requests between the end-user applications and FIWARE can be described as exchanging data objects, including login/registration information, classroom information, classroom issue information, materials uploaded to the repository. The structure of the data object storage is now composed of the following attributes:

- Users: id, type, password, is_online, in_class, TimeInstant, gyroscope, accelerometer, proximity, sound, light, create_time, update_time;
- Teachers: id, type, name, password, create_time, update_time;
- Rooms: id, type, password, name, description, is_online, recording, prof_id, course_id, create_time, update_time;
- Courses: id, type, academic_year, profs, create_time, update_time;
- Questions: id, type, asking, room_id, user_id, create_time, update_time;
- Group_Rooms: id,type,creator_id,password,create_time,update_time;
- Ratings: id, type, comment, rate, recommend, room_id, create_time, update_time;
- Media_Files: id, type, download_path, file_name, room_id, create_time, update_time;
- Environment: id, type, TimeInstant, light, noise, temperature.

All passwords are encrypted in a SHA-256 encryption format. The update_time attribute is updated to the current date in ISO8601 format (e.g., 2020-02-19T13:58:26.00Z). Besides, *is_online* serves to inform whether students are or not online. The media files can be stored in different formats, such as .ppt, .pdf, or audio and video extension. The *download_path* attribute indicates the path to download a file.

### 4.3.2 Frontend

#### 4.3.2.1 Web Application

We developed the web application in Python using the Dash by Plotly framework[20]. The Dash by Plotly framework is an open-source framework dedicated to building data analytics web applications based on Flask, Plotly.js, and React.js.

The construction of this end-user application relies on the use of a WebRTC Kurento. We thus can stream video in real-time with the architecture of *One to Many video calls*. We chose this server architecture because our target scenario is the provision of tools that enable the teacher to transmit its classes online and in real-time to their students. As for real-time video streaming, teachers can record the contents of the computer screen (e.g. for PowerPoint presentation), as well as record their webcam (e.g. to answer questions asked by students during class sessions). Finally, they can manage their classes, share complementary materials to the classes, and view the list the subjects assigned to then. All this data is stored on the central databases through FIWARE.

The features described above belong to the part relating to teachers and the management of their classes. Regarding the part of management provided by the administrator of the *5Gopenclasses* system, the administrator can add new subjects to a given teacher, view the current list of classes that belong to each subject, the files in the data repository, and control and analyze the feedback provided by students within the scope of a QoE process.

#### 4.3.2.2 Mobile Application

We used a library that supports WebRTC communications for Android OS. This library, *webrtcpeer-android*[21], is in charge of making communications with our Kurento WebRTC server and also for providing the functionality of presenting the real-time video stream from Kurento WebRTC server.

We remind that the mobile application will only receive real-time video content from the teacher since the communication architecture with the Kurento multimedia server is *One to Many video calls*.

[20] `https://plotly.com/`, accessed on 2020-04-14

[21] `https://github.com/nhancv/nc-android-webrtcpeer`, accessed on 2020-04-14

We can thus identify that the stream comes only from 1 teacher, while the students can be *N*. The communication of these video streams will be one-way, i.e. from the source of the video (teacher web-based application) to student video destination (mobile application). Besides the video streams from in-premises cameras, we can also stream video contents from outdoors via a Janus WebRTC server[22] for both UAV and Bodykit. Janus is a server that implements extra functionality, beyond the WebRTC real-time communications features initially designed. In the case of our Android application, we have a classroom dedicated to the cameras where the video stream will be received and presented.

Regarding the presentation of the video directly in this mobile application, we had some issues with displaying the video stream on a *SurfaceViewRenderer* from this Janus server. In the end, we decided to use a *WebView* and present this page of video streams in our mobile applications. This *WebView* allows layout content is to be displayed as if it were a web page.

## 4.4 Summary of the Chapter

In this chapter, we have presented the technical aspects leveraged to implement the scenarios for the validation of our models, pertained to the generic architecture in Figure 3.7. Thus, this chapter covered the set of solutions in terms of both hardware, software and frameworks that were have been leveraged from the state-of-the-art. We have also explained to which extend such set of tools served for our implementations. . Likewise, we detailed as much as possible the development and configuration choices that we have made. Finally, this chapter is written in a tutorial way to enable the reproducibility of most of the experiments as much as possible.

[22] `https://janus.conf.meetecho.com/`, accessed on 2020-04-14

# Chapter 5

# Obtained Results and Discussion

## Contents


**5.1 Management Functionalities** . . . . . . . . . . . . . . . **80**

5.1.1 Off-the-Shelf Testbed . . . . . . . . . . . . . . . . . 80

5.1.2 ISABELA Case Study . . . . . . . . . . . . . . . . . 82

**5.2 Ultralight versus LwM2M** . . . . . . . . . . . . . . . . **87**

5.2.1 Used Bandwidth . . . . . . . . . . . . . . . . . . . . 87

5.2.2 Delays . . . . . . . . . . . . . . . . . . . . . . . . . 88

5.2.3 Loss Ratio . . . . . . . . . . . . . . . . . . . . . . . 91

5.2.4 Summarizing the Comparison . . . . . . . . . . . . . 93

**5.3 Summary of the Chapter** . . . . . . . . . . . . . . . . **95**

We leveraged off-the-shelf technologies both in hardware and software, standardised semantics and protocols in *an approach to the unified management of heterogeneous IoT environments*. We have also built a PoCs and developed study cases where we performed both management functionalities and the assessments of the services that supported them. In this chapter, we will present the results that have been achieved in the different applied scenarios for the proposed architecture in Figure 3.7. Finally, we will make both a qualitative and quantitative analysis of the achievements and limitations that we identified in these scenarios.

## 5.1 Management Functionalities

### 5.1.1 Off-the-Shelf Testbed

This implementation was used for performing essential device management operations, following the ITU-T's reference model, namely read, write and execute operations. We used a browser on localhost:8080, in order to access the server management interface.

This interface in Figure 5.1 allowed access to:

- LwM2M Server: the default Leshan server configuration information;
- Device: default Leshan gateway configuration information;
- Location: default Leshan gateway location information;
- Temperature EBS: electronic-based transducer management interface;
- Comfort SBS: software-based entities management interface;
- Polarity HBS: human-based transducer management interface.

By clicking the first *Read* button from the interface in Figure 5.1, we performed the respective read operation and retrieved the set of stored values in the management database for these managed sensors. By clicking the first *Observe* button, we enabled the whole interface to be updated as changes were occurring in the management database resources. By clicking the *Write* button, we performed a write operation, i.e., we updated the content of the resource values both in the Leshan client variables and in the corresponding management database. Finally, by clicking the *Exec* button, we performed an execute function configured to reset the values indicated in the corresponding resources labels.

Although this was a straightforward implementation, it allowed us to validate the proposed conceptual approach and get a general feeling on how *unified management of heterogeneous IoT environments* could be deployed and work. For the next step, we have overcome the limitations of this prototyped testbed and developed use-cases of the unified management that could be validated in real scenarios.

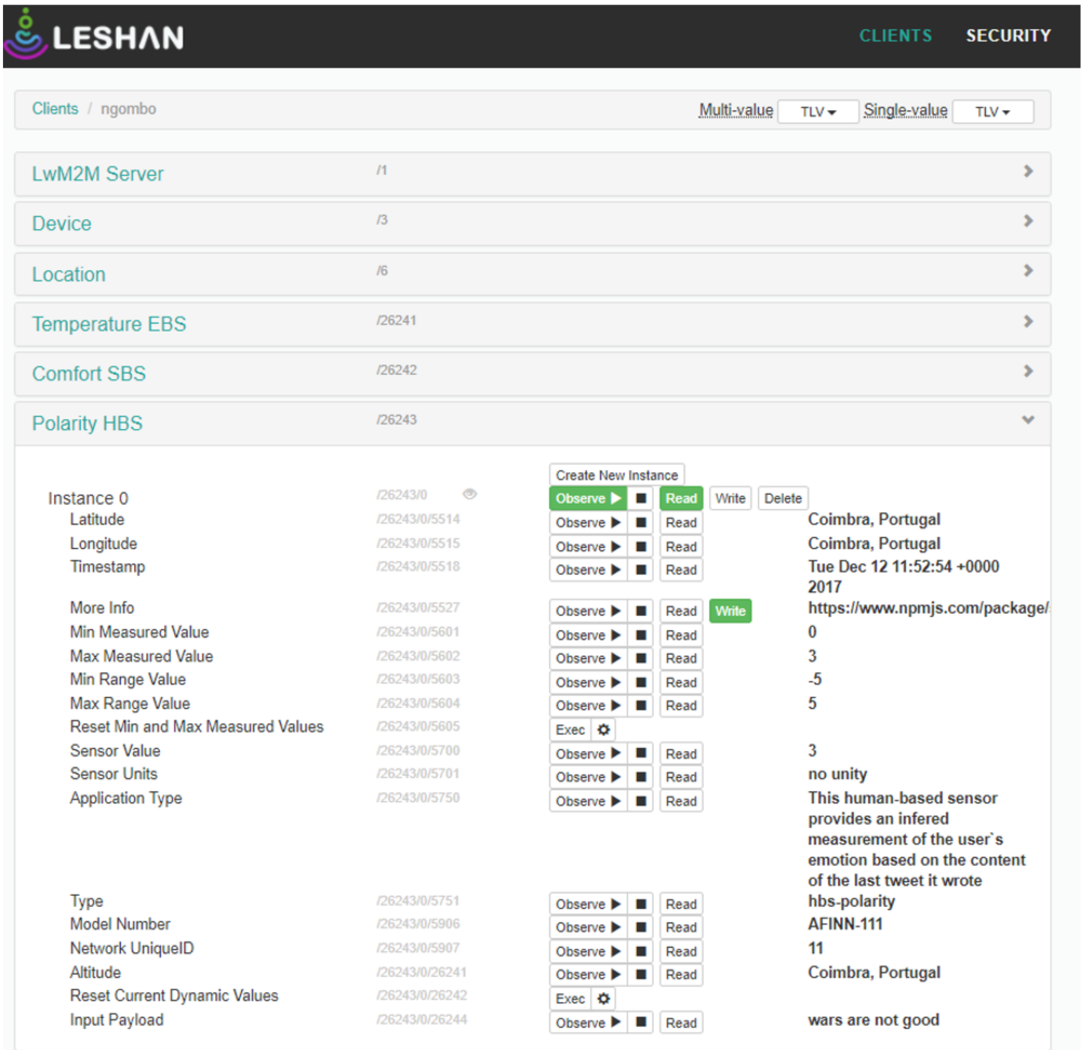


Figure 5.1: Interface and results of unified management testbed

In ISABELA case study, we leveraged FIWARE for combined data and device management. Such an approach enabled the use of a standardised data model and made the most a middleware, i.e., enhanced both the flexibility and scalability of our solution. We have also replaced the Node-red flow from the unified management straightforward implementation. Thus, we leveraged more advanced and open/tuneable solution to perform SA in OSN contents, both for English and Portuguese languages. With the new analytic tools n NLP, it was possible to go beyond the polarity measurements. Specifically, we explored the application of aspect-based SA to identify the target of the sentiment better, and emotion analysis to get a finer-grain analysis [Sharma et al., 2018].

Following, we will present the obtained results of *an approach to the unified management of heterogeneous IoT environments*, within the context of ISABELA case study.

### 5.1.2 ISABELA Case Study

ISABELA was subjected to several tests involving the members of our research team and students in their academic context [Fernandes et al., 2019, 2020]. These trials allowed to validate the functional aspects of the platform, namely the integration of the various sources of sensed data. In Portugal, we had 10 university participants. This local aimed at validating the implementation and assess the effectiveness of our SA classification engine. The biggest trial was conducted from the 12$^{th}$ of May to the 12$^{th}$ of June 2018, with a total of 30 students from Escuela Politécnica Nacional - Ecuador. The trial in Ecuador had two objectives also aligned to broader expectations of the case study (see section 4.2). Hence, we collected both EBS, and SBS data and we could (A) correlate students location with their academic performance; and (B) correlate students sociability with academic performance.

#### 5.1.2.1 HBS-Related Results

We noticed that the amount of real data produced via OSNs was limited due to the reluctance in sharing such kind of information. It happened even with Twitter, regardless of tweets being public information. Also, the participants who had a Twitter account did not register it in the ISABELA mobile app parameters. This was the only way that we configured the app to fetch the tweet objects. Figure 5.2 displays samples of tweets objects in the STH.

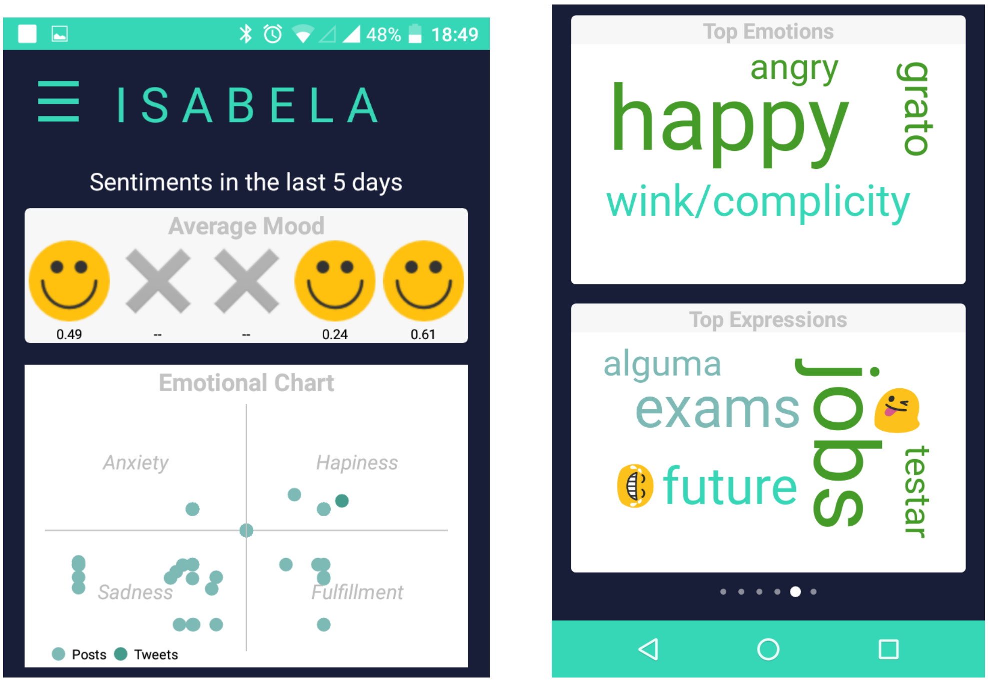

```
Fiware sth Query:
10.3.4.109:9007/STH/v1/contextEntities/type/mobile_hbs/id/twitter_XJ22P8qmbC01RnOrQK6K8Q/attrib
utes/post_message?aggrMethod=occur&aggrPeriod=second&dateFrom=2019-11-12T09:00:00&dateTo=2019-
11-13T20:00:00


Response:
{ "contextResponses": [{"contextElement": {"attributes": [{"name": "post_message",
…
"offset": 56, "samples": 15, "occur": {
               "back to the traffic": 3,
               "RT @VoceNaoSabiaQ: Acordar com o despertador faz mal a saúde, pois pode interro
               mper ciclos do sono, e gerar uma mudança muito abrupta na at…": 2,
               "RT @bancodeportugal: Emissões líquidas de títulos por residentes totalizaram €0
               ,5 mil milhões em setembro #estatísticas https:tcokIUjA1…": 1,
               "RT @MusicNewsRumor: @Camila_Cabello is rumored to release a new song titled #Cu
               rious, which is a collaboration featuring @LilNasX, on her…": 3,
               "RT @IEEEorg: Congratulations, Roel! https:tcoOdb0456cMH": 4,
               "RT @IEEEorg: #MIT’s Mini Cheetah #robots have advanced from doing back flips to
               playing soccer! Watch as these “virtually indestructible” #…": 2 } }… ],
                 "id": "twitter_XJ22P8qmbC01RnOrQK6K8Q",
                 "isPattern": false,
                 "type": "mobile_hbs"
             },
             "statusCode": { "code": "200", "reasonPhrase": "OK" } } ]}
```


Figure 5.2: Retrieved tweet objects saved in FIWARE

As for the scheduled entry into force of the General Data Protection Regulation (GDPR) [ENISA, 2017], OSNs operators started updating the usage conditions of their APIs. Thus, we had to submit validations reviews for ISABELA before Twitter and Facebook.

