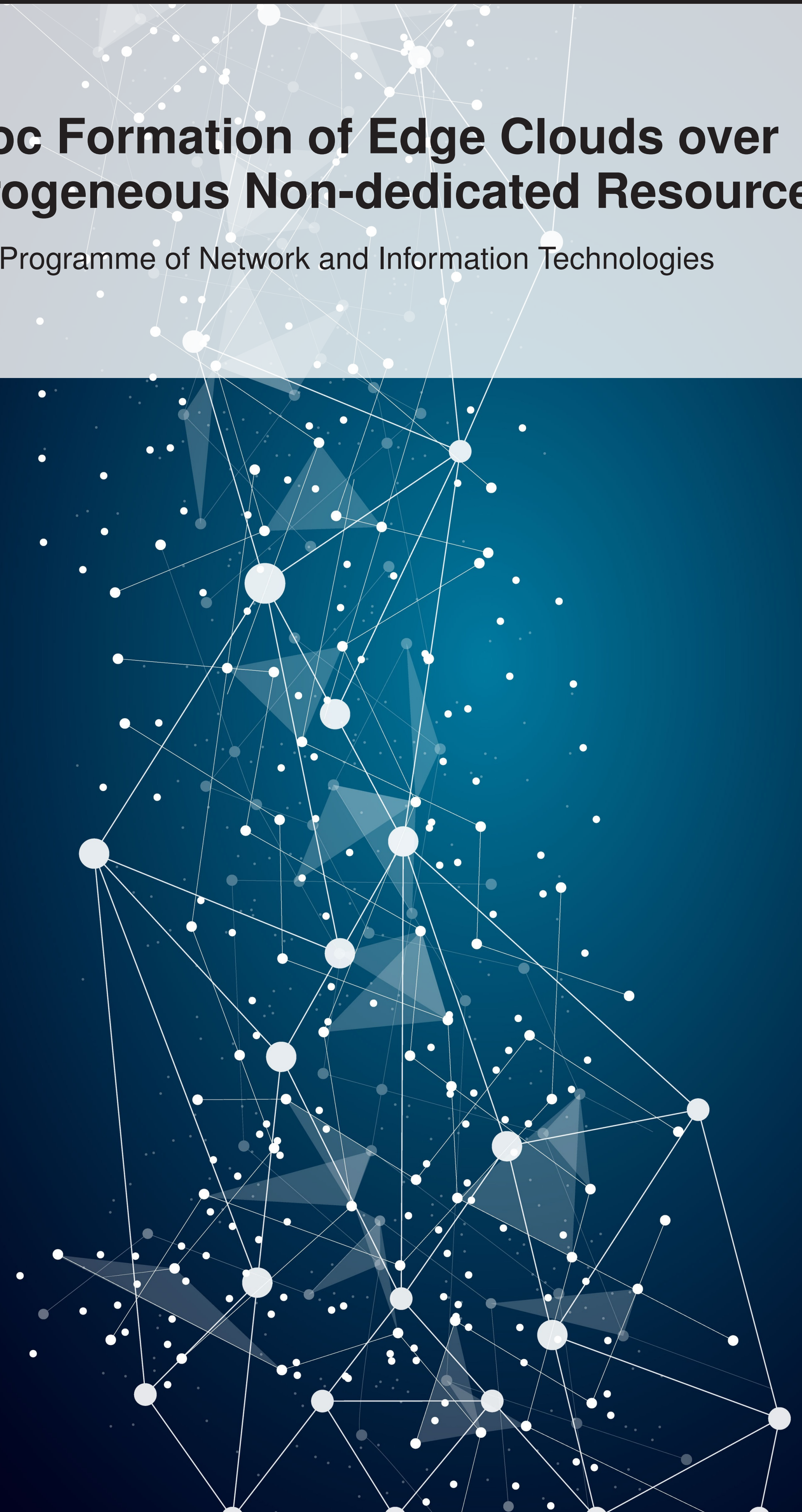
Universitat Oberta de Catalunya

Ad-hoc Formation of Edge Clouds over Heterogeneous Non-dedicated Resources

Doctoral Programme of Network and Information Technologies



UNIVERSITAT OBERTA DE CATALUNYA

Doctoral Programme of Network and Information Technologies

Ad-hoc Formation of Edge Clouds over Heterogeneous Non-dedicated Resources

Author Ana Juan Ferrer
Advisors Dr.Joan Manuel Marquès, Dr.Josep Jorba

Deposit authorization October 21st 2020

Defense date November 30th 2020

Dedication

This work is dedicated to Jordi, whom I could never thank enough for his unconditional support, insuperable common sense and constant encouragement.

Bibliographical sketch

Ana Juan Ferrer holds a UOC Master Degree in Information and Communication Technology Security (2011-2013). Prior to this, she had obtained a UOC Bachelor Degree Computer Engineering (2004-2010) complementing a EUG (UAB) Technical Engineering Degree in Computer Management (1995-1998).

At a professional level, Ana is Head of Edge Computing Unit at Atos, Research & Innovation. She is an Atos Distinguished Expert in Cloud Domain and Atos Scientific Community founding member in charge of Edge and Swarm computing Track. In Atos Research and Innovation department since 2006, her research focuses on Cloud and Edge Computing technologies, distributed systems and service engineering. Currently, she is presently coordinating the PLEDGER project, centring on tools and mechanisms to assess and enact QoS in combined Edge and Cloud scenarios. In addition, Ana has acted as coordinator and main Atos technical architect for mF2C and eFLY projects. MF2C investigated methods and tools to Edge to Cloud workload execution coordination. eFLY explored how Unmanned Autonomous Vehicles technologies can benefit from off-loading computation to Edge and Cloud environments. Previously, she has coordinated ASCETiC and OPTIMIS projects exploring energy efficient and hybrid cloud models. Moreover, she has served as main Atos scientific and technical contact in diverse Edge and Cloud computing research projects. These include among others the following projects: DECENTER, Decentralised technologies for orchestrated Cloud to Edge intelligence; CLOUDBUTTON, Serverless Data Analytics Platform; CLASS, Edge and Cloud Computation, A Highly Distributed Software for Big Data Analytics; and BASMATI, Cloud Brokerage Across Borders For Mobile Users and Applications.

Ana is currently the Cluster leader for the E2 Software & Cloud Unit "Future Cloud" cluster. Furthermore, she holds the position of European Commission Expert Reviewer for monitoring of Cloud actions and H2020 proposals selection.

Moreover, in the elaboration of this Thesis Ana has closely collaborated with Distributed Parallel and Collaborative Systems research group at the Universitat Oberta de Catalunya. This project has led to three publications detailed on Page x.