As reported in *Appendix B - Twitter GDPR Application Review*, the process proposed by Twitter has been quick and relatively simple to cope with. We had more difficulties with Facebook. Their review process included a detailed video screenshot of the whole application usage walk-through and the availability of the installation code for them to test the mobile application. The process took more than three round-trip revisions, and we have been refused to collect the posts for our purposes. We decided to prototype the integration of the OSN data only with Twitter. However, there are limitations in the use of standard its standard API. For instance, the limitation to 15 calls every 15 minutes[1]. In this context, a deployment of the case study for a massive usage would require an upgraded API.

The HBS-related results in Figure 5.3 include mainly data produced by the members of our research team and few number of students. Therefore, Figure 5.3 is to be interpreted as a typical data visualizations generated in ISABELA case study as the result of human-based sensing.




(a) Mood and Mapped emotions (b) Plain emotions and expressions

Figure 5.3: Sentiment Analysis visualizations

The emojis and emotions scatter plot in 5.3a are entirely processed in ISABELA's Android code. The weighted lists for the word clouds in 5.3b are processed in the SA engine. This hybrid approach (i.e., part of the treatment being done in ISABELA's Android code, while the SA engine processes other parts) was adopted to limit the energy consumption on the user's mobile device

[1] `https://developer.twitter.com/en/docs/basics/rate-limiting`, accessed on 2018-03-03

and to enable the immediate display of the Android activity. However, it is possible to have everything processed on either side.

The logic behind the average mood displayed in 5.3a is a computation of mean polarity values over the previous five days. Then, each daily polarity value is mapped to one of three emojis, depending on chosen intervals, that are displayed on the screen. The polarity values have been normalized to [-1,1] interval. Sad, neutral and happy emojis will be rendered in the screen for negative, neutral and positive polarity values, respectively. A cross sign corresponds to an absence of OSN objects timestamped for that day.

Concerning the scatter plot in the "Emotional Chart", we retrieve the scores for valence (horizontal axis) and arousal (vertical axis), then normalize them to [-4,4] interval. Finally, we plot the corresponding green spots in the Cartesian plane following the *Russell's simplified circumplex model*.

We chose not to update the *Word Cloud* entities every time a new message arrives. Instead, the *Build Word Cloud Object* class is triggered once a day, typically when less activity was produced from ISABELA users. This class processes the following tasks:

- Retrieve the last five days' historic messages, emotions and topics;
- Trigger both emotion and topic analysis based on historic messages;
- Compute the word cloud weighted list for "Top Emotions" and "Top Expressions";
- Compose the analysed OSN entity and send it to ORION.

Both LEED libraries and *Linguateca* dictionary are leveraged to create the "Top Emotions" dataset 5.3b. As for the "Top Expressions" dataset, it is fed by an array regrouping a weighted list of raw tokens and emojis from the OSNs objects. The set of results generated by a *topic modelling* or *opinion mining* engine can replace the "Top Expressions" results [Arora et al., 2018].

Figure 5.4 is the layout set, *in-lab*, for testing the functionalities of unified management in ISABELA. The tested cases covered a mobile scenario. We could change the "Sample Frequency" of an EBS, modify addressable parameters as for Software-based entities, and toggle the "Blocking Status" of an HBS.

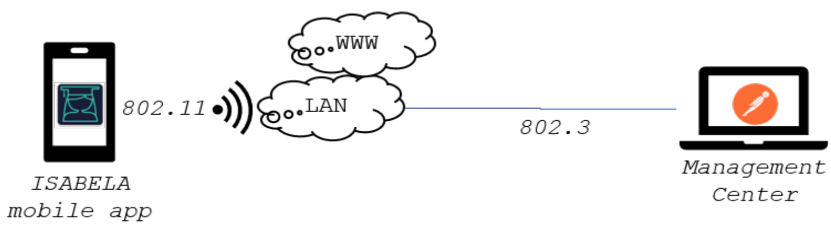



Figure 5.4: Scenario for unified management use-cases

#### 5.1.2.2 "Sample Frequency" Management

While we changed the "Sample Frequency" values, we registered the historic data STH-COMET. In section 4.2.3.2, we saw that the sensors data embed their collected timestamp. Hence, we could calculate the delta ($\vartriangle$) between the collected epoch timestamp for one sensor reading and its predecessor. The blue/white-dotted graph in Figure 5.5 represents these $\vartriangle$ values in seconds. We had also recorded the timestamps when a new "Sample Frequency" was set ISABELA mobile app. As a result, we could fill an dataset represented the dark-dotted graph in Figure 5.5.

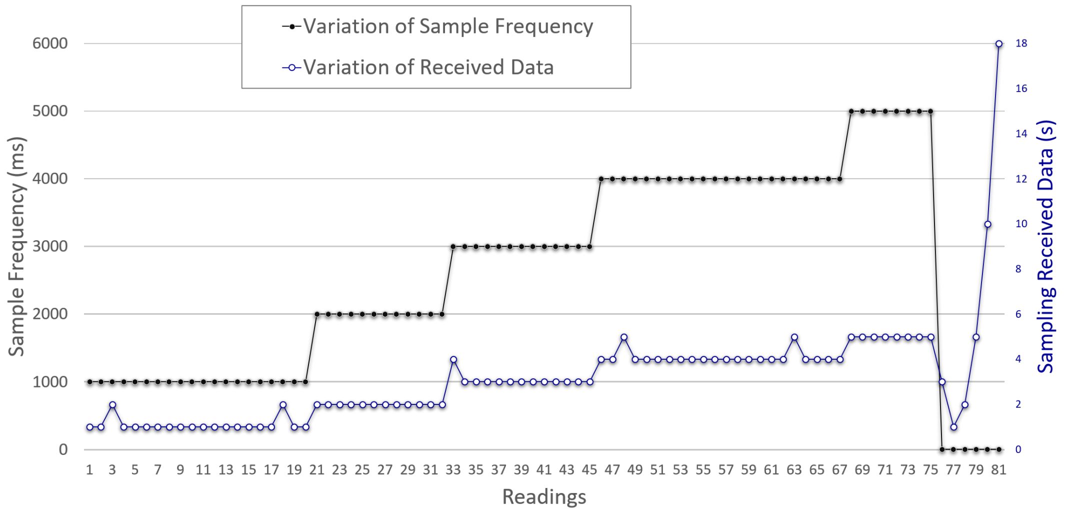



Figure 5.5: Results of the "Sample Frequency" management

#### 5.1.2.3 Addressable Parameters Management

We implemented the *addressable text* entity described in section 4.2.2.1. The resulted management functionality, depicted in Figure 5.6, shows that it can serve as an instant messaging system. For instance, teachers can send non-automatic and personalized recommendation to the students via the ISABELA platform. In the left-up side of Figure 5.6, we can see that the value for the "Set_parameters" is: "It works_yellow_advice". Such content is divided into three parts split by an underscore.

The first part is the *recommendation* content, while the second part sets the colour of the background in the notification icon. The third part is tagged *advice* that is leveraged to differentiate it from the "Blocking Status" notification type. We have configured the Android notification slider so that when the user clicks on it, it is directed to the chatbot from where he can reply. The reply content feeds the "reply_text" attribute. Hence, its content updates the corresponding value in the Broker. The content of the "reply_text" i.e., "Ola", appears both in the chatbot and in the ORION context broker.

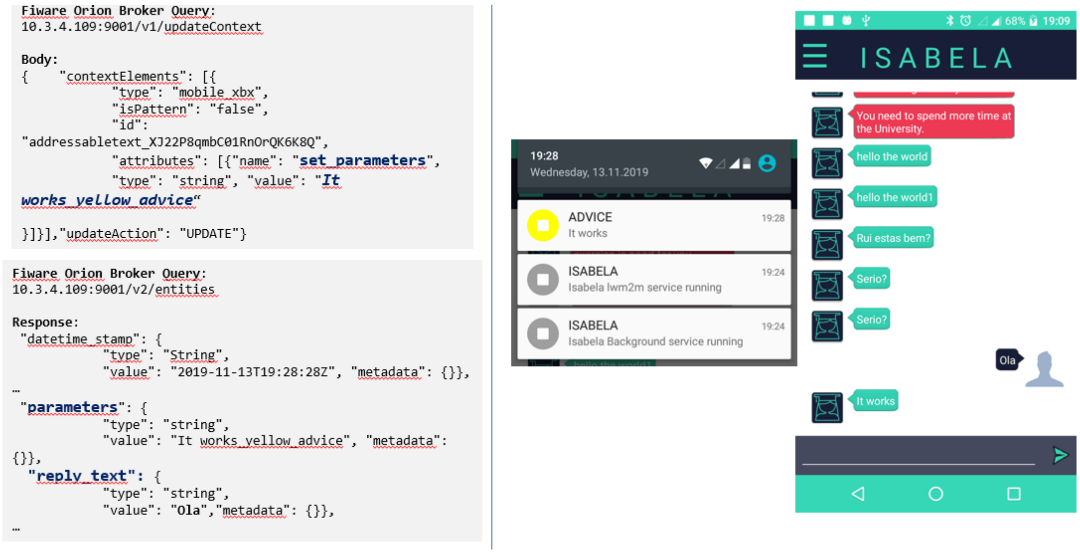



Figure 5.6: Addressable parameters in Software-based entities

### 5.1.2.4 Blocking Status Management

Even if Figure 5.7 displays a use-case of HBS, the blocking status works for any managed entity. By triggering the *set_blocking_status* attribute in the HBS managed entity, we change a boolean variable configured to run or not the Android service to retrieve the tweet objects. This is how we can enable or prevent a user from being a "Social Sensor". Whenever the *blocking_status* changes, the user receives a notification.

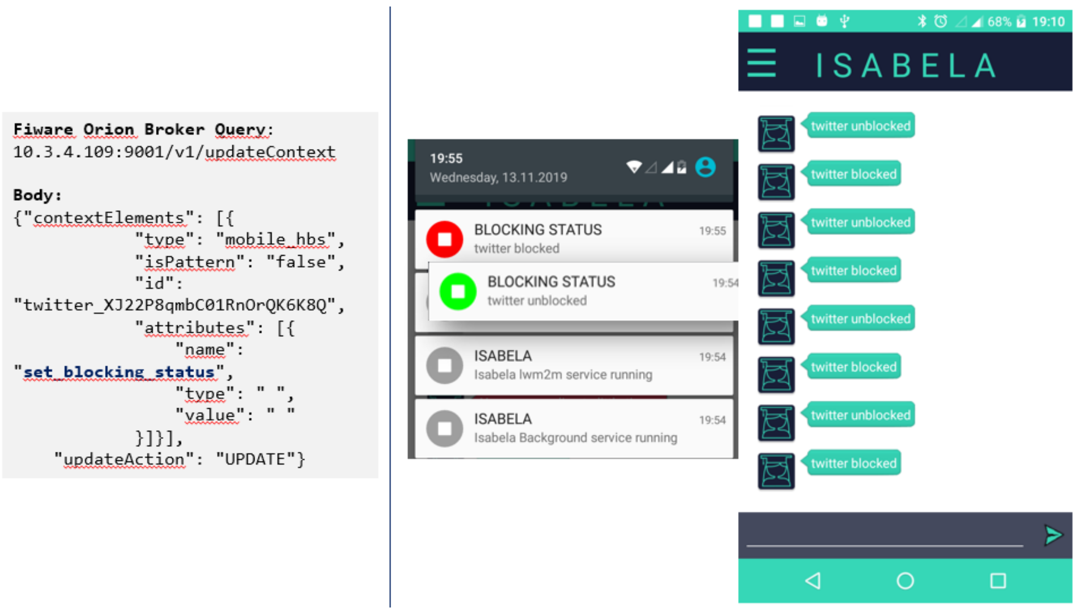



Figure 5.7: Blocking notifications

In this section (5.1), we showed that Human-based sensing was successfully integrated into the ISABELA system. Posts from the OSNs were collected and subsequently stored in the backend for further processing. This opens up a new field of action, in which all kinds of sensors  physical, software-based, or human-based  can be processed and managed using a unique framework. In the next section, we will focus on the performance aspects of the ISABELA case study.

## 5.2 Ultralight versus LwM2M

### 5.2.1 Used Bandwidth

It is essential to remind that for the Bandwidth Occupancy, we have considered all type of traffic in the network generated to send the 250 sensor readings. In Figure 5.8, each bar represents the sum of the captured frame lengths.

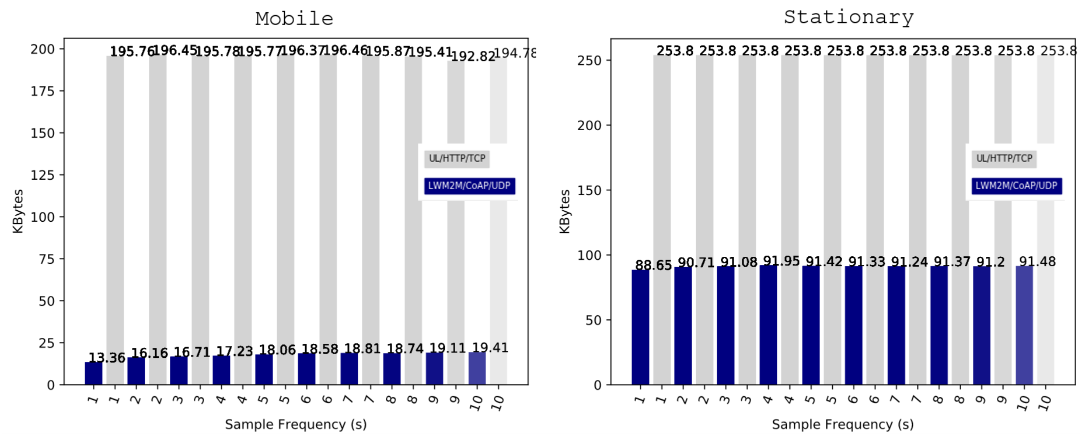



Figure 5.8: Occupied bandwidth

Two files from the "dataset" folder described in section 4.2.2.2, contain the lengths of the frame for each trial. They are, namely *"service"_outbound.csv* and *"service"_inbound.csv*. The number of frames in both documents are the same. That can be deduced as all the communications packets that left the client's interfaces (outbound traffic in Figure 4.7) have been received at the FIWARE-server network interface (inbound).

However, we remarked that sometimes the frames' lengths were not the same depending on the interface where they have been probed. To overcome such situations, we only consider the most significant length values. For instance, if the captured length value for the frame number X equalled to 52 and 60 Bytes, we discarded the former value and added latter value to the sum of the Used Bandwidth.