Other author publications

- Jukan, A., Carpio, F., Masip, X., **Juan Ferrer, A.**, Kemper, N., & Stetina, B. U. (2019). Fog-to-Cloud Computing for Farming: Low-Cost Technologies, Data Exchange, and Animal Welfare. Computer, 52(10), 41–51. https://doi.org/10.1109/mc.2019.2906837

- Papadakis-Vlachopapadopoulos, K., González, R. S., Dimolitsas, I., Dechouniotis, D., **Juan Ferrer, A.**, & Papavassiliou, S. (2019). Collaborative SLA and reputation-based trust management in cloud federations. Future Generation Computer Systems, 100, 498–512. https://doi.org/10.1016/j.future.2019.05.030
- Carpio, F., Jukan, A., Sosa, R., & **Juan Ferrer, A.** (2019). Engineering a QoS provider mechanism for edge computing with deep reinforcement learning. 2019 IEEE Global Communications Conference, GLOBECOM 2019 - Proceedings. https://doi.org/10.1109/GLOBECOM38437.2019.9013946
- Souza, V. B., Masip-Bruin, X., Marín-Tordera, E., Sànchez-López, S., Garcia, J., Ren, G. J., & **Juan Ferrer, A.** (2018). Towards a proper service placement in combined Fog-to-Cloud (F2C) architectures. Future Generation Computer Systems. https://doi.org/https://doi.org/10.1016/j.future.2018.04.042
- Bartolí, A., Hernández, F., Val, L., Gorchs, J., Masip-Bruin, X., Marín-Tordera, E Garcia, J., **Juan Ferrer, A** , Jukan, A. (2018). Benefits of a Coordinated Fog-to-Cloud Resources Management Strategy on a Smart City Scenario. In D. B. Heras & L. Bougé (Eds.), Euro-Par 2017: Parallel Processing Workshops (pp. 283–291). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-75178-8_23
- Quiñones, E., Bertogna, M., Hadad, E., **Juan Ferrer, A.** , Chiantore, L., & Reboa, A. (2018). Big Data Analytics for Smart Cities: The H2020 CLASS Project. Proceedings of the 11th ACM International Systems and Storage Conference, 130. Haifa, Israel. https://doi.org/https://doi.org/10.1145/3211890.3211914
- Masip-Bruin, X., Martín-Tordera, E., **Juan Ferrer, A.**, Queralt, A., Jukan, A., Garcia, J., & Cankar, M. (2018). mF2C: Towards a Coordinated Management of the IoT-fog-cloud Continuum. Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects, 8:1–8:8. https://doi.org/10.1145/3213299.3213307
- Altmann, J., Al-Athwari, B., Carlini, E., Coppola, M., Dazzi, P., **Juan Ferrer, A.**, J., & Violos, J. (2017). BASMATI: An Architecture for Managing Cloud and Edge Resources for Mobile Users. In C. Pham, J. Altmann, & J. Á. Bañares (Eds.), Economics of Grids, Clouds, Systems, and Services: 14th International Conference, GECON 2017, Biarritz, France, September 19-21, 2017, Proceedings (pp. 56–66). https://doi.org/10.1007/978-3-319-68066-8_5
- **Juan Ferrer, A.** (2016). Inter-cloud Research: Vision for 2020. In Procedia Computer Science, 97, 140–143. https://doi.org/10.1016/j.procs.2016.08.292
- **Juan Ferrer, A.** Pérez, D. G., & González, R. S. (2016). Multi-cloud Platform-as-a-service Model, Functionalities and Approaches. Procedia Computer Science, 97, 63–72. https://doi.org/10.1016/j.procs.2016.08.281
- Djemame, K., Armstrong, D., Kavanagh, R., Deprez, J.-C., **Juan Ferrer, A.**, Perez, D. G., & Georgiou, Y. (2016). TANGO: Transparent heterogeneous hardware Architecture deployment for eNergy Gain in Operation.

Proceedings of the 1st International Workshop on Program Transformation for Programmability in Heterogeneous Architectures (PROHA'2016), Barcelona (Spain).

- **Juan Ferrer, A.**, & i Montanera, E. (2015). The Role of SLAs in Building a Trusted Cloud for Europe. In C. Damsgaard Jensen, S. Marsh, T. Dimitrakos, & Y. Murayama (Eds.), Trust Management IX SE - 22 (pp. 262–275). https://doi.org/10.1007/978-3-319-18491-3_22

- Djemame, K., Armstrong, D., Kavanagh, R., **Juan Ferrer, A.** , Perez, D. G., & Antona, D.. (2014). Energy efficiency embedded service lifecycle: Towards an energy efficient cloud computing architecture. Energy efficiency. In: Penzenstadler, B and Lohmann, W, (eds.) CEUR Workshop Proceedings. Joint Workshop Proceedings of the 2nd International Conference on ICT for Sustainability 2014 , 24-27 August 2014, Stockholm, Sweden. CEUR Workshop Proceedings , 1 - 6.

- **Juan Ferrer, A..**, Hernández, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R.M., Djemame, K., Ziegler, W., Dimitrakos, T., Nair, Srijith K, Kousiouris, G., Konstanteli, K., Varvarigou, T., Hudzia, B., Wesner, S., Corrales, M., Sharif, T., & Sheridan, C. (2012). OPTIMIS: A Holistic Approach to Cloud Service Provisioning. Future Generation Computer Systems, 28(1), 66–77. https://doi.org/10.1016/j.future.2011.05.022

- Badia, R. M., Corrales, M., Dimitrakos, T., Djemame, K., Elmroth, E., **Juan Ferrer, A.**, & Hudzia, B. (2011). Demonstration of the OPTIMIS toolkit for cloud service provisioning. Proceedings of the 4th European Conference on Towards a Service-Based Internet, 331–333. https://doi.org/https://doi.org/10.1007/978-3-642-24755-2_40

- Nair, S. K., Porwal, S., Dimitrakos, T., **Juan Ferrer, A.** , Tordsson, J., Sharif, T., & Khan, A. U. (2010). Towards secure cloud bursting, brokerage and aggregation. Proceedings - 8th IEEE European Conference on Web Services, ECOWS 2010, 189–196. https://doi.org/10.1109/ECOWS.2010.33

- Kouvas, G., Grefen, P., **Juan Ferrer, A.** (2010). Business Process Enactment. In N. Mehandjiev & P. Grefen (Eds.), Dynamic Business Process Formation for Instant Virtual Enterprises (pp. 113–132). https://doi.org/10.1007/978-1-84882-691-5_8

- Gaeta, A., Dimitrakos, T., Brossard, D., Piotter, R., Schwichtenberg, H., Gemünd, A., **Juan Ferrer, A.**, Thomson, C. (2010). Bringing it all Together. In T. Dimitrakos, J. Martrat, & S. Wesner (Eds.), Service Oriented Infrastructures and Cloud Service Platforms for the Enterprise (pp. 179–210). https://doi.org/10.1007/978-3-642-04086-3_9

Acknowledgements

I wish to express my most sincere gratitude to my Advisors, Prof. Joan Manuel Marquès and Prof. Josep Jorba for their support over the years in the preparation of this Thesis.

I would also like to acknowledge the assistance of Javier Panadero in utilising and modifying the Distributed large-scale resource allocation simulator(Panadero et al., 2018). I extend my appreciation to Jessica de Armas whose active participation was present in the initial phases of this thesis.

Additionally, I wish to thank the members of Distributed and Operating Systems group at Technical University of Berlin for their warm welcome, fruitful discussions and firm support.