### 5.2.2 Delays

Both registration and connection processes in the mobile scenarios lasted around 6 to 44 ms on average. It corresponds to the flow (2) in Figure 3.9. This value is obtained by calculating the $\triangle$ of the epoch times between the following "String Values" columns in *.../10/idaslw_orion.csv* files:

- *"mobile_ebs, false,accelerometer_XJ22P8,sensor,string, ,APPEND"* && the corresponding
- *"mobile_ebs,false, accelerometer_XJ22P8,sensor,string„200,OK"* ||
- *"mobile_ebs,false, accelerometer_XJ22P8,APPEND"* &&
- the corresponding "*P1Y,5xx*"

The connection process in the stationary scenario lasted around 2 ms on average. This value has been obtained by calculating the $\triangle$ the epoch timestamps between the following "String Values" columns in *.../10/idaslw_orion.csv* files:

- *"box_ebs,false,loudness_7b7d,APPEND"* && the corresponding
- *"box_ebs,false,loudness_7b7d,200,OK"*

Overall, Figures 5.9 and 5.10 show that the delays in the network decrease as the "Sample Frequency" decrease. The traffic represented in their graphs (a) includes the 250 frames, while the traffic represented in graphs (b) includes frames generated in the network to send the 250 reading values: E.g. *provisioning, connection, acknowledgements, retransmissions, posts.*

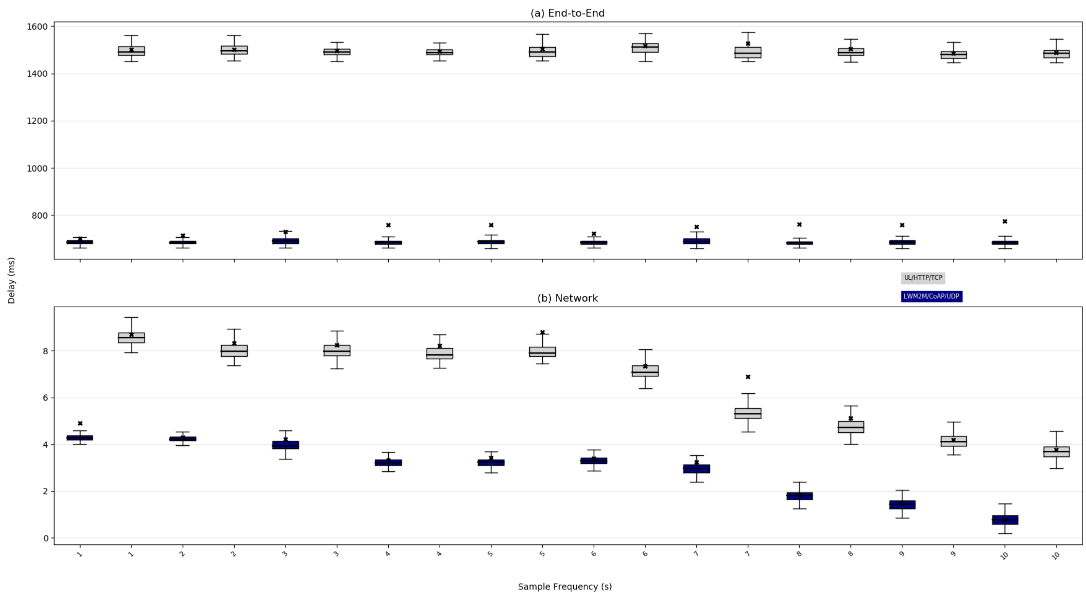



Figure 5.9: Delay distributions for mobile scenarios

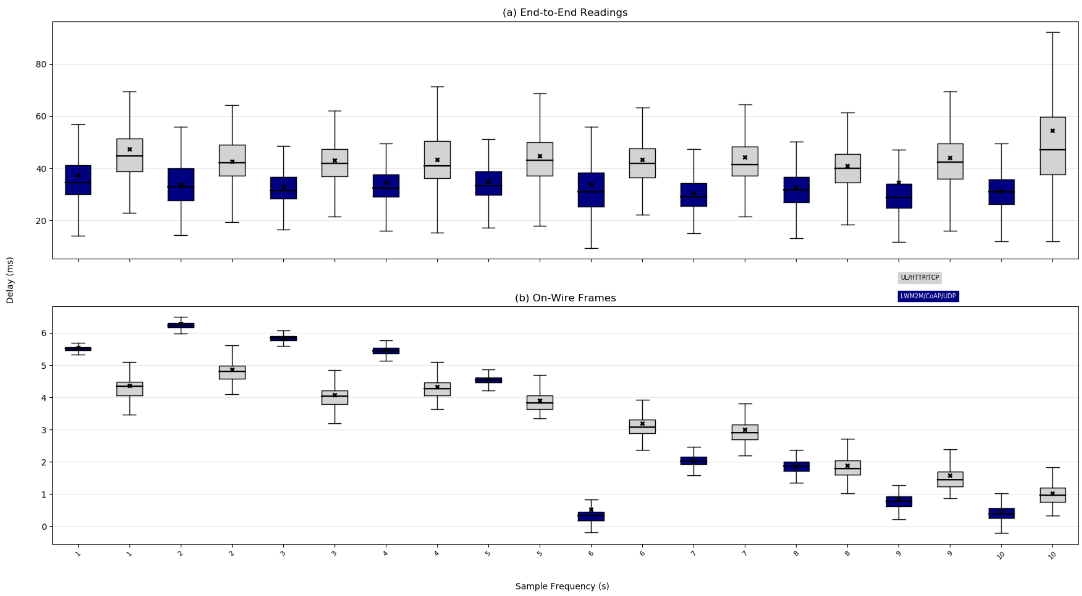



Figure 5.10: Delay distributions for stationary scenarios

The impact of the first interruption of test scenario 3 mentioned in Table 4.8[2], can be seen in Figure 5.10 (b). In the "Sample Frequency" of 6 seconds the network delay dropped to the lowest level as a consequence of most applications in the client were shut down prior to the interruption. In such a situation, all networks queues were basically empty. The impact of the second interruption can also be seen in Figure 5.10 (b), on the left-hand side, where LwM2M delays became higher than Ultralight delays for "Sample Frequencies" 5 to 1. We note that during this second interruption all applications kept running, including the Ultralight trials.

In the cases where data captures were carried on continuously for the ten "Sample Frequency" values, namely in test scenarios 1, 2, and 4, the network delays have a substantially linear increase with the decrease of the "Sample Frequency". It is the case for both stationary and mobile clients, and LwM2M and Ultralight (See both Figure 5.9 (b) and 5.10 (b)). Nevertheless, in the case of mobile clients, the delays for Ultralight are more or less twice the delays for LwM2M, as can be seen in Figure 5.9 (b).

Both Table 5.1 and 5.2 show the difference in delay when dispatching the packets via Ultralight and LwM2M services from FIWARE. The values in both Table 5.1 (a) and 5.2 (a), have been obtained by computing the $\Delta$ between the *.csv* files from the "dataset" described in section 4.2.2.2. For instance, the mean value of 3.78 for the "Sample Frequency" 1 has been obtained by calculating the first value in *../statio/networkdelay_ul.csv minus* the first value in *.../statio/networkdelay_lw.csv*. The negative results in Table 5.2 are aligned with the results on the left side of Figure 5.10 (b). A negative result corresponds to the situation when the value of Ultralight is smaller than LwM2M.

[2]stationary client && LwM2M protocol

Table 5.1: Delays delta between Ultralight and LwM2M. Mobile scenario

(a) Detailed

| Sample Frequency (s) | Network* | | | E2E* | | |
|---|---|---|---|---|---|---|
| | Mean | Median | Mode | Mean | Median | Mode |
| **1** | 3.78 | 4.29 | 4.13 | 799.38 | 807.14 | 830.77 |
| **2** | 4.01 | 3.74 | 3.56 | 789.08 | 813.66 | 793.34 |
| **3** | 4.03 | 4.05 | 3.98 | 767.35 | 799.2 | 791.44 |
| **4** | 4.9 | 4.62 | 4.48 | 734.88 | 803.88 | 793.34 |
| **5** | 5.38 | 4.66 | 4.78 | 745.71 | 804.3 | 795.29 |
| **6** | 3.94 | 3.78 | 3.58 | 796.76 | 828.04 | 790.25 |
| **7** | 3.65 | 2.33 | 2.3 | 774.88 | 798.68 | 791.82 |
| **8** | 3.27 | 2.91 | 2.96 | 743.5 | 806.95 | 788.49 |
| **9** | 2.7 | 2.66 | 2.71 | 726.64 | 795.62 | 763.41 |
| **10** | 2.99 | 2.89 | 3.09 | 713.23 | 802.74 | 787.89 |

(b) Summarized

| | | | | | | |
|---|---|---|---|---|---|---|
| **Mean** | 3.865 | 3.593 | 3.557 | 759.141 | 806.021 | 792.604 |
| **Median** | 3.86 | 3.76 | 3.57 | 756.53 | 804.09 | 791.63 |
| **Mode** | #NA | | | | | 793.34 |

**These values are expressed in ms.*

Table 5.2: Delays delta between Ultralight and LwM2M. Stationary scenario

(a) Detailed

| Sample Frequency (s) | Network* | | | E2E* | | |
|---|---|---|---|---|---|---|
| | Mean | Median | Mode | Mean | Median | Mode |
| **1** | -1.18 | -1.17 | -1.43 | 9.88 | 10.25 | 2.91 |
| **2** | -1.42 | -1.43 | -1.7 | 8.76 | 9.47 | 1.32 |
| **3** | -1.76 | -1.8 | -1.65 | 10.31 | 10.28 | 26.97 |
| **4** | -1.15 | -1.18 | -1.41 | 8.9 | 8.52 | 1.44 |
| **5** | -0.64 | -0.72 | -0.83 | 9.64 | 9.83 | 29.72 |
| **6** | 2.67 | 2.75 | 2.52 | 9.57 | 10.72 | 18.21 |
| **7** | 0.95 | 0.87 | 0.52 | 13.72 | 12.35 | 3.85 |
| **8** | 0.01 | -0.07 | -0.12 | 8.47 | 8.31 | 7.15 |
| **9** | 0.73 | 0.68 | 0.72 | 9.62 | 13.49 | 3.01 |
| **10** | 0.57 | 0.58 | 0.61 | 23.36 | 16.05 | 1.41 |

(b) Summarized

| | | | | | | |
|---|---|---|---|---|---|---|
| **Mean** | -0.122 | -0.149 | -0.277 | 11.223 | 10.927 | 9.599 |
| **Median** | -0.315 | -0.395 | -0.475 | 9.63 | 10.265 | 3.433 |
| **Mode** | #NA | | | | | |

**These values are expressed in ms.*

The values in both Table 5.1 (b) and 5.2 (b) result from the values in both Table 5.1 (a) and 5.2 (a), respectively. E.g., The first "**Mean**" value of 3.865 in Table 5.2 (b) is the arithmetic average of the "**Mean**" values above, i.e., from 3.78 to 2.99. The "#N/A - Non /Applicable" results for "**Mode**" metric have been obtained when any value in the column of Figure 5.9 (a) appears more frequently than others.

The results from Tables 5.1 (b) and 5.2 (b) show that for both stationary and mobile scenarios, packets will arrive faster to the ORION broker when dispatched via LwM2M than Ultralight service in FIWARE. Specifically, in the mobile scenario, the packets sent via LwM2M will be delayed 3.557 ms less, on average, by the network than the packet sent via LwM2M. However, the same packets will arrive 792.604 ms faster, on average, to ORION when dispatched via LwM2M than Ultralight.

Finally, in the stationary scenario, the packets sent via Ultralight will be delayed 0.277 ms less, on average, by the network than the packet sent via LwM2M. However, the same packets will arrive 9.60 ms faster, on average, to the ORION broker when dispatched via LwM2M than Ultralight.

Although the structure of a UDP frame is lighter than TCP (average measured around 8 and 300 Bytes, respectively), the network will not treat them equally. In fact, it will depend on the current conditions of the network and the QoS of the application processes transported by the Open Systems Interconnection Model (OSI) L4 protocols. TCP implements more robust mechanisms for traffic delivery, especially collision avoidance, which can explain the lower delays taken in the network as compared to UDP transported applications. Therefore, Figure 5.10 (b) confirms the results of the testbed illustrated in *Appendix C - Ultralight vs LwM2M. Comparative Performances in a Corporate LAN*, that were performed within a dense-traffic network.

### 5.2.3 Loss Ratio

The values in Figure 5.11 the ratios between the total number of lines identified as readings in both *.../idaslw_orion* and *.../idasul_orion* files and the known number of readings realized, i.e. 250. As described in section 4.2.2.2, the two types of files are the registries from Tskark, regarding the traffic received in the Broker(see Figure 4.7). Forty registries have been generated in total.

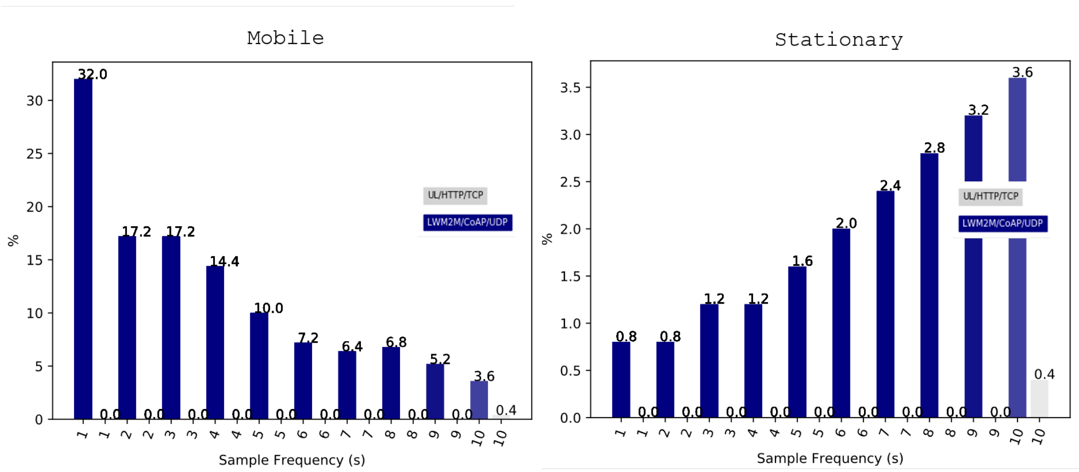



Figure 5.11: E2E packet loss

In Figure 5.11, we can remark that 1 out of 250 readings transmitted via Ultralight service weas not delivered for the "Sample Frequency" of 10 seconds. Such happened both for the stationary and mobile scenario. By observing the capture registries in the datasets, we could not identify a particular event around the lost readings. It is interesting to notice such an occurrence in the testbed. In fact, the transmission has been made over HTTP, which is designed with retransmission and control mechanisms capabilities.

In the mobile scenario from Figure 5.11, the "Packet Loss" starts at a high level (32 %) and keeps on decreasing with higher "Sample Frequency". This trend shows that our implementation of the LwM2M client, for Android OS by Leshan, struggled to process short data acquisition and sending periods. We remind that the leveraged LwM2M for the mobile scenario was not developed by the teams in charge of the FIWARE server. This probably explains a global lower performance as compared to stationary scenario results.

On the other hand, the loss Ratio pertained to the stationary scenario is globally low. It starts from 0.8% to 3.6%. We remind that the LwM2M client for the stationary scenario was developed by the teams in charge of the FIWARE server. At first sight, the low result could make us conclude of relatively excellent compatibility between the two components from the same development team. However, these results may also point to some LwM2M problems in the client version that we leveraged or at least, in the way we implemented the adaptations described in section 4.2.2.2. Specifically, we noticed that the Loss Ratio kept on growing with the "Sample Frequency". After careful analysis, such behaviour turned out to be related to the growing number of downtimes observed in the stationary client for the LwM2M service, as displayed in Figure 5.12. The results in Figure 5.12 show a pattern of an oscillating system that downgrades as the "Sample Frequencies" increase.

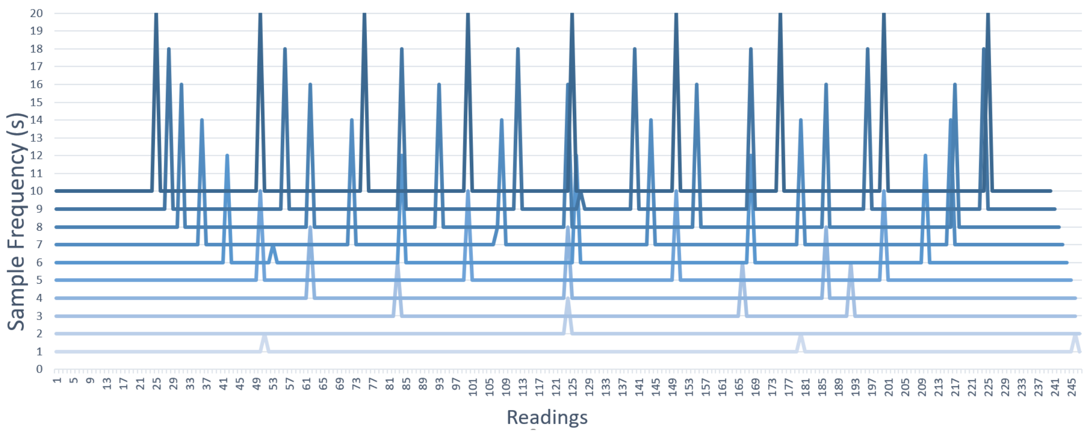



Figure 5.12: Regular activities and downtimes. Stationary scenario

Finally, in Figure 5.13 we can see regular downtimes of 1 second and two points where all the activities take more than 50 seconds to recover. It seems that around the 102$^{nd}$ and the 205$^{th}$ sensor reading shipment, the mobile hosted client could not receive other values during more than 50 seconds.