Abstract

The proliferation of connected devices in today's society has reached staggering figures particularly in the last decade. At present, connected devices are not only becoming available everywhere, but they are rapidly gaining complexity in terms of their ability to hold significant compute and storage capacities. In view of this, computing ceases to be confined to certain stationary compute devices in order to allow computing to be embedded and pervasive to everything. This feature combined with the ever-increasing need for timely data processing at the Edge, reveals the urgency to exploit all compute capacity available at the Edge of the network.

This represents a significant breakthrough compared to initial Edge computing developments concentrated on providing low latency compute environments for which IoT devices are solely considered as data sources. Ad-hoc Edge Cloud is a distributed and decentralised Edge Computing system dynamically formed out of IoT Edge computing resources, which aims to exploit this increasingly available compute capacity at the Edge. The marked characteristics of the IoT devices, which form this infrastructure, pose specific challenges to resource management in this context especially due to heterogeneity, dynamicity, and volatility of resources, resulting in the probability of node churn.

Whilst the proposed decentralised Cloud model represents a major step forward from a Cloud perspective, it is rooted in existing research areas such as Mobile Cloud Computing, Mobile Ad hoc Computing, and Edge computing. This Thesis conducts a comprehensive review of the available findings in the field to determine their role in Decentralised Cloud and future computing development.

More importantly, this Thesis dissertation proposes an a specific approach to Decentralised Cloud by developing the concept of Ad-hoc Edge Clouds and creating dynamic ecosystems of IoT Edge Devices in a distributed and decentralised manner, alongside the formulation of an Architecture proposal for its implementation.

The main novelties of Ad-hoc Edge Cloud architecture stem from three key aspects: (1) the consideration of IoT Edge devices beyond their ability to gather data, instead developing the approach of considering these as appropriated execution environments of services to be executed at the Edge of the network; (2) The fact of considering unreliability of resources an essential characteristic of the proposed architecture by designing it centred around node churn and resource volatility issues as the keystone around which the architecture definition gravitates; (3) The view of a fully decentralised architecture which distributes management features to specific participant devices in infrastructures either ephemeral or on-demand created by design. Hence, Ad-hoc Edge Cloud prevents a single point of failure, eliminates the reliance on external management layers with the potential to hinder operation in the event of unreliable connectivity and possesses inherent mechanisms to handle scale.

In addition, this Thesis analyses the challenges posed by the particularities of IoT Edge devices which form this infrastructure in two main research areas.

At the level of Resource management, it elaborates on the mechanisms for enabling dynamic Ad-hoc Edge Cluster formation and management which rely on build-in capabilities of distributed storage and consensus algorithm demonstrating how these can be beneficial in addressing the specific challenges of Resource management in Ad-hoc Edge Clouds.

In terms of Admission control and Service placement, it presents an Admission Control mechanism and together with an associated resource availability prediction model driven by the needs of the dynamic behaviour of participant IoT Edge devices and specifically addressing the aspects of resource instability, dynamic availability and probability of node churn.

List of publications

P1 **Juan Ferrer, A.**, Marquès, J. M., & Jorba, J. (2019). Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing. ACM Computing Surveys 51, 6, Article 111 (January 2019), 36 pages. DOI: https://doi.org/10.1145/3243929

P2 **Juan Ferrer, A.**, Marquès, J. M.,& Jorba, J. (2019). Ad-hoc Edge Cloud: A framework for dynamic creation of Edge computing infrastructures. 2019 28th International Conference on Computer Communication and Networks (ICCCN), 1–7. https://doi.org/10.1109/ICCCN.2019.8847142

P3 **Juan Ferrer, A.**, Panadero, J., Marquès, J.-M., & Jorba, J. (2021). Admission Control for Ad-hoc Edge Cloud. Future Generation Computer Systems, 114, 548–562. https://doi.org/https://doi.org/10.1016/j.future.2020.08.024

Resumen

La proliferación de los dispositivos conectados en la sociedad actual ha alcanzado cifras enormes, particularmente durante la pasada década. Hoy en día, los dispositivos conectados no están solo disponibles en cualquier lugar, sino que están rápidamente adquiriendo mayor complejidad en cuanto a su habilidad para contener capacidad de cómputo y almacenamiento. De este modo la computación deja de estar limitada a ciertos dispositivos computacionales fijos y es capaz de estar embebida e impregnar cualquier objeto. Esta característica, junto con la siempre creciente necesidad de rápido procesamiento de datos en el borde de la red, expone la urgencia de aprovechar toda la capacidad computacional existente en el Edge.

Esto representa un avance significativo en los desarrollos iniciales de Edge Computing centrados en proveer entornos de baja latencia para los cuales los dispositivos IoT son únicamente fuentes de datos. Ad-hoc Edge Cloud es un sistema Edge computing distribuido y descentralizado que se forma a partir de los recursos de computación en dispositivos IoT y cuyo objetivo es explotar toda la capacidad de computo que en mayor medida se encuentra disponible en el borde de la red. Las características específicas de los dispositivos IoT que constituyen esta infraestructura plantean retos específicos en cuanto a la gestión de recursos, debido a la heterogeneidad así como la dinamicidad y volatilidad, que dan lugar a la probabilidad de pérdida de nodo.

Aunque el modelo de Cloud Descentralizado propuesto en este trabajo representa un paso adelante desde una perspectiva Cloud, tiene su origen en áreas de investigación tales como Mobile Cloud, Mobile Ad hoc, y Edge computing. Esta tesis realiza una revisión de la literatura en estas áreas para determinar su papel en el concepto de Cloud Descentralizado. Más importante aún, este trabajo propone un enfoque para la Cloud Descentralizado al desarrollar del concepto de Ad-hoc Edge Clouds, mediante la creación de ecosistemas dinámicos de dispositivos de IoT Edge de forma distribuida y descentralizada, además de proponer una arquitectura para su implementación.

Las novedades que presenta la arquitectura Ad-hoc Edge Cloud derivan de tres aspectos fundamentales: (1) la consideración de los dispositivos IoT Edge más allá de su capacidad de recopilar datos, y alternativamente, desarrollar el enfoque de considerarlos como entornos de ejecución apropiados para servicios que se ejecutarán en el borde de la red; (2) El hecho de considerar la falta de fiabilidad de los recursos como una característica esencial de la arquitectura y por tanto diseñar la arquitectura centrada en la pérdida de nodos y en los problemas de volatilidad de los recursos como piedra angular sobre la que gravita su definición; (3) La visión de una arquitectura totalmente descentralizado que distribuye su gestión entre los dispositivos participantes puntualmente, que es, por definición, generada bajo demanda y efímera. Por lo tanto, Ad-hoc Edge Cloud evita un tener un punto único de fallo, elimina la dependencia de capas de administración externas y posee mecanismos inherentes para manejar recursos a escala.

Además, esta Tesis aborda los retos que presentan las particularidades de los dispositivos IoT Edge que forman esta infraestructura en dos áreas principales de investigación.