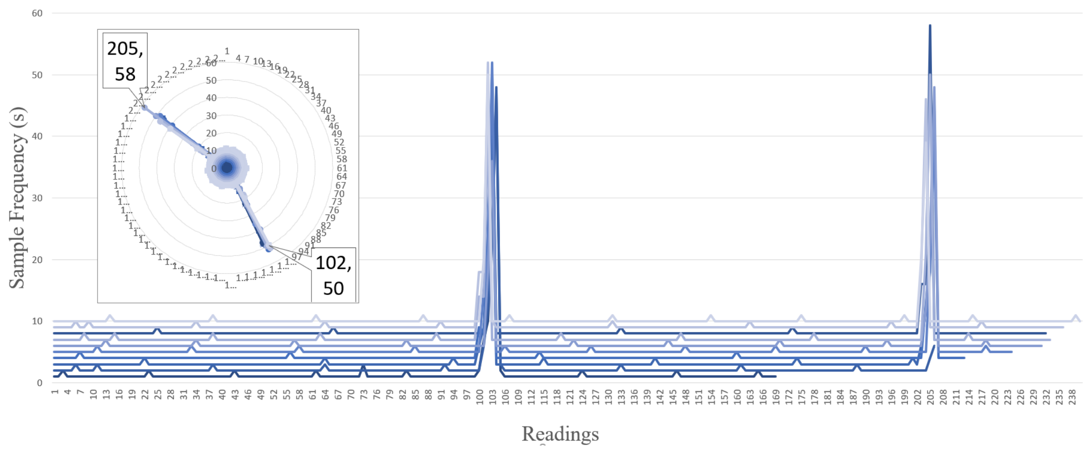



Figure 5.13: Regular activities and downtimes. Mobile scenario

## 5.2.4 Summarizing the Comparison

Both Table 5.3 and Figure 5.14 summarize the comparison between the two services. Each metric is represented as a distribution of Ultralight to LwM2M ratio values. E.g., the Used Bandwidth in the stationary scenarios has been obtained by the values in "networkoccupancy_ul.csv"[3] *divided* by the values in "networkoccupancy_lw.csv", for the corresponding "Sample Frequencies" values. We have represented the E2E Delivery Ratio instead of the Loss Ratio to avoid *null* numbers in the denominator of the ratio as such occurrence happened often to Ultralight scenarios. The former is the complement of the latter in such way that the former = 100% *minus* the latter.

Table 5.3: Dataset of the ratios Ultralight to LwM2M

| **"Sample Frequency" (s)** | **Stationary Scenario** | | | | **Mobile Scenario** | | | |
|---|---|---|---|---|---|---|---|---|
| | **Network** | | **E2E** | | **Network** | | **E2E** | |
| | **UBw** | **Delay*** | **Delay*** | **DR** | **UBw** | **Delay*** | **Delay*** | **DR** |
| **1** | 2.8 | 0.7 | 1.2 | 1 | 14.6 | 2 | 2.3 | 1.5 |
| **2** | 2.8 | 0.7 | 1.1 | 1 | 12.2 | 1.8 | 2.2 | 1.2 |
| **3** | 2.8 | 0.7 | 2.9 | 1 | 11-7 | 2 | 2.2 | 1.2 |
| **4** | 2.8 | 0.7 | 1.1 | 1 | 11.4 | 2.4 | 2.2 | 1.2 |
| **5** | 2.8 | 0.8 | 2.9 | 1 | 10.9 | 2.5 | 2.2 | 1.1 |
| **6** | 2.8 | 7.6 | 2.9 | 1 | 10.6 | 2.1 | 2.2 | 1.1 |
| **7** | 2.8 | 1.3 | 1.3 | 1 | 10.4 | 1.8 | 2.2 | 1.1 |
| **8** | 2.8 | 0.9 | 1.8 | 1 | 10.4 | 2.9 | 2.2 | 1.1 |
| **9** | 2.8 | 2 | 1.3 | 1 | 10.1 | 3 | 2.1 | 1.1 |
| **10** | 2.8 | 2.7 | 1.1 | 1 | 10 | 5.5 | 2.2 | 1 |
| **Mean** | 2.8 | 1.8 | 1.8 | 1 | 11.2 | 2.6 | 2.2 | 1.1 |
| **Median** | 2.8 | 1.1 | 1.6 | 1 | 10.7 | 2.5 | 2.2 | 1.1 |

*UBw = Used Bandwidth. DR = Delivery Ratio. *Mode Delays.*

[3] from the "dataset" folder described in section 4.2.2.2

If a value in Figure 5.14 is above 1, then the current metric is higher when the traffic is delivered through the Ultralight as compared to LwM2M shipment. Hence, Figure 5.14 shows that the Bandwidth, and both Network and E2E delays are, consistently, favourable to using LwM2M, as opposed to Ultralight, for both the stationary clients and mobile scenarios. For the mobile scenario's case, the bandwidth was 10 to over 14X higher for Ultralight, when compared to LwM2M, which is quite significant.

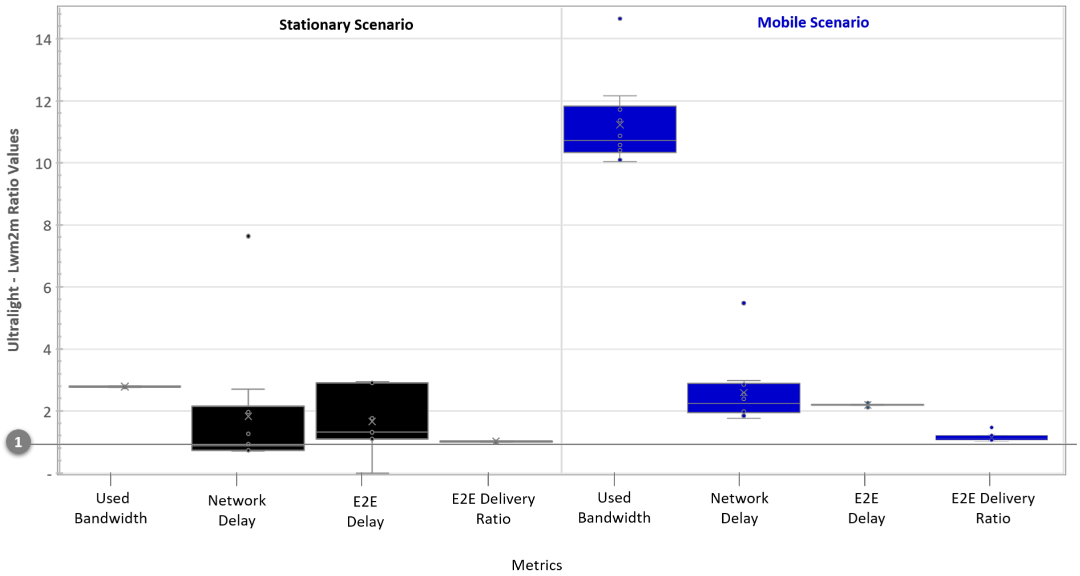



Figure 5.14: Ratios Ultralight to LwM2M.

Overall, the network delay decreases with higher "Sample Frequency", as can be seen in both Figure 5.9 (b) and 5.10 (b). On the other hand, the E2E delay does not change significantly with the "Sample Frequencies" (Figure 5.9 (a) and 5.10 (a)). Nevertheless, the E2E delay is generally higher for Ultralight, when compared to LwM2M and, in the case of the mobile scenario, the difference is quite significant.

Given the results in Figure 5.14, it is apparent that Ultralight requires more bandwidth and resources than LwM2M. Due to this, Ultralight's higher resource consumption and delay lead to a slightly better E2E delivery ratio as compared to LwM2M. Nevertheless, it is clear that this slightly better Ultralight performance comes at a high price. There is, thus, not a clear "winner". Both alternatives are usable in IoT management, and the choice for one or the other depends on different factors. For instance, Ultralight may be preferred if both reliability and easy integration with web applications are essential, and there are no significant limitations in terms of bandwidth. On the other hand, LwM2M may be the right choice if there is a need to minimize delays and required bandwidth.

## 5.3 Summary of the Chapter

In this chapter, we presented the results for the experiments that we framed in chapter 4. Besides the functionalities of unified management, we have also presented numerical results related to the performance of the leveraged management protocols.

We studied the aspects that can be considered for the choice of two management services offered by FIWARE. The obtained results point to mixed advantages/disadvantages of one protocol over the other. While LwM2M generally requires fewer resources and leads to less delay, Ultralight leads to better results in terms of packet loss.

The obtained results also point to some issues in the maturity/stability of the existing implementations for management services in FIWARE. These should be addressed in the future, including a deploying of the whole testbed in a corporate Local Area Network (LAN) to study the variations of the QoS metrics. It is also relevant to explore a variety of sensing scenarios, with and without interruption of the sensing activities. Such advanced experiments would enable to confirm or infirm the observed patterns in both Figure 5.12 and 5.13.

# Chapter 6

# Conclusions and Future Work

## Contents

**6.1 Synthesis** . . . . . . . . . . . . . . . . . . . . . . . . . . **98**

**6.2 Research Opportunities and Future Work** . . . . . . **100**

6.2.1 A Model for Privacy Management . . . . . . . . . . . 100

6.2.2 Opportunistic Selection of the IoTagents . . . . . . 100

In our research, we leveraged a middleware as the central component to provide the flexibility and scalability of the proposed architectural model. We have developed a set of test-case scenarios showing that a unified management of traditional and new "Devices" in the IoT is feasible from both functional and performance viewpoints. In this chapter, we will present the degree of success in achieving such a goal. We will also present the noted limitations and finally, the challenges that we have identified for future work.

## 6.1 Synthesis

We will synthesize the research leading to this dissertation through the following set of 8 questions:

(1) What was the main research problem addressed during PhD work?

We wanted to propose management solutions for *heterogeneous IoT environments*, i.e., beyond traditional entities. We wanted to address this problem by exploiting both a unique framework and widely adopted standards for data and device management.

(2) Why was it that important/relevant?

Because such an approach copes with the broad concept of heterogeneity in the IoT perception layer. The collection of contextualized data in the "new" IoT comprises not only electronic-based devices, but also virtual sensors (i.e., software agents that abstract one or more physical sensors), and even human sensors, such as human-originated data collected from Online Social Network.

(3) What were the research objectives?

- Carry on a review of the literature to identify the open issues within the scope of real and virtual devices integration in current IoT systems;
- Propose a generic high-level architecture to address one of the open issues.
- Validate the architecture with testbed and real scenario case studies.

(4) What did already exist in the literature?

Many proposals addressing IoT management issues can be found in the literature but, to the best of our knowledge, they are focused on well-established physical sensing. Well, one may want to change the algorithm of a virtual sensor on-the-fly or prevent a social sensor from feeding the IoT systems. While many such solutions are proprietary, some standards-based approaches can also be found, aiming at facilitating overall management of IoT systems.

(5) What methodology was proposed for the research problem?

We started our research by analysing the literature, and we studied the trends on it in a systematic way to be reproduced by our peers. Once we have identified the gaps in the literature, we filtered them to select the research opportunity that was most relevant for our team. With the research topic in mind, we designed the set of research objectives, which achievements have been based mainly on experimental approach with testbed PoC and real case scenarios.

As for the evaluation methodology, we used both qualitative and quantitative approaches. The former approach enabled us to know how well the solution we propose worked in terms of functionalities for the end-users. The latter approach enabled us to measure the performance of the frameworks that supported the solution we proposed.

(6) Which were the related investigation topics?

- Social sensing;
- Virtual sensing;
- IoT middleware;
- NLP for IoT systems;
- Data privacy in sensing-based mobile applications;
- 5G products and services.

(7) What results were obtained, and what are their limitations?

Besides the functionalities of the unified management architecture, we also have present numerical results related to the performance of the leveraged components and the measured QoS from the network.

We studied the aspects that can be considered for the choice of two management services offered by FIWARE. The obtained results point to mixed advantages/disadvantages of one protocol over the other. While LwM2M generally requires fewer resources and leads to less delay, Ultralight leads to better results in terms of packet loss. The obtained results also point to some issues in the maturity/stability of the existing implementations for management services in FIWARE.

Because of these issues in the stability of the existing implementations, it was not cautious adopting LwM2M in the management pipeline of the *Mobilizador 5G* project. With the 5GOpenclasses case study, we would like to validate the models developed during the thesis in a massive trial. We believe this would be the best way to test the limitations of our results.

(8) What were the most relevant aspects appreciated during the thesis project?

I appreciated intervening into different layers of IoT systems, i.e., perception, network, middleware and application. I also appreciated exploring different phases of building knowledge towards innovation and business in IT systems. It was interesting to have intervened in several axes of the life-cycle of innovative products and services, as referred to the TRL representation. It included research studies, designs of use-case scenarios, the branding aspects and business model, eventually.

Finally, I am happy for being the author of the names for ISABELA and 5GOpenclasses. Both branding proposals which were approved by our research team and the Leader of the *Mobilizador 5G* project.

## 6.2 Research Opportunities and Future Work

### 6.2.1 A Model for Privacy Management

With concerns of Privacy issues, the European Union created a set of new Data Protection laws to secure the privacy of the European citizens. In section 4.2.1.4, we described both administrative difficulties that we went through under the General Data Protection Regulation (GDPR). Both this situation and the reluctance of the end-users during the ISABELA trials, inspired us to explore ways that would enhance data privacy in mobile-based IoT systems.

A model for privacy management by design and by default goes in this direction. It would be interesting to study a model that allows users to share their data, without the need to install applications in smartphones. Let us imagine one scenario where a user is entering in the shopping and connect to the Free Wi-Fi available. In this context, it would be relevant to provide the *data owner* with a service on its mobile device that would enable him to share or not some sensors data to specific data consumers and at the desired time slot.

### 6.2.2 Opportunistic Selection of the IoTagents

In section 5.2.4, we saw that each management protocol has advantages and drawbacks. Therefore, we believe that it is relevant to design a solution that integrates an array of IoT agents in FIWARE. Then, one agent could be selected on-the-fly to serve the app/service, depending on both the expected performance and the current conditions offered by the network.

Besides the processing capabilities from the machine that will host the middleware, this opportunistic management solution will require challenging capabilities from the network itself. We believe that the capabilities and services such as those designed for 5G, i.e., uRLLC, Virtual Network Functions (VNF), and MEC, could be leveraged to overcome these challenges.

# Bibliography


Abbas, N., Zhang, Y., Taherkordi, A., and Skeie, T. (2018). Mobile Edge Computing: A Survey. *IEEE Internet of Things Journal*, 5(1):450–465.

Abdelwahab, S., Hamdaoui, B., Guizani, M., and Znati, T. (2016). Cloud of Things for Sensing-as-a-Service: Architecture, Algorithms, and Use Case. *IEEE Internet of Things Journal*, 3(6):1099–1112.

Abu Alsheikh, M., Niyato, D., Leong, D., Wang, P., and Han, Z. (2017). Privacy Management and Optimal Pricing in People-Centric Sensing. *IEEE Journal on Selected Areas in Communications*, 35(4):906–920.

Alsheikh, M. A., Jiao, Y., Niyato, D., Wang, P., Leong, D., and Han, Z. (2017). The Accuracy-Privacy Trade-off of Mobile Crowdsensing. *IEEE Communications Magazine*, 55(6):132–139.

Amin, M. T. A., Li, S., Rahman, M. R., Seetharamu, P. T., Wang, S., Abdelzaher, T., Gupta, I., Srivatsa, M., Ganti, R., Ahmed, R., and Le, H. (2015). Social Trove: A Self-Summarizing Storage Service for Social Sensing. In *2015 IEEE International Conference on Autonomic Computing*, pages 41–50. IEEE.

Armando, N., Raposo, D., Fernandes, M., Rodrigues, A., Silva, J. S., and Boavida, F. (2017). WSNs in FIWARE  Towards the Development of People-Centric Applications. In *Communications in Computer and Information Science*, volume 722, pages 445–456. Springer, Porto.

Armando, N., Rodrigues, A., Pereira, V., Silva, J. S., and Boavida, F. (2018). An Outlook on Physical and Virtual Sensors for a Socially Interactive Internet. *Sensors*, 18(8):2578.

Arora, S., Ge, R., Halpern, Y., Mimno, D., Moitra, A., Sontag, D., Wu, Y., and Zhu, M. (2018). Learning topic models  Provably and efficiently. *Communications of the ACM*, 61(4):85–93.

Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012). The social internet of things (SIoT) - When social networks meet the internet of things: Concept, architecture and network characterization. *Computer Networks*, 56(16):3594–3608.

Bachiller, R., Matthys, N., Del Cid, J., Joosen, W., Hughes, D., and Van Laerhoven, K. (2016). @migo: A comprehensive middleware solution for participatory sensing applications. *Proceedings - 2015 IEEE 14th International Symposium on Network Computing and Applications, NCA 2015*, pages 1–8.

Botts, Michael E. and Robin, Alexandre and Greenwood, Jim and Wesloh, David (2014). OGC SensorML: Model and XML Encoding Standard, http://www.opengeospatial.org/standards/sensorml.

Brogaard, J., Hendershott, T., and Riordan, R. (2017). High frequency trading and the 2008 short-sale ban. *Journal of Financial Economics*, 124(1):22–42.

Buttyan, L., Gessner, D., Hessler, A., and Langendoerfer, P. (2010). Application of wireless sensor networks in critical infrastructure protection: challenges and design options [Security and Privacy in Emerging Wireless Networks. *IEEE Wireless Communications*, 17(5):44–49.

Cardell-Oliver, R. and Sarkar, C. (2016). Robust sensor data collection over a long period using virtual sensing. In *ACM International Conference Proceeding Series*, pages 2–7, New York, New York, USA. ACM Press.

Cui, W., Wu, Y., Liu, S., Wei, F., Zhou, M. X., and Qu, H. (2010). Context preserving dynamic word cloud visualization. In *2010 IEEE Pacific Visualization Symposium (PacificVis)*, pages 121–128. IEEE.

Daniel, M., Neves, R. F., and Horta, N. (2017). Company event popularity for financial markets using Twitter and sentiment analysis. *Expert Systems with Applications*, 71:111–124.

Durisic, M., Tafa, Z., Dimic, G., and Milutinovic, V. (2012). A survey of military applications of wireless sensor networks. *Mediterranean Conference on Embedded Computing (MECO) 2012*, pages 196–199.

El Mhouti, A., Nasseh, A., and Erradi, M. (2016). Towards a collaborative e-learning platform based on a multi-agents system. In *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 511–516. IEEE.

Englert, F., Lieser, P., Alhamoud, A., Boehnstedt, D., and Steinmetz, R. (2015). Electricity-metering in a connected World: Virtual sensors for estimating the electricity consumption of IoT appliances. *Proceedings - 2015 International Conference on Future Internet of Things and Cloud, FiCloud 2015 and 2015 International Conference on Open and Big Data, OBD 2015*, pages 317–324.

ENISA (2017). Privacy and data protection in mobile applications, https://www.enisa.europa.eu/publications/privacy-and-data-protection-in-mobile-applications/at_download/fullReport.

Faculty Librarian, Mayer, P., Fettke, P., Stelzer, D., von Brocke, J., Simons, A., Niehaves, B., Riemer, K., Plattfaut, R., Cleven, A., Brocke, J. V., Reimer, K., Is, W., and Deposition, A. (2009). Analyzing the past to prepare for the future: writing a literature review. *Jane Webster, Richard T. Watson*, 48(4):2206–2217.

Fazio, M., Celesti, A., Marquez, F. G., Glikson, A., and Villari, M. (2015). Exploiting the FIWARE cloud platform to develop a remote patient monitoring system. In *2015 IEEE Symposium on Computers and Communication (ISCC)*, volume 2016-Febru, pages 264–270. IEEE.

Fei, Z., Li, B., Yang, S., Xing, C., Chen, H., and Hanzo, L. (2017). A Survey of Multi-Objective Optimization in Wireless Sensor Networks: Metrics, Algorithms, and Open Problems. *IEEE Communications Surveys & Tutorials*, 19(1):550–586.

Feldman, R. (2013). Techniques and applications for sentiment analysis. *Communications of the ACM*, 56(4):82.

Fernandes, J., Raposo, D., Armando, N., Sinche, S., Silva, J. S., Rodrigues, A., Pereira, V., and Boavida, F. (2019). An Integrated Approach to Human-in-the-Loop Systems and Online Social Sensing. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 478–483. IEEE.

Fernandes, J., Raposo, D., Armando, N., Sinche, S., Silva, J. S., Rodrigues, A., Pereira, V., Oliveira, H. G., Macedo, L., and Boavida, F. (2020). ISABELA A Socially-Aware Human-in-the-Loop Advisor System. *Online Social Networks and Media*, 16:100060.

Fonseca, J., Ferraz, C., and Gama, K. (2016). Doctoral symposium: A policy-based coordination architecture for distributed complex event processing in the internet of things. *DEBS 2016 - Proceedings of the 10th ACM International Conference on Distributed and Event-Based Systems*, pages 418–421.

Giorgetti, A., Member, S., Lucchi, M., Tavelli, E., Barla, M., Gigli, G., Casagli, N., Chiani, M., Dardari, D., and Member, S. (2016). A Robust Wireless Sensor Network for Landslide Risk Analysis : System Design , Deployment , and Field Testing. *IEEE Sensors Journal*, 16(16):6374–6386.

Giridhar, P., Wang, S., Abdelzaher, T., Amin, T. A., and Kaplan, L. (2017). Social Fusion: Integrating Twitter and Instagram for Event Monitoring. In *2017 IEEE International Conference on Autonomic Computing (ICAC)*, pages 1–10. IEEE.

Gupta, A. and Mukherjee, N. (2016). Poster. In *Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services Companion - MobiSys '16 Companion*, pages 28–28, New York, New York, USA. ACM Press.

Guth, J., Breitenbücher, U., Falkenthal, M., Fremantle, P., Kopp, O., Leymann, F., and Reinfurt, L. (2018). A Detailed Analysis of IoT Platform Architectures: Concepts, Similarities, and Differences. *Internet of Things book series (ITTCC)*, pages 81–101.

Haydarov, R. (2017). LwM2M Demo Client Android, https://github.com/ApplicationPlatformForIoT/LwM2MDemoClientAndroid, Accessed on 2019-03-14.

He, H., Maple, C., Watson, T., Tiwari, A., Mehnen, J., Jin, Y., and Gabrys, B. (2016). The security challenges in the IoT enabled cyber-physical systems and opportunities for evolutionary computing & other computational intelligence. In *2016 IEEE Congress on Evolutionary Computation (CEC)*, pages 1015–1021. IEEE.

Héder, M. (2017). From NASA to EU: The evolution of the TRL scale in Public Sector Innovation. *Innovation Journal*, 22(2):3.

Huang, C. and Wang, D. (2015). Link weight based truth discovery in social sensing. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks - IPSN '15*, pages 326–327, New York, New York, USA. ACM Press.

Hutto, Clayton J. and Gilbert, E. (2014). VADER: A Parsimonious Rule-Based Model for Sentiment Analysis of Social Media Text. In *Eighth International AAAI Conference on Weblogs and Social Media*, Michigan, USA. The AAAI Press.

I. F. Akyildiz, W. S., Sankarasubramaniam, Y., and E.Cayirci (2002). Wireless Sensor Networks : A Survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, 38:393–422.

ISO (2018). IEC JTC 1/SC 41 and Technologies Internet of Things and Related Internet of Things (IoT) Reference Architecture.

ITU-T Study Group 20 (2012). Overview of the Internet of things, ITU-T Y.2060, https://www.itu.int/rec/T-REC-Y.2060-201206-I.

ITU-T Study Group 20 (2016). Common requirements and capabilities of device management in the Internet of things, http://handle.itu.int/11.1002/1000/12780.

ITU-T Study Group 5 (2015). IMT Vision - "Framework and overall objectives of the future development of IMT for 2020 and beyond, https://www.itu.int/rec/R-REC-M.2083.

Jia, Y., Zhou, Z., Chen, F., Duan, P., Guo, Z., and Mumtaz, S. (2017). A Non-Intrusive Cyber Physical Social Sensing Solution to People Behavior Tracking: Mechanism, Prototype, and Field Experiments. *Sensors*, 17(1):143.

Khan, A. A., Rehmani, M. H., and Rachedi, A. (2017). Cognitive-Radio-Based Internet of Things: Applications, Architectures, Spectrum Related Functionalities, and Future Research Directions. *IEEE Wireless Communications*, 24(3):17–25.

Khan, I., Belqasmi, F., Glitho, R., Crespi, N., Morrow, M., and Polakos, P. (2016). Wireless sensor network virtualization: A survey. *IEEE Communications Surveys & Tutorials*, 18(1):553–576.

Khan, W. Z., Xiang, Y., Aalsalem, M. Y., and Arshad, Q. (2013). Mobile Phone Sensing Systems: A Survey. *IEEE Communications Surveys & Tutorials*, 15(1):402–427.

Knight, J. (2012). Approaches of Wireless Sensor Network Dependability Assessment. *Fundamentals of Dependable Computing for Software Engineers*, pages 355–391.

Krco, S., Pokric, B., and Carrez, F. (2014). Designing IoT architecture(s): A European perspective. In *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*, pages 79–84. IEEE Computer Society.

Kumar, A., Srivastava, V., Singh, M. K., and Hancke, G. P. (2015). Current Status of the IEEE 1451 Standard-Based Sensor Applications. *IEEE Sensors Journal*, 15(5):2505–2513.

Lee, C. H., Yang, H. C., and Lin, S. J. (2015). Incorporating big data and social sensors in a novel early warning system of dengue outbreaks. *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pages 1428–1433.

Lee, K. (2002). IEEE 1451 and Wireless Sensor Standard, http://www.nist.gov/el/isd/ieee/upload/Information-on-1451_1588-V36.pdf.

Li, Y., Pandis, I., and Guo, Y. (2016). Enabling Virtual Sensing as a Service. *Informatics*, 3(2):3.

Liberati, A., Altman, D. G., Tetzlaff, J., Mulrow, C., Gøtzsche, P. C., Ioannidis, J. P. A., Clarke, M., Devereaux, P. J., Kleijnen, J., and Moher, D. (2009). The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: explanation and elaboration. *Journal of clinical epidemiology*, 62(10):e1–34.

Liu, J., Shen, H., Narman, H. S., Chung, W., and Lin, Z. (2018). A Survey of Mobile Crowdsensing Techniques: A Critical Component for The Internet of Things. *ACM Transactions on Cyber-Physical Systems*, 2(3):18.

Lou, Y., Zhang, Y., Li, F., Qian, T., and Ji, D. (2020). Emoji-Based Sentiment Analysis Using Attention Networks. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 19(5):1–13.

Maia, B. and Santos, D. (2018). Language, emotion, and the emotions: The multidisciplinary and linguistic background. *Language and Linguistics Compass*, 12(6):e12280.

Marshall, J. and Wang, D. (2016). Mood-Sensitive Truth Discovery For Reliable Recommendation Systems in Social Sensing. *Proceedings of the 10th ACM Conference on Recommender Systems - RecSys '16*, pages 167–174.

Melcherts, H. E. (2017). The Internet of Everything and Beyond. In *Human Bond Communication*, pages 173–185. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Milton, R., Hay, D., Gray, S., Buyuklieva, B., and Hudson-Smith, A. (2018). Smart IoT and soft AI. In *IET Conference Publications*.

Miori, V. and Russo, D. (2017). Improving life quality for the elderly through the Social Internet of Things (SIoT). In *2017 Global Internet of Things Summit (GIoTS)*, pages 1–6. IEEE.

Nakashima, K., Yokoyama, M., Taniyama, Y., Yohshihisa, T., and Hara, T. (2016). S3 System: A System for Sharing Social Sensor Data and Analytical Programs. In *Proceedings of the 13th International Conference on Mobile and Ubiquitous Systems: Computing Networking and Services - MOBIQUITOUS 2016*, pages 147–152, New York, New York, USA. ACM Press.

National Instruments (2006). An Overview of IEEE 1451.4 Transducer Electronic Data Sheets, https://standards.ieee.org/develop/regauth/tut/teds.pdf http://standards.ieee.org/develop/regauth/tut/teds.pdf.

Nielsen, F. Å. (2011). A new ANEW: Evaluation of a word list for sentiment analysis in microblogs. In *Making Sense of Microposts (#MSM2011)*.

Nitti, M., Pilloni, V., Colistra, G., and Atzori, L. (2016). The Virtual Object as a Major Element of the Internet of Things: A Survey. *IEEE Communications Surveys & Tutorials*, 18(2):1228–1240.

O'donovan, T., Brown, J., Büsching, F., Cardoso, A., Cecílio, J., Furtado, P., Gil, P., Jugel, A., Pöttner, W.-B., Roedig, U., and Others (2013). The GINSENG system for wireless monitoring and control: Design and deployment experiences. *ACM Transactions on Sensor Networks (TOSN)*, 10(1):4.

OMA (2018). Lightweight Machine to Machine Technical Specification: Core, http://www.openmobilealliance.org/release/LightweightM2M/V1_1-20180710-A/OMA-TS-LightweightM2M_Core-V1_1-20180710-A.pdf.

OneM2M (2018). Draft Release 3 TS0001V3.11.0 Functional Architecture, p507. Technical report, OneM2M.

Oteafy, S. M. A. and Hassanein, H. (2016). Resilient IoT Architectures over Dynamic Sensor Networks with Adaptive Components. *IEEE Internet of Things Journal*, XX(X):1–1.

Posner, J., Russell, J. A., and Peterson, B. S. (2005). The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology*, 17(03):715–734.

Psomakelis, E., Aisopos, F., Litke, A., Tserpes, K., Kardara, M., and Campo, P. M. (2016). Big IoT and Social Networking Data for Smart Cities - Algorithmic Improvements on Big Data Analysis in the Context of RADICAL City Applications. In *Proceedings of the 6th International Conference on Cloud Computing and Services Science*, pages 396–405. SCITEPRESS - Science and and Technology Publications.

Raposo, D., Rodrigues, A., Silva, J. S., and Boavida, F. (2017). A Taxonomy of Faults for Wireless Sensor Networks. *Journal of Network and Systems Management*, 25(3):591–611.

Razafimandimby, C., Loscri, V., and Vegni, A. M. (2016). A neural network and IoT based scheme for performance assessment in internet of robotic things. *Proceedings - 2016 IEEE 1st International Conference on Internet-of-Things Design and Implementation, IoTDI 2016*, pages 241–246.

Razzaque, M. A., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2016). Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal*, 3(1):70–95.

Revision, K. (2020). About Kurento and WebRTC Kurento 6.10.0 documentation.

Rodrigues, D., Prada, M., Gaspar, R., Garrido, M. V., and Lopes, D. (2018). Lisbon Emoji and Emoticon Database (LEED): Norms for emoji and emoticons in seven evaluative dimensions. *Behavior Research Methods*, 50(1):392–405.

Rui, L., Zhang, P., Huang, H., and Qiu, X. (2016). A location-dependent task assignment mechanism in vehicular crowdsensing. *International Journal of Distributed Sensor Networks*, 12(9).

Saleem, Y., Crespi, N., Rehmani, M. H., Copeland, R., Hussein, D., and Bertin, E. (2017). Exploitation of social IoT for recommendation services. In *2016 IEEE 3rd World Forum on Internet of Things, WF-IoT 2016*, pages 359–364. IEEE.

Sarkar, C., Rao, V. S., Venkatesha Prasad, R., Das, S. N., Misra, S., and Vasilakos, A. (2016). VSF: An Energy-Efficient Sensing Framework Using Virtual Sensors. *IEEE Sensors Journal*, 16(12):5046–5059.