A nivel de gestión de recursos, elabora los mecanismos que permiten la formación dinámica de clústeres de Ad-hoc Edge y su gestión, los cuales dependen de las capacidades proporcionadas de forma nativa por los sistemas de almacenamiento distribuido y de sus algoritmos de consenso.

En cuanto a control de la admisión y planificación de servicios, esta tesis presenta un mecanismo de control de la admisión, junto a un modelo para la predicción de la disponibilidad de recursos que contempla el comportamiento dinámico de los dispositivos IoT Edge y que trata los aspectos de inestabilidad de recursos, la disponibilidad dinámica y probabilidad de pérdida de nodo.

Lista de publicaciones

P1 **Juan Ferrer, A.**, Marquès, J. M., & Jorba, J. (2019). Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing. ACM Computing Surveys 51, 6, Article 111 (January 2019), 36 pages. DOI: https://doi.org/10.1145/3243929

P2 **Juan Ferrer, A.**, Marquès, J. M.,& Jorba, J. (2019). Ad-hoc Edge Cloud: A framework for dynamic creation of Edge computing infrastructures. 2019 28th International Conference on Computer Communication and Networks (ICCCN), 1–7. https://doi.org/10.1109/ICCCN.2019.8847142

P3 **Juan Ferrer, A.**, Panadero, J., Marquès, J.-M., & Jorba, J. (2021). Admission Control for Ad-hoc Edge Cloud. Future Generation Computer Systems, 114, 548–562. https://doi.org/https://doi.org/10.1016/j.future.2020.08.024

Resum

La proliferació dels dispositius connectats a la societat actual ha assolit xifres enormes, sobretot durant la passada dècada. Avui en dia, els dispositius connectats no estan únicament disponibles a qualsevol lloc, sinó que a més estan adquirint ràpidament una gran complexitat en la seva capacitat de contenir capacitat de còmput i d'emmagatzemament. D'aquesta forma la computació deixa d'estar limitada a certs dispositius computacionals estacionaris i pot estar embeguda i impregnar qualsevol objecte. Aquesta característica, junt amb la sempre creixent necessitat de processar ràpidament les dades a la vora de la xarxa, exposa la urgència d'explotar tota la capacitat computacional existent a la "vora de la xarxa".

Això representa un avenc significatiu en els desenvolupaments inicials de Edge computing -la computació "a la vora de la xarxa"- que ha estat fins ara focalitzada en proveir entorns de baixa latència pels quals els dispositius IoT són únicament considerats com a fonts de dades. Les característiques específiques dels dispositius IoT que constitueixen aquesta infraestructura plantegen reptes particulars en quant a la gestió de recursos, degut a l'heterogeneïtat, el dinamisme i la volatilitat, resultant en la probabilitat de pèrdues de node.

El model de Núvol Descentralitzat que es proposa en aquest treball, tot i representar un pas endavant des d'una perspectiva Cloud, té els seus orígens en àmbits d'investigació de Mobile Cloud Computing, Mobile Ad hoc Computing, i Edge computing. Aquesta tesi realitza una revisió de la literatura en aquestes àrees per a determinar el seu paper en el concepte de Núvol Descentralitzat.

Encara mes important, aquest treball proposa un enfoc específic per al Núvol Descentralitzat al desenvolupar el concepte de Ad-hoc Edge Clouds, mitjancant la creació d'ecosistemes dinamics de dispositius de IoT Edge de manera distribuïda i descentralitzada, a més de proposar una arquitectura per a la seva implementació.

Les principals novetats que presenta la arquitectura Ad-hoc Edge Cloud deriven de tres aspectes fonamentals: (1) la consideració dels dispositius IoT Edge mes enllà de la seva capacitat de recopilar dades, i per contra desenvolupar l'enfoc de considerar-los com a entorns d'execució apropiats per als serveis que s'executaran a la "vora de la xarxa"; (2) El fet de considerar la manca de fiabilitat dels recursos com a una característica essencial de l'arquitectura proposada en basar el seu disseny en la pèrdua de nodes i als problemes de volatilitat dels recursos com a la pedra angular sobre la que gravita la definició de l'arquitectura; (3) La visió de una arquitectura totalment descentralitzada que distribueix les funcionalitats de la seva gestió entre els dispositius participants en un moment puntual a la infraestructura i que és, per definició, generada baix demanda i efímera. Per tant, Ad-hoc Edge Cloud evita tenir un punt únic de fallida, elimina la dependència de les capes d'administració externes que podrien dificultar la operació en casos de connectivitat poc fiable i té mecanismes inherents per a gestionar recursos a escala. A més, aquesta Tesi analitza els reptes que presenten les particularitats dels dispositius IoT Edge que formen part d'aquesta infraestructura en dues àrees principals de investigació.

A nivell de gestió de recursos, elabora en els mecanismes que permeten la formació dinàmica de clústers de Ad-hoc Edge i la seva gestió, els quals depenen de les capacitats proporcionada de forma nativa pels sistemes d'emmagatzemament distribuït i els seus algoritmes de consens associats.

En quant al control de l'admissió i la planificació de serveis, aquesta tesi presenta un mecanisme per al control de l'admissió, conjuntament a un model per a la predicció de la disponibilitat dels recursos que adreca el comportament dinàmic dels dispositius IoT Edge participants i que tracta els aspectes de inestabilitat de recursos, la disponibilitat dinàmica i la probabilitat de pèrdua de nodes.

Llista de publicacions

P1 **Juan Ferrer, A.**, Marquès, J. M., & Jorba, J. (2019). Towards the Decentralised Cloud: Survey on Approaches and Challenges for Mobile, Ad hoc, and Edge Computing. ACM Computing Surveys 51, 6, Article 111 (January 2019), 36 pages. DOI: https://doi.org/10.1145/3243929

P2 **Juan Ferrer, A.**, Marquès, J. M.,& Jorba, J. (2019). Ad-hoc Edge Cloud: A framework for dynamic creation of Edge computing infrastructures. 2019 28th International Conference on Computer Communication and Networks (ICCCN), 1–7. https://doi.org/10.1109/ICCCN.2019.8847142

P3 **Juan Ferrer, A.**, Panadero, J., Marquès, J.-M., & Jorba, J. (2021). Admission Control for Ad-hoc Edge Cloud. Future Generation Computer Systems, 114, 548–562. https://doi.org/https://doi.org/10.1016/j.future.2020.08.024

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

1.1 Motivation

Computing plays a crucial part in our everyday reality, this fact is attributable to the development of microprocessor technologies throughout the last decades. This has allowed us to have computation at the palm of our hands on our smartphones and thanks to its help we have discovered new ways to utilise applications and services.

At the same time, the advances in Internet of Things (IoT) are triggering the global rise of the number of connected devices, from figures of billions of units estimated to exist today, to over tens of billions of units expected to be available in the near future. Challenges in this context do not only revolve around managing this massive number of connected devices but also to the mechanisms able to cope with data volumes generated by these devices. Reproducing today's IoT environments, created data from connected devices would have to be transmitted over the network and processed into centralised Cloud data centres. This results into significant latencies for IoT data processing. The expected rise of device connections together with the latency across IoT deployments locations and cloud environments, makes the existing IoT approach unsustainable over time. Therefore, the development of IoT heightens the need to process close to the data sources. In addition, the emergence of IoT favours not only the presence of computing on our smartphones but in all kinds of connected devices.

The principal aim of Edge Computing is to tackle this issue by providing a novel paradigm which extends capacities of Cloud Computing to the edge of the network avoiding present latency constraints.

Edge computing (referred to as Fog computing by some precursor authors) has emerged so as to bring Cloud computing capacities at the Edge of the network to address latency issues present in IoT scenarios. Edge computing serves the purpose of providing a compute environment located in the vicinity of data generation sources able to prevent latency issues detected in accessing Cloud services. Edge Computing brings together networking with distinctive cloud principles to define distributed computing platforms in charge of meeting the specific needs of IoT (Bonomi et al., 2012). More precisely Edge, computing is defined as "the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services" (Shi et al., 2016). This definition encompasses the complete set of compute and network devices on their path from IoT data sources to cloud data centres.

Edge computing has undoubtedly taken an initial step towards the decentralisation of Cloud computing by initiating its transformation from the provision of services in dedicated data centres for which resources were perceived as unlimited to a more decentralised approach in which these cloud services are offered in combination with stationary Edge devices (Shi et al., 2016).

Current Edge computing developments regard the Edge as stationary single device environments which provide computing and storage services to a set of IoT devices located in the vicinity. In these, IoT devices are solely viewed as sources of data, and their increasing complexity in terms of computing and storage capacity is disregarded.

However, IoT devices are presently widely proliferating while gaining noteworthy compute resources which lead to significantly expand their capacities. Thanks to the progression of microprocessor technologies materialised over the last decades by the development of Moore's law, compute units have become "smaller, faster and cheaper" (Markoff, 2016) with the passing of the time. This allows today for compute units to be embedded into a wide diversity of IoT devices. Consequently, the materialisation of IoT is conducive to the presence of computing not only on our smartphones, but also on a wide diversity of devices (cars, televisions, cameras, etc.). Current IoT environments encompass simple sensors with 8-bit microprocessors (D. Chen et al., 2016) as well as increasingly complex devices which represent assemblies of non-negligible computing and storage resources aggregated together with diverse sensors and actuators. In this a way, IoT devices are relinquishing their often basic sensing features, and rapidly gaining complexity and sophistication by means of their ability to incorporate considerable computing power. Hence, a significant number of IoT devices are presently becoming de facto, Edge devices.

Moreover, innovative compute devices are being released on the market endowed with application specific processors for AI processing to facilitate embedding compute intelligence into all kinds of IoT devices. IDC("IDC FutureScape: Worldwide IT Industry 2019 Predictions", 2018) predicts that "By 2022, over 40% of organisations' cloud deployments will include edge computing and 25% of endpoint devices and systems will execute AI algorithms.". Some examples of extant market developments which pave the path towards supporting this trend are provided: NVIDIA's Jetson systems for processing on-board edge devices("NVIDIA Jetson", 2019); and Intel movidius("Intel Movidious", 2019) and Google Edge TPU("Google Cloud Edge TPU - Run Inference at the Edge", 2019).

Considering the abovementioned context, computing will cease to be confined to certain devices located on large data centres or stationary edge devices, instead it will be embedded and pervasive to virtually everything. This emergence of innovative computing capacity at the edge of the network is expected to have a long-lasting impact with the rise of all kinds of compute enabled connected IoT devices, such as smart fabrics, connected cars and roads, diverse forms of nano-computing, smart cities and robots are called to be in the short term part of our daily lives.

This enables the emergence of an unprecedented computing continuum which ranges from cloud environments to a myriad of devices at the Edge included dedicated Edge and complex IoT devices. The fact that IoT devices are progressively drawing on noteworthy resources points to an unjustifiable waste to employ them as data sources of resource richer computing environments. The growth in the complexity of IoT devices is calling for an evolution in Edge computing approach addressed by Ad-hoc Edge Ad-hoc Edge Clouds from the perception of IoT devices solely as data sources to the fully exploitation of all compute and storage capacity available in a specific location in all kinds of IoT edge devices.

This is additionally intensified by the expected rise in compute demand at the Edge in the coming years together with expert assessments regarding the development of micro-processor technologies which might not be capable of coping with this demand at the same pace as it has been performed over the last decades. OpenAI (Amodei & Hernandez, 2018) has recently published a study which states that "the amount of compute used in the largest AI training runs has been increasing exponentially with a 3.4-month doubling time since 2012". Increasingly execution of AI computational workloads at the Edge is perceived as one of the major drivers for Edge computing development in the coming years (J. Chen & Ran, 2019; Zhou et al., 2019). At the same time, confidence in Moore's law to be able to respond to this intensified compute demand may somewhat be surpassing the physical capacity of providing ever miniaturised cheaper and faster low-power computing units, as it has been the trend for the last fifty years (Guess, 2015). According to the experts, this anticipated slowdown in Moore's low does not directly transmit the message to computing progress to suddenly stall, nevertheless it can represent that the nature of that progress has to gradually change in two main directions: First, as previously mentioned, by driving the need to exploit hardware heterogeneity with the help of specialised processors to be embedded into all sorts of IoT devices; and secondly, pushing the demand of benefitting from all computing capacity available everywhere(The Economist, 2016).

1.2 Purpose and Scope of the Thesis

The aim of Ad-hoc Edge Cloud is to define an Edge management system which harnesses the increasingly available computing capacity at the Edge of the network so as to form ephemeral compute infrastructures out of resources available in a certain physical space at a specific point in time.

The Ad-hoc Edge Cloud concept, formulated in this Thesis is a novel Edge computing infrastructure management system with the purpose to respond to the rising demands for processing at the Edge driven by the advances of AI via exploiting the existing capacity. Its overall ambition is to enable the on-demand and opportunistic formation of fully centralised and distributed compute infrastructures by making use of the ever-increasing resources already available on IoT Edge devices.

The Ad-hoc Edge Cloud concept materialises the idea of Decentralised Cloud (Juan Ferrer, Marquès, et al., 2019) to avoid unnecessary latencies and exploit accessible complex compute capacity at the edge of the network. At present, IoT Edge devices are solely deemed as objects from which to extract data in Edge computing environments. However, over the last years and thanks to Moore's law, these devices have progressively increased their complexity and many of them today are equipped with substantial compute capacity. Ad-hoc Edge Cloud extends existing Edge computing concept by considering as valid execution environments the vast amount of IoT devices enabled with compute features which is progressively available in all kind of static and mobile devices at the Edge.

The particular characteristics of the IoT Edge devices which can partake in this compute infrastructure pose special challenges to its overall resource management practices, including its Admission control processes and Service management.

These challenges relate to the dynamicity of the resources availability. The dynamicity of resources availability is motivated on the one hand by the yet constrained nature in terms of battery and compute capacities of the heterogeneous

IoT devices. But more importantly, it is determined by the mobile nature of some of these IoT devices due to their associated connectivity instability.