Sá Silva, Jorge , Silva, R. M., and Boavida, F. (2016). Redes de Sensores sem Fios. In FCA, editor, *IT*, page 8. FCA, Lisboa, fca edition.

Sharanya, S. and John, S. (2017). Comfort Sensor Using Fuzzy Logic and Arduino. In *Proceedings of 2nd International Conference on Intelligent Computing and Applications: ICICA 2015*, pages 155–160, Singapore. Springer, Singapore.

Sharma, R., Ribeiro, B., Pinto, A., Cardoso, F., Raposo, D., Armando, N., A, R., J, S. S., Gonçalo Oliveira, H., Macedo, L., and Boavida, F. (2018). Unveiling Markers of Stress Via Smartphone Usage. *in Proceedings of the 24th Portuguese Conference on Pattern Recognition, Coimbra, Portugal( RecPad2018 )*, pages 129–131.

Silva, M. J., Carvalho, P., and Sarmento, L. (2012). Building a Sentiment Lexicon for Social Judgement Mining. In *Building a Sentiment Lexicon for Social Judgement Mining*, pages 218–228. Springer, Berlin, Heidelberg.

Sinche, S., Hidalgo, P., Fernandes, J. M., Raposo, D., Silva, J. S., Rodrigues, A., Armando, N., and Boavida, F. (2020). Analysis of Student Academic Performance Using Human-in-the-Loop Cyber-Physical Systems. *Telecom 2020, Vol. 1, Pages 18-31*, 1(1):18–31.

Sinche, S., Raposo, D., Armando, N., Rodrigues, A., Boavida, F., Pereira, V., and Silva, J. S. (2019). A Survey of IoT Management Protocols and Frameworks. *IEEE Communications Surveys & Tutorials*, 22(2):1168 –1190.

Sinche, S., Silva, J. S., Raposo, D., Rodrigues, A., Pereira, V., and Boavida, F. (2018). Towards Effective IoT Management. In *Proceedings of IEEE Sensors*, volume 2018-Octob, pages 1–4. IEEE.

Soares, A. P., Comesaña, M., Pinheiro, A. P., Simões, A., and Frade, C. S. (2012). The adaptation of the Affective Norms for English Words (ANEW) for European Portuguese. *Behavior Research Methods*, 44(1):256–269.

Sousa Nunes, D. S., Zhang, P., and Sa Silva, J. (2015). A Survey on Human-in-the-Loop Applications Towards an Internet of All. *IEEE Communications Surveys & Tutorials*, 17(2):944–965.

Steve Liang (2014). Introduction to the OGC SensorThings API, https://www.geonovum.nl/sites/default/files/SensorThings-Tutorial-20140924 Masterclass Steve Liang.pdf, Accessed on 2017-11-17.

Sucasas, V., Mantas, G., Radwan, A., and Rodriguez, J. (2016). A Lightweight Privacy-Preserving OAuth2-Based Protocol for Smart City Mobile Apps. In *2016 IEEE Globecom Workshops (GC Wkshps)*, pages 1–6. IEEE.

Taneja, M. and Davy, A. (2017). Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm. In *2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM)*, pages 1222–1228, Lisbon. IEEE.

Tranfield, D., Denyer, D., and Smart, P. (2003). Towards a methodology for developing evidence-informed management knowledge by means of systematic review. *British Journal of Management*, 14:207–222.

Vallati, C., Virdis, A., Mingozzi, E., and Stea, G. (2016). Exploiting LTE D2D communications in M2M Fog platforms: Deployment and practical issues. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, pages 585–590.

Vmware (2019). Configure Time Synchronization Between Guest and Host Operating Systems, vSphere 5 Documentation Center, https://pubs.vmware.com/vsphere-50/index.jsp?topic=%2Fcom.vmware.vmtools.install.doc%2FGUID-C0D8326A-B6E7-4E61-8470-6C173FDDF656.html, Accessed on 2019-11-20.

Wang, D., Szymanski, B. K., Abdelzaher, T., Ji, H., and Kaplan, L. (2019). The age of social sensing. *Computer*, 52(1):36–45.

Wang, S., Su, L., Li, S., Hu, S., Amin, T., Wang, H., Yao, S., Kaplan, L., and Abdelzaher, T. (2015). Scalable social sensing of interdependent phenomena. In *Proceedings of the 14th International Conference on Information Processing in Sensor Networks - IPSN '15*, pages 202–213, New York, New York, USA. ACM Press.

Yang, S., Adeel, U., and McCann, J. (2015). Backpressure meets taxes: Faithful data collection in stochastic mobile phone sensing systems. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1490–1498. IEEE.

Yang, S., Guo, J., and Wei, R. (2017a). Semantic interoperability with heterogeneous information systems on the internet through automatic tabular document exchange. *Information Systems*, 69:195–217.

Yang, Y., Wu, L., Yin, G., Li, L., and Zhao, H. (2017b). A Survey on Security and Privacy Issues in Internet-of-Things. *IEEE Internet of Things Journal*, 4(5):1250–1258.

Yao, S., Hu, S., Li, S., Zhao, Y., Su, L., Kaplan, L., Yener, A., and Abdelzaher, T. (2016). On Source Dependency Models for Reliable Social Sensing: Algorithms and Fundamental Error Bounds. *Proceedings - International Conference on Distributed Computing Systems*, 2016-Augus:467–476.

Zach Shelby (2014). Tutorial on OMA Lightweight M2M, https://community.arm.com/iot/b/blog/posts/tutorial-on-oma-lightweight-m2m, Accessed on 2018-01-26.

# Appendix A - The Protocol of the Systematic Review

## Versioning

| Version | Modifications & Rationales | Date | Authors |
|---|---|---|---|
| 0.0.0 | Creation of the document | 15/03/2017 | Ngombo |
| 0.0.1 | Minor. Last phrase of The Backgrounds. Layout orientation. The title of the document. Appendix I & III | 21/03/2017 | Ngombo |
| 0.0.2 | *The following modifications were done following the tutors' recommendation to meet the objective of the SOCIALITE project better.*<br>• Update of the context.<br>• Update of the type of interventions.<br>• Update of the review title.<br>• Update of the search terms and strings.<br>• Update of the content in Sections VII, VIII, and X. | 28/03/2017 | Ngombo |
| 0.0.3 | *The following modifications were mostly done due to the high number of articles in the identification phase.*<br>• Update of the search terms and strings to refine the initial results.<br>• Update of the assessment quality strategy to better ensure the reproducibility of the review<br>• Update of the content of the Appendixes I and III as a consequence of the updating of the search strategy and assessment quality | 01/04/2017 | Ngombo |
| 0.0.4 | *The following modifications were mainly done due to the reproducibility concerns and the high number of materials in the screening phase (130).*<br>• Update of the contents of the appendixes (ex II and III)<br>• Adding the new Appendix IV<br>• Limitation of the review time scope to 2015-2017, except the recommended materials. In the Screening phase, we had 153 articles, and we wanted to have a more limited result to ensure a better Eligibility assessment. After the new timeframe, the # of selected materials for Eligibility dropped do 117. | 06/04/2017 | Ngombo |
| 0.0.5 | • Update of the content of the Appendixes II & III. The idea is to start extracting and writing the sum-up of each article as we proceed to the strategic reading. This critical raw information will then be interpreted to fill the historical review of each material in the Inclusion form.<br>• Adding of the new Appendix (Screening form)<br>• Updating of the numbering of the appendixes | 06/04/2017 | Ngombo |
| 0.0.6 | • Updating of the content of Appendix III. (To enable it to be more straight related to the keywords of the review; Thus, to the objective of the review)<br>• 117 materials were selected in Screeing V3. They will be strategically browsed before being chosen to be included in the review report. Since we have scheduled 15 days for the Eligibility exercise and in order to ensure a better focus on a short but relevant number of materials, we temporarily discard strategic reading and the filling of the Appendix III information for all materials with more than ten pages of content (Both the Appendix and bibliography pages do not count). Thus we had 85 materials left making an average of 6 materials per day to Full-text articles assessed for Eligibility. However, the materials with more than ten pages will be tagged "Standby", reserved as the primary source for any direct inclusion beyond the initial 68 materials. | 07/04/2017 | Ngombo |
| 0.0.7 | • Updating of appendix I : PRISMA Flow Statistics Results | 21/04/2017 | Ngombo |
| 0.0.8 | • Updating of the appendix V. Indeed, the dimension labelled *Cross, complementary, and contradictory approaches to other materials* are to be analyzed directly in the draft report once we will insert the narrative reviews in the report | 25/04/2017 | Ngombo |
| 0.0.9 | • Final update of the appendix list and contents | 01/05/2017 | Ngombo |

Table of Contents


I. Review title ..... 2
II. Reviewer ..... 2
III. Review question/objective ..... 2
IV. Background ..... 2
V. Inclusion criteria ..... 3
VI. Data sources ..... 3
VII. Search strategy ..... 4
VIII. Study quality assessment and risk of bias ..... 5
IX. Data synthesis ..... 6
X. Conflicts of interest ..... 6
XI. Acknowledgements ..... 6
XII. References ..... 6
XIII. Appendices ..... 6


# I. Review title

Models for the Integration of Virtual and Physical Sensors in the Internet of Things (IoT).

# II. Reviewer


- Ngombo Armando, Ph.D. Student[1,2]

[1] Centre for Informatics and Systems of the University of Coimbra - Portugal, narmando@dei.uc.pt
[2] Universidade Kimpa Vita / Escola Superior Politécnica do Uíge – Angola


The review will be conducted and reported as part of a PhD program, for both a doctoral subject assignment and the thesis proposal. In this respect, there will be only one reviewer. Both the Professor and the Tutors will instead give advises on the structural issues and ensure the general supervision of the review. We will also count on the opinions of the LCT team members during group and ad-hoc meetings. Since only one reviewer will conduct the assessment of the paper, there is a high risk of bias. The ideal situation would be the one recommended in the best practices [2]-[4], i.e. have at least two reviewers for the whole process. In the case of disagreement, a third party would judge the best option possible. Any assessment of the risk of bias will be reported in the data-extraction form or the synthesis of the review.

# III. Review question/objective

The purpose of the review is to analyze the current state-of-the-art on mechanisms to ensure both pervasive and ubiquitous sensing in the IoT.

More specifically, the qualitative objectives of the review are:

- Identify the mechanisms towards both pervasive and ubiquitous sensing in IoT.
- Leverage research opportunities to integrate the heterogeneity of the sensing mechanisms at the IoT Fog-Layer,
- Consider the cutting-edge normalized protocols both for communication and networking management.

# IV. Background

In the SOCIALITE project[1], the infrastructure design is based on a dynamic set of connected devices and systems, which requires establishing a uniform notion of abstraction throughout the whole architecture. The current architecture integrates a heterogeneity of devices for data acquisition. The sensor devices are mainly divided into two groups. The first group is set physical sensors embedded in mobile devices, i.e. smartphones accoupled to smartwatches. The second group is a set of Arduino's physical and stationary sensors. The heterogeneity of sensors is handled by an IoT platform middleware called FIWARE[2]. To enhance both ubiquitous and pervasive

[1] https://www.cisuc.uc.pt/projects/show/215
[2] fiware.org, accessed on 12/12/2016

sensing in the SOCIALITE architecture, the LCT team is considering new paradigms that leverage new models of data acquisition and network computing and normalized communications. For this reason, we are studying concepts of Virtual Sensing, Social Sensing, Virtual Sensors, Virtual Sensor Networks, and Fog-Computing. Using the normalized standards (6TiSCH, 6lo, and MIB+CoMI) is the approach in consideration to enable the communications and the management of the heterogeneity of sensors in the Fog network.

From the study, we made on Virtual Sensing, Social Sensing, and Virtual Sensors Networks, we think such concepts aim to guarantee the availability of the resources in the pathway to provide context-aware data. By resources, we mean communication, processing, and power. IoT platform middlewares seem to be the most advanced solutions to enable the availability of sensing resources. Any mechanism intended to fetch physical, biological, visual, and witnessed phenomenon can be integrated into such platforms. In fact, providing sensed data from different sources such as Social Sensors, Virtual Sensors, Mobile Sensors, and WSNs is the current trend to achieve ubiquitous and pervasive sensing towards the development of smart applications.

The present trend in literature does not seem to tackle the integration of the data acquisition mechanisms leveraging the Fog-computing. We think this approach could offer solutions to some of the open issues. Since ‘‘the sweet spot for the edge computing is thus in the infrastructure, where it can amplify the capabilities of proximate mobile devices and sensors’’; The idea is to integrate the heterogeneity of sensing mechanisms by developing an application that executes network management functions. For the communications issues, both Device-to-Device and Machine-to-Machine, we plan to leverage the normalized communications tailored for Low-energy and Lossy Networks such as 6LowPan, and the solutions of the works in progress, i.e. 6TiSCH, and 6lo. Concerning the management issues, we plan to leverage the normalized standards in progress for constrained devices and networks (MIB + CoMI).

## V. Inclusion criteria

### Types of interventions

This review will consider studies that propose, evaluate, investigate, and describe approaches in the IoT field, to create frameworks that enable to virtualize the heterogeneity of both Physical and Virtual Sensors.

### Types of outcomes

This review will consider studies that include the following outcome measures:

- Report concepts for data acquisition in IoT.
- Describe approaches to guarantee the availability of context-aware data.
- Report mechanisms to diversify the data acquisition process in IoT.

### Types of studies

The review will consider both experimental and conceptual study designs. Since we focus on describing the State-of-the-art around concepts, the review will privilege qualitative studies and textual papers, at the expense of quantitative ones.

Furthermore:

- Sources are to be considered when dated from 2013 to the search day.
- Published and unpublished studies are to be considered, with priority to the former.
- Both electronic and paper-based sources are to be considered, with priority to the former.
- Advice from experts is to be considered.
- Only open access materials will be considered.
- Only materials in English, Portuguese, and French will be considered.

## VI. Data sources

Besides the advice from experts, our data sources will include:

### Websites

Tarek Abdelzaher works, IETF and CoRE working groups

### Databases

IEEExplore, ACM Digital Library

**Hub**

Google Scholar via Harzing's Publish or Perish 5.

We will use the B-on[3] signature of the University of Coimbra to browse the online sources.

Following advice from the LCT team, we will focus on the most relevant outcomes from last meetings in the fields of IoT, i.e. IoTDI[4], and SocialSens[5]. Tarek Abdelzaher is one of the leading researchers in the field and organizing committee member in the last meetings. Thus, an initial limited search will be conducted on his works as listed on his page[6]. Then, we will do the searches in the database where the proceedings of the meetings above are published[7][8], namely IEEEXplore[9] and ACM Digital Library[10]. The other databases, namely IETF[11], and CoRE working Group[12] have been recommended both by the member of the LCT team and by the colleagues of the Doctoral program. Both IETF and CoRE working Group websites are exclusively for the search of normalized communication and standardized network management, respectively.

Complementary search to fetch additional material will be carried on using Google Scholar. Thus, we will use "search strings" as a priority. We used *Harzing's Publish or Perish tool*[13] to refine the searches and manage the results in tables. When fetching articles, sometimes they do not appear at first places in some databases, e.g. ACM DL. Leveraging its complementary effectiveness for these cases, we will use Google Scholar as the entering point to fetch the integral content of the materials; sometimes peer-to-peer shared in open access.

We are aware of the risk of bias in limiting the data sources, but we think the one we selected covered relevant materials in the literature, concerning the outcome of our study.

To all the databases, the default sorting will be done by the relevance of the results.

## VII. Search strategy

**Primary search**

Because the concepts are the focus in our review question, the primary search terms to be used will be[14]:

- "Virtual Sensor"
- "Virtual Sensing"
- "Social Sensor"
- "Social Sensing"
- "Fog Computing."
- "Virtual Sensor Networks"

**Secondary search terms**

Since the concepts are meant to serve as tools in our study, the secondary search terms to be used will be[15]

- "6TiSCH"
- "6lo"
- "CoMI"
- "MIB"

---

[3] http://www.b-on.pt/

[4] http://conferences.computer.org/IoTDI/

[5] http://www3.nd.edu/~dwang5/SocialSens2017/call.html

[6] http://web.engr.illinois.edu/~zaher/publications.html/

[7] http://www3.nd.edu/~dwang5/SocialSens2017/submission.html

[8] http://sn.committees.comsoc.org/journal-conference-publications/the-2nd-ieee-international-conference-on-internet-of-things-design-and-implementation-iotdi-2017/

[9] http://ieeexplore.ieee.org/search/advsearch.jsp : Metadata only (To make sure the focus of the articles match with our review question), all results (because not accurate. Fails to fetch some papers with no restriction to accessing = Open padlock), 2013-present.