These underline the pressing need for Ad-hoc Edge Cloud to handle scale, heterogeneity, dynamicity, and volatility of resources, resulting into a probability of node churn. In this context, node churn is characterised by the dynamic and volatile behaviour of IoT Edge resources being intermittently available and unavailable to the system.

1.3 Objectives

The following list of objectives further specifies the purpose of this thesis:

O1 **To design and validate key Resource management and Service placement aspects of a framework which enables opportunistic and on-demand formation of Ad-hoc Edge Clouds.** This objective expounds on the investigation of mechanisms which allow the opportunistic formation of ephemeral computing infrastructures making use of capacity available in heterogeneous IoT Edge devices at the Edge of the network. It entails the proposal of a reference architecture for the Ad-hoc Edge Cloud framework, as well as, the identification of use cases to exemplify their potential use and expected benefits. In addition, specific aspects of the Ad-hoc Edge Cloud infrastructure such as Resource management and Service placement will be validated by means of creating prototype tools to analyse the feasibility of ad-hoc creation and decentralised management of Clouds of non-dedicated IoT Edge devices.

O2 **Examining how the specific characteristics of IoT Edge devices affect resource and service management in Ad-hoc Edge Clouds.** IoT Edge devices have specific characteristics with regards to resource availability as part of the Ad-hoc Edge Cloud infrastructure due to their expected massive scale, their heterogeneity and inherent volatility. This objective will characterise IoT Edge resources particularities in two main contexts: Resource management, with the purpose of examining tools and mechanisms which enable the dynamic formation of ad-hoc compute infrastructures exploiting heterogeneous non-dedicated IoT Edge resources usable at a precise moment and location; and Service management, specifically concentrating on service placement and admission control techniques to determine the adequate mechanisms to schedule services to be executed in available IoT Edge resources of the Ad-hoc Edge Cloud infrastructure.

O3 **Research on the specific Resource Management procedures capable of handling dynamic resource availability present in Ad-hoc Edge Clouds.** This objective investigates which are the implications of resource dynamicity to Resource management mechanisms in the overall Ad-hoc Edge infrastructure and its performance. It will elaborate on the implications of resource dynamic addition and removal as well as to analyse the specific levels of node churn (as sudden resource unavailability to the Ad-hoc Edge infrastructure) that can be supported in order not to hinder the appropriate provision of services from Ad-hoc Edge Cloud infrastructures. Overall, this analysis to study which the degrees on resource dynamicity which can be supported in Ad-hoc Edge

Clouds for appropriate service provision. It will make use of both physical in Lab resources (accounting up to 8 nodes) and simulated devices by means of cloud services (utilising up to 100 simulated nodes).

O4 **Study the Admission Control mechanisms to be employed in Ad-hoc Edge infrastructures and validate the use of a resource availability prediction model in this context.** This objective determines the mechanisms to be used in order to perform service placement and admission control decisions in Ad-hoc Edge Cloud infrastructure. It proposes an Admission control mechanism for Ad-hoc Edge Clouds. In addition, it reviews whether the analysis of information of past resource behaviour in relation to its availability to the Ad-hoc Edge infrastructure can serve as a useful indicator for the Admission control mechanism to assess the future behaviour of the node.

1.4 Research Questions

Specific considerations for the proposed objectives are addressed with the help of the following research questions:

RQ1 Are IoT devices suitable devices to create ad-hoc Ad-hoc Edge Computing infrastructures based on decentralised computing approaches? Is completely decentralised management feasible in all contexts? What are the performance overheads that decentralisation and complete distribution bring to ad-hoc infrastructures? What specific use cases could benefit from such ad-hoc edge infrastructures?

RQ2 How are IoT Edge resources characterised? Do IoT Edge devices specific characteristics' condition or determine the applicability of the approach? What are the implications of scale and heterogeneity? What are the particularities of IoT devices which have to be considered on Admission control and Resource management?

RQ3 What are the specific Resource management issues derived from the use of IoT Edge devices? what is the impact of dynamicity and node churn for Resource management in this context? What is the degree of node churn that makes the infrastructure unmanageable?

RQ4 What could be the mechanisms to decide on service acceptance or rejection based on current Ad-hoc Edge infrastructure status? What are the node parameters that determine the Quality of a Node? What are the parameters which indicate their capacity? Is the stability of a node as part of the Ad-hoc Edge an effective measurement to predict resource/node behaviour? Is past behaviour a good indicator of probability of node churn?

The table 1.1 presents a mapping between objectives and the identified specific considerations.

1.5 Research Methodology

The research methodology followed in this thesis has applied a Design and Creation methodology. The intention has been to apply the concept of Learning via Making,

Research Objective	**Specific considerations**
O1- To design and validate key Resource management and Service placement aspects of a framework which enables opportunistic and on-demand formation of Ad-hoc Edge Clouds.	RQ1
O2- Investigate on how the specific characteristics of IoT Edge devices affect resource and service management in Ad-hoc Edge Clouds.	RQ2
O3- Determine which degrees on resource dynamicity can be supported in Ad-hoc Edge Clouds.	RQ3
O4- Study the Admission Control mechanisms to be employed in Ad-hoc Edge infrastructures and validate the use of a resource availability prediction model in this context	RQ4

Table 1.1: Mapping of the specific considerations to research objectives

to design and develop an overall Ad-hoc Edge Cloud framework architecture as a first step. Secondly, by considering the development of the relevant parts and implementation of algorithms imperative for the validation of Resource management and Admission Control processes.

The generated artefacts are a combination of the following elements:

Constructs describing the Ad-hoc Edge Cloud concept, which has evidenced the importance of having a decentralised and distributed resource and service management in the Edge computing context. This process has also included in-depth literature review with the purpose of highlighting the creation of knowledge through analysis of the state of the art in the relevant fields of Mobile Cloud computing, Mobile Ad-hoc Computing and Edge Computing. This analysis has concluded by exposing the crtical need to address decentralisation of cloud computing environments in novel computing continuum environments such as the ones tackled by Ad-hoc Edge Cloud.

Models Demostrating the relation between Edge devices resources and services entities to provide the envisaged capabilities taking into account Edge device specific characteristics exposed as part of the ad-hoc Edge Cloud architecture.

Instantiations A set of prototype tools in charge of assessing feasibility of ad-hoc creation and decentralised management of Ad-hoc Edge Clouds for distributed service management and admission control processes in this infrastructure.

Methods to describe practice derived from the experience. The approach adopted in this thesis is a problem-solving one, where the applied process is described below:

Awareness It is defined as the requirement to articulate problem statement together with the analysis of existing approaches describing differences/similarities of this work approach. Ad-hoc Edge Cloud approaches computing at the Edge from a different perspective than many existing Edge computing tools and frameworks. Currently, existing Edge computing approaches generally consider Edge devices as data sources, neglecting the increasingly available compute capacities present on these devices- In Ad-hoc Edge Cloud the defined problem statement has focused on defining the tools and mechanisms which are able to profit from this increasingly available capacity of heterogeneous Edge devices in order to make them participate in ephemeral compute infrastructures.

Suggestion In order to provide a potential solution to the stated problem, in Ad-hoc Cloud the approach relies on the application of cloud computing and distributed storage techniques technologies to enable the overall resource management in Ad-hoc Edge Cloud infrastructures handling the specific Edge device characteristics and limitations.