[10] http://dl.acm.org/advsearch.cfm?coll=DL&dl=ACM&CFID=918844839&CFTOKEN=79797924: The ACM Guide to Computing Literature, where title, Abstract and author keyword matches all the terms. Publication year 2013-2017.

[11] https://datatracker.ietf.org/

[12] https://core-wg.github.io/comi/

[13] All the words, Years 2013-2017

[14] http://www.harzing.com/resources/publish-or-perish/windows: In the databases, the search terms are to be launched in quotations i.e. ""

[15] In the databases, the search terms are to be launched in quotations i.e. ""

### Search strings / Booleans

The search strings to be used will be the following. They are meant to target as relevant as possible the studies tackling the objectives of our review.

- "Internet of Things" & "Virtual Sensor Networks"
- "Internet of Things" & "Virtual Sensor"
- "Internet of Things" & "Virtual Sensing"
- "Internet of Things" & "Social Sensor"
- "Internet of Things" & "Social Sensing"
- "Internet of Things" & "Fog computing"
- "Internet of Things" & "Virtual Sensor Networks" & "Fog computing"
- "Internet of Things" & "Virtual Sensor" & "Fog computing"
- "Internet of Things" & "Virtual Sensing" & "Fog computing"
- "Internet of Things" & "Social Sensor" & "Fog computing"
- "Internet of Things" & "Social Sensing" & "Fog computing"
- "Internet of Things" & "Social Sensing" & "Virtual Sensing" & Fog computing
- "Internet of Things" & "Social Sensor" & "Virtual Sensor" & "Fog computing"
- "Internet of Things" & "Virtual Sensor" & "Physical Sensor" & "Fog computing"
- "Internet of Things" & "Virtual Sensing" & "Physical Sensor" & "Fog computing"
- "Internet of Things" & "Social Sensor" & "Physical Sensor" & "Fog computing"
- "Internet of Things" & "Social Sensing" & "Physical Sensor" & "Fog computing"

## VIII. Study quality assessment and risk of bias

To summarize the study selection processes, we will use a PRISMA-based flow diagram.

### Identification

The Identification phase will consist of downloading all the citations displayed as search results from the databases. In general, the databases enable to export the results in many formats. We will be working with the CSV format since our Forms are edited in Excel spreadsheets. There will be no assessment in for the Identification phase since it will be the result of the strategy applied in the previous section. For the identification process, we will use the computer-based data-extraction in Appendix I.

### Screening

For the Screening Phase 1, we will employ a form in Appendix II. We will start by removing the instantiations of the materials according to the identified titles. We assume the risk of bias, but we trust in a double-criteria since there may be different ways they are written. Then, the inclusion criteria will be applied to the identified materials in addition to some preliminary assessment to validate the titles.

For the Screening phase 2, if more than ten results are identified for one search keyword / String with one source, only the ten most relevant will be considered. The sorting assures the relevance of the source. For the Website, we will sort by year and considered the top ten most recent[16].

Finally, in Screening phase 3, we will assess the articles approved in phase 2. Here, the assessment will be undertaken analyzing the relevance of both the title and abstract of the articles according to our search objectives.

### Eligibility

For the Eligibility, we will download all the materials selected in Screening 3 using first and foremost Google Scholar as querying entrance to the databases[17]. Then, we will be conducting a strategic reading both to underline and extract relevant information to fill the cells of Appendix III. The reviewer can carry on accurate reading according to the interest the material presents for the outcome of the review. The reviewer will consider all the ideas in the papers that can contribute to the results of the review. The critical analysis during the strategic or the accurate reading will be based on the reviewer's experience, who will not focus on assessing the methodological quality of the published materials. Indeed, the current review is concept-centric, i.e. the concepts will determine the organizing framework of the review reference [4]. Thus, for the published resources, we will assume that the Editors cover both the methodological and the overall quality. Finally, to reduce both human error and bias, we will employ the form in Appendix III.

---

[16] Spreadsheet, MS Excel 365 ProPlus version 1702 (Build 7870.2024)

[17] With the title & sometimes the Authors, the publisher, in case there is an ambiguity in the search results by G. Scholar

#### Included

The materials considered relevant in the Eligibility phase will be included in the report. To summarize the findings, we will use the form in Appendix IV. For the citation management; we will first load all the included material into the Mendeley workspace[18]. Then, if the report of the review is edited in MS Word, we will use the MS plugin. In case the report is edited in LaTex, we will first export the Mendeley files to a BibTeX[19] file.

It is advised to take notes on the outlines of the review report while carrying on the strategic reading. It is here where we begin to have a systemic view of the literature. It means, we begin to understand the vacuum in the literature that can be covered by further researches aligned to the outcome of the review.

## IX. Data synthesis

The results of the study will be pooled using the data-extraction form in Appendix IV. We will involve the synthesis of conclusions to generate a set of statements that represent the aggregation. We will assemble the results rated according to their quality, and categorizing these results by the similarity in meaning. These categories will then be subjected to a meta-synthesis to produce a single comprehensive set of synthesized findings. "Meta-synthesis provides a means of considering: 'all significant similarities and differences in language, concepts, images, and other ideas around a target experience.' (Sandelowski, Docherty and Emden, 1997, p.669)" [3]. Where textual pooling is not possible the findings will be presented in a narrative form that is, we will identify what has been written in the paper along with and the outcomes of the review [3].

## X. Conflicts of interest

No conflicts of interest have been identified.

## XI. Acknowledgements

The work presented in this paper was partly carried out in the scope of the SOCIALITE Project (PTDC/EEI-SCR/2072/2014), co-financed by COMPETE 2020, Portugal 2020 - Operational Program for Competitiveness and Internationalization (POCI), European Union's ERDF (European Regional Development Fund), and the Portuguese Foundation for Science and Technology (FCT).

## XII. References

1. JBI Library of Systematic Reviews. Available: http://joannabriggs.org/assets/docs/jbc/operations/prot-sr-bpis-tech-templates/jbi-sr-protocol-template.docx. Accessed 20 March 2017
2. The PRISMA Statement for Reporting Systematic Reviews and Meta-Analyses of Studies That Evaluate Health Care Interventions: Explanation and Elaboration. http://journals.plos.org/plosmedicine/article?id=10.1371/journal.pmed.1000100. Accessed 21 March 2017.
3. D. Tranfield, D. Denyer, and P. Smart, "Towards a methodology for developing evidence-informed management knowledge by means of systematic review ," *Br. J. Manag.*, vol. 14, pp. 207–222, 2003.
4. Faculty Librarian et al., "Analyzing the past to prepare for the future: writing a literature review," Jane Webster, Richard T. Watson, vol. 48, no. 4, pp. 2206–2217, 2009.

## XIII. Appendices

The details of the appendixes are in the data-extraction spreadsheets available at the following link: https://goo.gl/W5ATRz

[18] https://www.mendeley.com/

[19] http://www.bibtex.org/

*Appendix I: Identification Form*

| # | Date of Search / retrieval | Search keywords | Search Booleans | Title | Authors | Type of source | Type of publishing | Publisher | Data source | Year of publication (YoP) | Abstract | Identification Notes |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 001 | 01-04-17 | "Social Sensing" | -- | Sensing | ran Zhao, M | Paper | Conference | on Distribut | ois.edu/~zah | 2017 | | |
| 002 | 01-04-17 | "Social Sensing" | -- | by Inte | iridhar, Lanc | Paper | Workshop | p on Social | ois.edu/~zah | 2017 | | |

*Appendix II: Screening Assessment Form*

| **Duplication / Instantiated?** *(03-04-2017)* | **Is it an Duplication / Instantiation?** | **Valid Title?** *(if not excluded instantiation* | **Rationale if "No"** | **Inclusion Screening V1** *(03-04-2017)* | **Inclusion Screening V2?** *(03-04-2017)* | **Inclusion Screening V3?** *(04, 05, 07-04-2017* | **Rationale for Exclusion V3** |
|---|---|---|---|---|---|---|---|
| **Yes** | **No** | | | **Yes** | **Yes** | **Yes** | |
| | **No** | | | **Yes** | **Yes** | **Yes** | |

*Appendix III: Eligibility Assessment Form*

| **Eligible?** *(07 to 21-04-2017)* | **Rationale if No** | **Type of study** *(If elected)* | **Fog or related topic Tackled?** *(If elected)* | **"Social Sensing" Tackled?** *(If elected)* | **"Virtual Sensing" Tackled?** *(If elected)* | **Virtual Sensor Networking Tackled?** *(If elected)* | **D2D communication tackled?** *(If elected)* | **IoT open standards Tackled?** *(If elected)* | **Key information** *(If elected)* | **Challendges & Open issues** *(If elected)* | **Additional notes** *(If elected)* |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Yes** | | Conceptual , Experimental | Yes | Yes | Yes | No | | No | participatory | g term, we will also explore how clou | |
| No | note submitted to CCR. It | | | | | | | | | | |

*Appendix IV: Inclusion Sum up Form*

| **Narrative review / Sum up of the Article** *(22-29 /04/2017)* | **Additional notes** *(22-29 /04/2017)* |
|---|---|
| Participatory sensing is a way of including | 0 |
| survey concerning the 5G networks, the | ing sevices (Broker and the |

# Appendix B - Twitter GDPR Application Review

**Request** – =09/11/2018

1) The core use case, intent, or business purpose for your use of the Twitter APIs

Our research team works on people-centric applications [a]. We are currently leveraging online social media as an enhanced communication channel for contextualised data. This approach is known as Social Sensing in the literature [b].

2) If you intend to analyze Tweets, Twitter users, or their content, share details about the analyses you plan to conduct and the methods or techniques

On Twitter, we work with the text available in "What´s happening space", i.e. tweets and retweets. With such text data set and applying supervisioned Natural Language Processing techniques, we try to produce insights about the user's mood and the events in their surroundings [c]. Combined with data from traditional electronic sensors, social sensors will enable us to produce richer contextualized data and thus the possibility of having a more accurate actuation in IoT.

3) If your use involves Tweeting, Retweeting, or liking content, share how you will interact with Twitter users or their content

Not applicable, because our application is a read-only one.

4) If you'll display Twitter content off of Twitter, explain how and where Tweets and Twitter content will be displayed to users of your product or service, including whether Tweets and Twitter content will be displayed at row level or aggregated

Individual tweets and content will never be displayed. Instead, it is their aggregated results that will feed three kinds of data visualisation: emojis, Russell´s circumplex scatterplot, and emotions and topics word clouds. The results of the aggregated data will be displayed to the users via a mobile application they will install, in compliance with GDPR laws. Secondly via a desktop management interface, e.g. for teachers, in academic use cases or care staff in AAL use case. The tweets will be stored in a secured database leveraging FIWARE middleware [c]. FIWARE will interact with the Analytics module on one side and the two displaying applications on the other side. The objects created in databases linked to FIWARE will not store any private information about the users. The relationship between aggregated data and users will be made upon fictitious ID.

[*a] https://www.cisuc.uc.pt/projects/show/215*

*[b] https://arxiv.org/ftp/arxiv/papers/1801/1801.09116.pdf*

*[c] N. Armando et al., "An Outlook on Physical and Virtual Sensors for a Socially Interactive Internet," Sensors, vol. 18, no. 8, p. 2578, Aug. 2018*

*[d] Fazio, A et al., "Exploiting the FIWARE cloud platform to develop a remote patient monitoring system," Proc. - IEEE Symp. Comput. Commun., vol. 2016–Febru, pp. 264–270, 2016*

**Response**

*Twitter developer account application [ ref:_00DA0K0A8._5004A1X8qiy:ref ]*
*Your Twitter developer account application has been approved!*
*Thanks for applying for access. We've completed our review of your application, and are excited to share that your request has been approved.*
*Sign in to your developer account to get started.*
*Thanks for building on Twitter!*

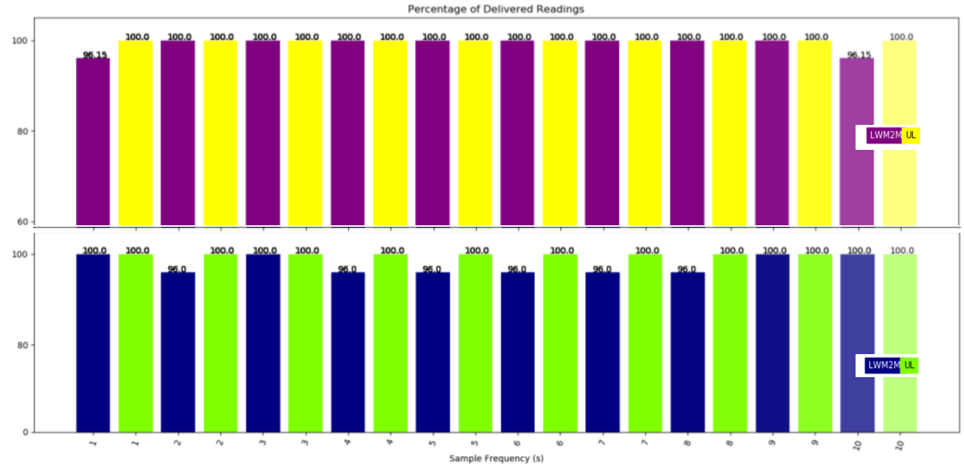
*https://developer.twitter.com/en/developer-terms/agreement-and-policy*

# Appendix C - Ultralight vs LwM2M. Comparative Performances in a Corporate LAN

## Delivery Rate

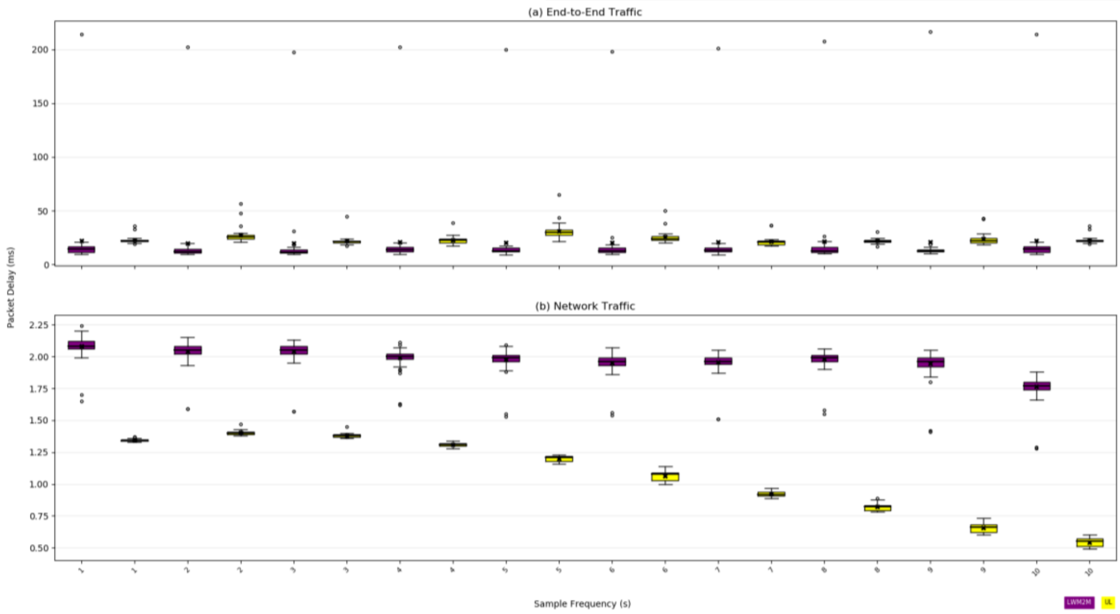
BOX

MOBILE



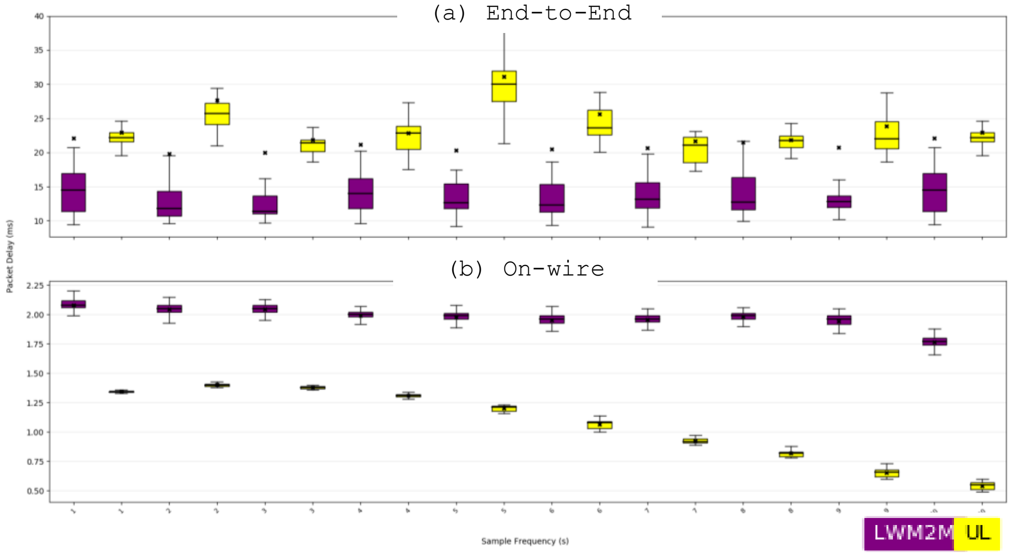

LW presented the worst performance. In 20% of the experiments with mobile-hosted clients, 1 out of the **26** Readings was not received in ORION; while in 60% of the experiments with mobile-hosted clients, the packet loss was of 1 out of the **25** Readings

## Delay, Box , Posts Only



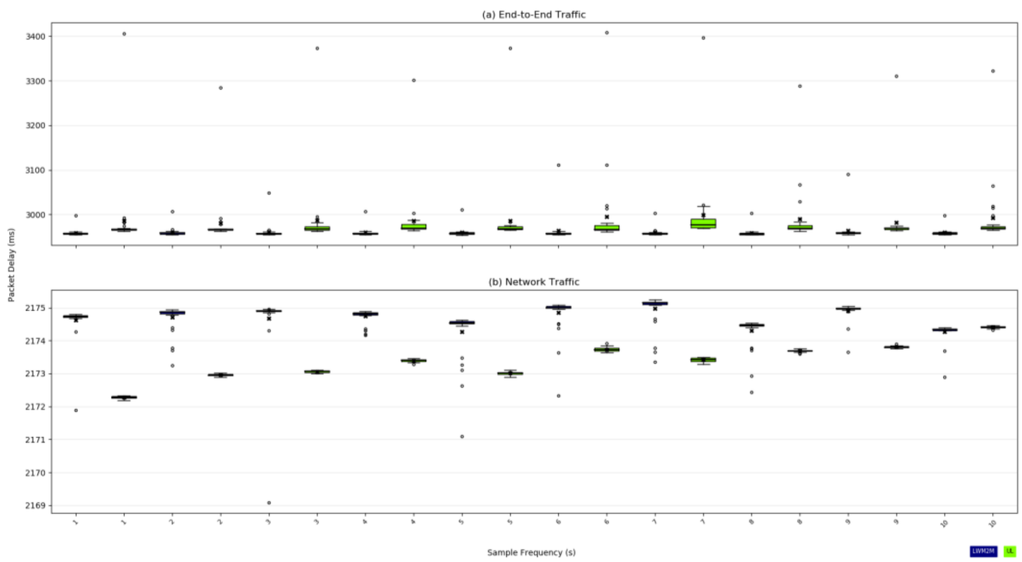

More outliers for LW in both end-to-end and Network traffic.

## Delay, Box, Posts Only, No outliers



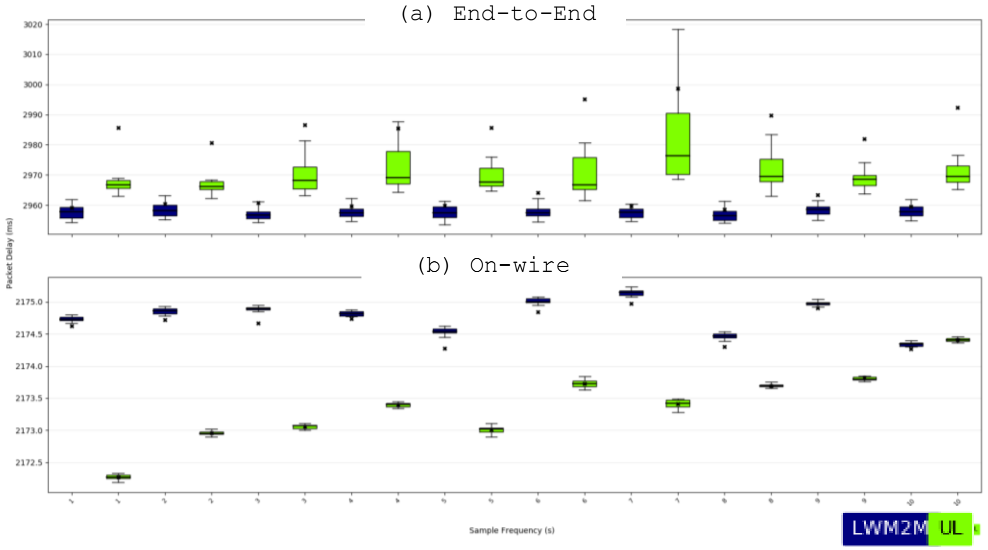

Very high value of outliers compared to the rest of distribution. Hence, mean values in end-to-end traffic for LW are often out of the IQR. End-to-end traffic is slightly more delayed in UL than in LW, while in network we have **reversed results**.
The trend of delays in network decrease with the sampling Frequency. It is more obvious to UL traffic.

## Delay, Mobile, Posts Only




More outliers for UL in end-to-end traffic, while in the network the results concerning the outliers were reversed.

## Delay, Mobile , Posts Only, No outliers




UL distribution very instable in end-to-end traffic. The interquartile range was particular wider for 7 ms sample frequency
end-to-end traffic is slightly more delayed in UL than in LW, while in network we met the **reversed results**.
The trend of delays in UL trafic in network growns with as the sampling Frequency.

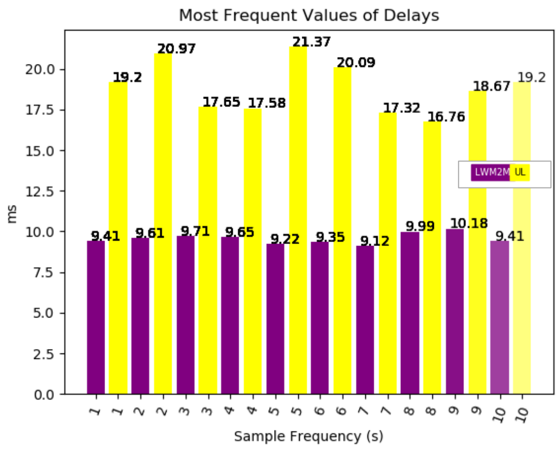
## Traffic Delay Deltas, EndToEnd - Network

| Sample Frequency *(in Seconds)* | Box-Hosted Clients | | | | | | Mobile-Hosted Clients | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Network | | | End-to-End | | | Network | | | End-to-End | | |
| | Mean | Median | Mode | Mean | Median | Mode | Mean | Median | Mode | Mean | Median | Mode |
| 1 | 20.01 | 12.45 | 7.34 | 21.63 | 20.81 | 17.86 | 784.54 | 783.05 | 779.52 | 813.38 | 794.51 | 790.74 |
| 2 | 17.82 | 9.77 | 7.56 | 26.27 | 24.29 | 19.57 | 785.73 | 783.28 | 780.31 | 807.75 | 793.25 | 789.28 |
| 3 | 17.94 | 9.33 | 7.66 | 20.49 | 20.04 | 16.27 | 786.09 | 781.78 | 779.31 | 813.58 | 795.05 | 790.05 |
| 4 | 19.16 | 12.03 | 7.66 | 21.55 | 21.53 | 16.28 | 784.94 | 782.61 | 779.82 | 812.11 | 795.70 | 790.82 |
| 5 | 18.35 | 10.70 | 7.21 | 29.93 | 28.79 | 20.15 | 785.56 | 782.89 | 779.03 | 812.58 | 794.65 | 791.75 |
| 6 | 18.53 | 10.38 | 7.42 | 24.53 | 22.57 | 19.01 | 789.23 | 782.44 | 779.42 | 821.40 | 793.06 | 792.59 |
| 7 | 18.72 | 11.16 | 7.16 | 20.79 | 20.17 | 16.43 | 784.63 | 782.38 | 779.46 | 825.29 | 802.96 | 795.15 |
| 8 | 19.48 | 10.72 | 8.02 | 21.01 | 20.98 | 15.93 | 784.34 | 782.12 | 779.58 | 816.04 | 795.83 | 789.32 |
| 9 | 18.84 | 10.89 | 8.21 | 23.19 | 21.36 | 18.05 | 788.53 | 783.60 | 780.00 | 808.14 | 794.69 | 789.98 |
| 10 | 20.32 | 12.76 | 7.62 | 22.44 | 21.60 | 18.65 | 785.22 | 783.46 | 780.52 | 817.85 | 795.07 | 790.90 |
| **Aproximative Averages (ms)** | **19** | **11** | **8** | **23** | **22** | **18** | **786** | **783** | **780** | **815** | **795** | **791** |

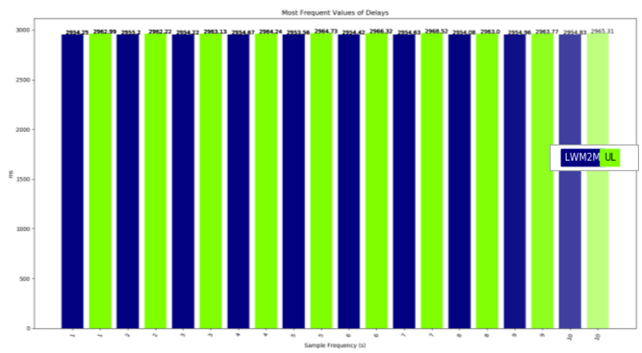
The approximative averages values correspondo to the Procesing Times (PT). It includes the server PT and the Clients hosts PT. Both machines are supposed to be carrying other taskjs than those for the purpose of our experiments. However, we remind that each time one protocol was experimented, the codes fro the other one were commented in the cliens side. In the server side the PT is supposed irrelevante because it deals with communication between twon container of the same Docker. Moreover the ser PT can be doesnot affect the comparison here since it is the same implementation for both protocols. Propagation ~2-3 times slower than processing in mobile-hoste. It is ~7-20 times faster in Box-hostes clients

## Mode Delays – Invisible in Boxplots

BOX



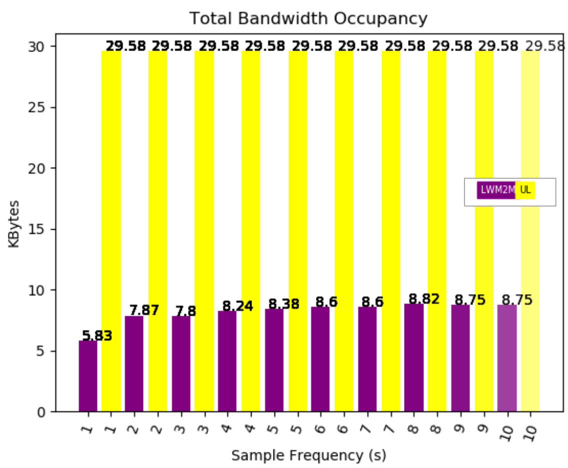

MOBILE



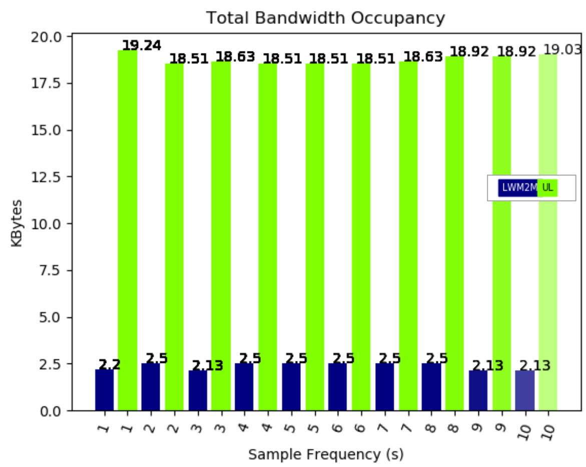

For box-hosted clients, UL the delays most frequent values (math.mode) was almost twice higher compared to LW traffic.
For mobile-hosted clients, the delays modes were almost the same for both protocols (around 2900 ms).

## BW Occupancy, Full Traffic (ack, retrans …)

BOX




MOBILE




UL Constant at 29.58 Kbytes
This includes the whole packets generated in the network to enable the readings to be sent from client to FIWARE server.
UL traffic presented more impact in the network bandwidth, weather with clients instlled in the box or in mobile app.

## Ratio Values UL / LWM2M

| Sample Frequency *(in Seconds)* | Box-Hosted Clients | | | | Mobile-Hosted Clients | | | |
|---|---|---|---|---|---|---|---|---|
| | Network | | End To End | | Network | | End To End | |
| | Bandwidth Occupancy | Delay Mode | Delivery Rate | Delay Mode | Bandwidth Occupancy | Delay Mode | Delivery Rate | Delay Mode |
| 1 | 5.07556971 | 0.64734300 | 1.04004160 | 2.04038257 | 8.73868678 | 0.99885963 | 1.00000000 | 1.00295845 |
| 2 | 3.75912137 | 0.68292683 | 1.00000000 | 2.18210198 | 7.40140570 | 0.99910340 | 1.04166667 | 1.00237547 |
| 3 | 3.79443818 | 0.67317073 | 1.00000000 | 1.81771370 | 8.75803489 | 0.99915859 | 1.00000000 | 1.00301602 |
| 4 | 3.59196016 | 0.65326633 | 1.00000000 | 1.82176166 | 7.40140570 | 0.99934248 | 1.04166667 | 1.00323894 |
| 5 | 3.52918560 | 0.60696517 | 1.00000000 | 2.31778742 | 7.40140570 | 0.99928720 | 1.04166667 | 1.00378188 |
| 6 | 3.43903270 | 0.55958549 | 1.00000000 | 2.14866310 | 7.40140570 | 0.99941609 | 1.04166667 | 1.00402786 |
| 7 | 3.43903270 | 0.45408163 | 1.00000000 | 1.89912281 | 7.44826240 | 0.99917248 | 1.04166667 | 1.00470110 |
| 8 | 3.35337097 | 0.42131980 | 1.00000000 | 1.67767768 | 7.56306130 | 0.99962290 | 1.04166667 | 1.00301955 |
| 9 | 3.38144675 | 0.31472081 | 1.00000000 | 1.83398821 | 8.89302112 | 0.99946206 | 1.00000000 | 1.00298143 |
| 10 | 3.38144675 | 0.30726257 | 1.04004160 | 2.04038257 | 8.94811754 | 1.00004599 | 1.00000000 | 1.00354674 |
| **Approximative Averages** | **3.67** | **0.53** | **1.01** | **1.98** | **8.00** | **1.00** | **1.03** | **1.00** |

**Conclusion**

**Confirmed UL ? LW**

For mobile-hosted clients, the traffic for both protocols were delayed the same way by the network. For box-hosted, UL is half time delayed than LW.

The number of Reading values that arrive to the server is also relatively the same, i.e almost no packet loss.

The bw occupancy in the network was ~3.7 and ~8 times higher by UL for box and mobile-hosted clients respectively.

The end-to-end delay is ~twice higher for UL than for LW in box-hosted clients, while for mobile hosted clients, is almost the same.