Development It consists of the development of the architecture and set of prototype tools for resource availability prediction (as part of Admission controls processes) and distributed service management for Ad-hoc Edge Cloud Framework.

The process followed has not been a linear step-wise process, rather an iterative process, in which, i.e. development provides inputs to problem awareness, based on achievement findings from architecture definition and validation.

1.6 Thesis Contributions

In the upcoming subsections we highlight the main contributions of this Thesis. In addition, we associate them in Table 1.2 to the defined Objectives and Research Questions, as well as generated publications as part of this Thesis.

Contribution #1 A systematic survey of the literature in the areas of Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing

This work aims to set a precedent by providing a systematic literature review of papers in the areas of Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing helping to identify their relations and existing developments as potential contributions to the further evolution of Decentralised Cloud concept. In Chapter 2 we have identified the diverse models of decentralised cloud we have encountered in today's literature including the specific relations among them. Subsequently, we have elaborated on the details of these different approaches. For each of the research areas we have defined existing challenges and approaches; and we have analysed existing papers according to the defined taxonomies. To conclude we observe significant gaps still to be covered by research in order to convert the Decentralised Cloud vision into a reality.

Objectives O1-4

Research Questions RQ1-4

Publications P1

Contribution #2 A Characterisation of the Ad-hoc Edge Cloud Concept, Architecture and Use cases

Taking as starting point the detailed state-of-the-art analysis, Cloud computing technologies are regarded as indispensable to progress from existing data centre based centralised architectures towards the incorporation of complete decentralised models which prosper from the growing computing capacities at the Edge of the network over all kinds of heterogeneous IoT connected devices. With market penetration figures of connected devices escalating to unprecedented levels, the ability to exploit growing compute capacity at the Edge of the network available on heterogeneous IoT Edge devices has become paramount. Moreover considering that these IoT Edge devices have ceased to be considered mere data sources, considering the fact that today several of these devices sustain significant compute and storage resources. With this purpose, we define the innovative concept of Ad-hoc Edge Cloud as the enabler to dynamically forming computing infrastructures out of available IoT devices in a decentralised and distributed manner. In addition, we determine the distinctive and novel aspects of Resource management in this context and we detail use cases which would benefit from the envisaged computation hyper distribution in Ad-hoc Edge Clouds.

In the interest of bringing the Ad-hoc Edge Cloud concept to a concrete concept implementation, we have defined an architecture which entails two main building blocks: Edge Device and Ad-hoc Edge Cloud Context. Both building blocks are expected to be deployed in any IoT Edge device participating into an Ad-hoc Edge Cloud. Contexts determine the two dimensions we define for the architecture features: the management of a certain node belonging to the infrastructure at a specific point in time, Edge Device Context; and the management of the Edge overall cluster which is fully decentralised and distributed thanks to the use of distributed storage systems, Ad-hoc Edge Cloud Context. To the best of our knowledge which is derived from the analysis of existing works in the Chapter 2, the provided architectural approach clearly distinguishes from existing works in Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing.

The main novelties stem from three main aspects: (1) the consideration of IoT Edge devices beyond their ability to gather data, instead developing the approach of considering these as appropriated execution environments of services to be executed at the Edge of the network; (2) The fact of considering unreliability of resources a essential characteristic of the proposed architecture by designing it centered around node churn and resource volatility issues as the keystone around which the architecture definition gravitates; (3) The view of a fully decentralised architecture which spreads management features over punctual participant devices in infrastructures which can be by design ephemeral and on-demand created. Hence Ad-hoc Edge Cloud avoids a single point of failure, eliminates reliance on external management layers which can hinder operation in cases of unreliable connectivity and possesses inherent mechanisms to handle scale.

Objectives O1, O2

Research Questions RQ1, RQ2

Publications P2

Contribution #3 A mechanism for Ad-hoc Edge Cluster instantiation and management

Ad-hoc Edge Clouds requires specific features in relation to Resource management in order to cope with existing challenges regarding nodes massive scale, heterogeneity and availability, especially regarding node churn. With a view to addressing these Resource management specific requirements, this Thesis has developed a protocol which determines the life-cycle of resources in the Ad-hoc Edge infrastructure. More importantly, it has elaborated the crucial mechanisms for Ad-hoc Cloud cluster initialisation and management relying nn built-in capabilities of distributed storage and consensus algorithm demonstrating how these can sustain the specific challenges of Resource management in Ad-hoc Edge Clouds.

This Thesis has examined and determined the specific levels of node churn assumable for a proper service provision on Ad-hoc Edge Cluster management. While, current Edge computing systems rely on uniquely on static Edge devices, the evaluation performed as part of this work and reported in P2, demonstrates the feasibility of fully decentralised cluster management systems based on widely used distributed storage Etcd and its associated consensus mechanism Raft. This set up has displayed acceptable performance under node churn rates below 75% , permitting to determine as feasible in these cases the Ad-hoc Cloud overall ambition of a decentralized and distributed Ad-hoc Edge cluster management system exploiting accessible capacity at a specific moment and location.

Objectives O1, O2, O3

Research Questions RQ1, RQ2, RQ3

Publications P2

Contribution #4 A mechanism for Admission control and service placement which includes a validated IoT Edge Resource Availability prediction model

Ad-hoc Edge Cloud Admission Control processes selects from the existing pool of resources at a specific point in time, those assessed to be the existing set of resources which better serve the service execution request. This mechanism establishes a two step process in which resources are first filtered according to their dynamic and static features considering the service execution needs. Results of the filtering process are then ranked according to assessed Node Quality. This concept stems from previous research in the context of Volunteer Computing in (Panadero et al., 2018), adapting it and validating it in the context of the Ad-hoc Edge Computing. As a result, this Thesis addresses recognised needs (see Section 5.5) in Fog and Edge computing service placement and scheduling in order to respond to the challenges regarding the phenomena of resource instability, dynamic availability and probability of node churn.

While mechanisms for Resource availability have been an extensive area of research in Volunteer and Contributory systems (see Section 5.5 for details). The Resource Availability prediction model defined of Ad-hoc Edge Cloud represents to the best of our knowledge, the first serious attempt to bring Resource availability prediction research area to the Edge computing environment. We understand this is due to the fact that so far typically Edge computing environments have not yet considered IoT Edge devices as appropriate execution environments for Edge service execution. Therefore, their dynamic behaviour in which this dissertation focuses on is to be considered in the course of time. In addition, it is worth remarking, that our work on this Thesis also differs from existing previous papers in the areas of Volunteer and Contributory computing in the time granularity of interest for this prediction model, motivated again from the expected dynamic behaviour of participant IoT Edge devices.

Objectives O1, O2, O4

Research Questions RQ1, RQ2, RQ4

Publications P3

Contribution	Related Research Objectives	Related Research Questions	Publications
Contribution #1 A systematic survey of the literature in the areas of Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing	O1-4	RQ1-4	P1
Contribution #2 A Characterisation of the Ad-hoc Edge Cloud Concept, Architecture and Use cases	O1, O2	RQ1, RQ2	P2
Contribution #3 A mechanism for Ad-hoc Edge Cluster instantiation and management	O1, O2, O3	RQ1, RQ2, RQ3	P2
Contribution #4 A mechanism for Admission control and service placement which includes a validated IoT Edge Resource Availability prediction model	O1, O2, O4	RQ1, RQ2, RQ4	P3

Table 1.2: Mapping of Thesis contributions to Research Objectives and Questions

1.7 Outline of the Thesis

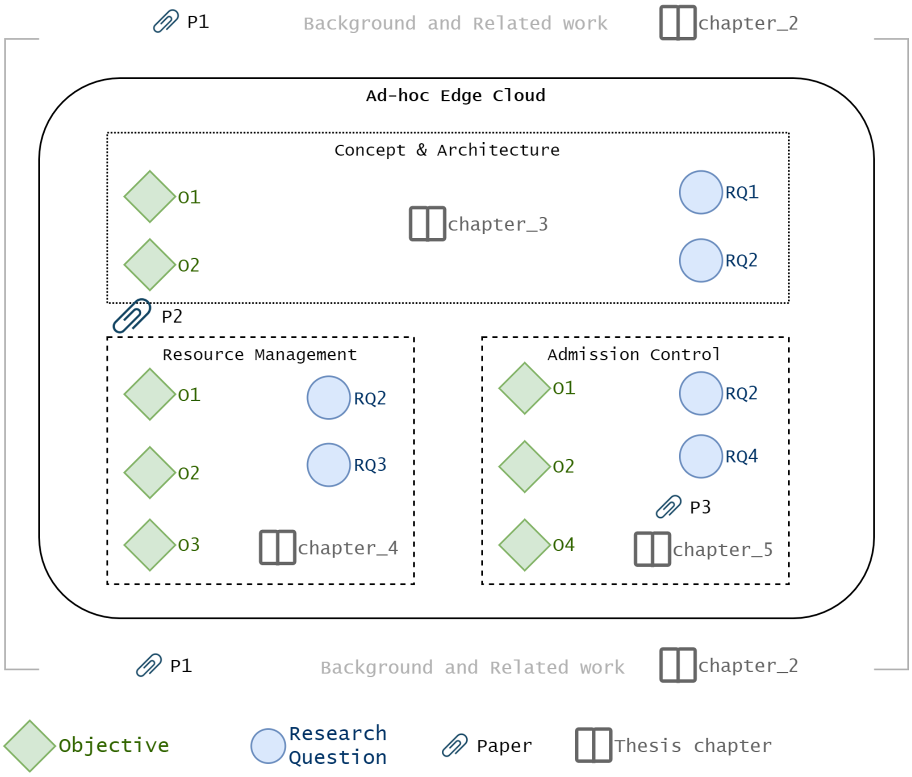


Figure 1.1: Thesis outline.

Figure 1.1 presents the overall structure of this Thesis, representing the Thesis Chapters, related publications as well as addressed Objectives and Research questions in each Thesis parts.

The remainder of this Thesis is organized as follows:

Chapter 2 provides the necessary background and a detailed State of the Art analysis for existing papers related to the concept of Ad-hoc Edge Cloud, and generally speaking, towards the decentralisation of Cloud computing. In particular, this section elaborates on the existing relation among Cloud Computing decentralisation and background technologies such as Mobile Cloud, Mobile Ad-hoc and Edge computing. For these three technologies existing challenges and approaches, as well as, works classifications are defined in order to provide the necessary context for Ad-hoc Edge Cloud thesis.

Chapter 3 expands on the concept of Ad-hoc Edge Cloud by defining the specific characteristics of IoT Edge devices partaking in this infrastructure and the implications in its infrastructure model and overall architectural framework. In particular, the different building blocks and components for the Ad-hoc Edge cloud framework are presented in this section, as well as, potential use cases which could benefit from such developments.

Chapter 4 depicts the specificities of Resource management in the context of Ad-hoc Edge Clouds. In order to do so, it introduces the protocol which enables IoT devices to participate into Ad-hoc Edge Cluster infrastructure. Afterwards, it provides details on enabling mechanisms for cluster instantiation and management. To finalise it presents the results of the evaluation of two critical aspects of services management in this context: scalability and impact of node churn. Experimentation evidenced the relevance of scale both in terms of having the ability to support churn rates along with the management performance overheads it brings.

Chapter 5 develops the Admission control processes for Ad-hoc Edge Cloud. Admission Control processes represent the decision which determines whether to accept a service to be executed in the infrastructure and in the affirmative case, to identify the set of the most appropriate available IoT Edge resources in order to place the different service components. The proper considerations regarding the level of Admission control mechanisms in Ad-hoc Edge Cloud are defined taking into account its past behaviour in terms of connections and disconnections, representing the source on which we base the resource availability prediction model for infrastructure.

Finally, Chapter 6 provides a recapitulation of this Thesis achievements as well as, the perspective of the author on future research in this context.

CHAPTER 2

Background and Related Work

2.1 Overview

Cloud computing initially emerged in the space in which “we transitioned from an era in which underlying computing resources were both scarce and expensive, to an era in which the same resources started to be cheap and abundant“ (Kushida et al., 2015). Current approaches for Cloud computing are based on dedicated Data Centres managed by enterprises where resources are perceived as unlimited, in which everything is delivered as a service in stationary resources set-ups. Cloud computing has enabled the democratisation of computing. It has provided the illusion of infinite computing and allowed for the radical acceleration of commoditization of computing by making the concept of utility computing a reality.

Existing Cloud computing developments emerged as part of a centralised paradigm in which large and fully equipped Data Centre concentrate the available computing power. Gartner's Edge Manifesto (Gartner, 2017) has demanded “the placement of content, compute and Data Centre resources on the edge of the network, closer to concentrations of users. This augmentation of the traditional centralised Data Centre model ensures a better user experience demanded by digital business“.

Initial steps towards decentralisation of Cloud Computing are being realised through the emergence of Fog (Bonomi et al., 2012) and Edge computing (Garcia Lopez et al., 2015). These are recognised to be rooted in the Cloudlet concept in Mobile Computing (Bilal et al., 2018; Satyanarayanan, Simoens, et al., 2015). Edge and Fog computing are currently being developed under the premise of static computing devices (or sets of them) which serve as computing environments located in the vicinity of data generation areas in order to avoid latencies generated by application of Cloud computing to IoT scenarios. In this context, IoT devices are solely considered as mere sources of data presenting minimal actuation capacities.

Nevertheless, the expected gains in complexity of IoT devices becoming complex IoT Edge devices anticipate a future in which connected things go beyond existing basic data gathering and actuation and offer enhancements to execute deep learning and AI processing. Therefore, this evolution will bring about novel opportunities to future evolution of Edge computing by summing up ever increasing computing capacities available in complex connected IoT devices at the edge of the network.

Complex IoT devices are assembles of computing and storage resources with diverse actuators and sensors, conceptually similar, to Mobile Devices. Connections among Mobile Cloud Computing and Evolution of Edge Computing do not end

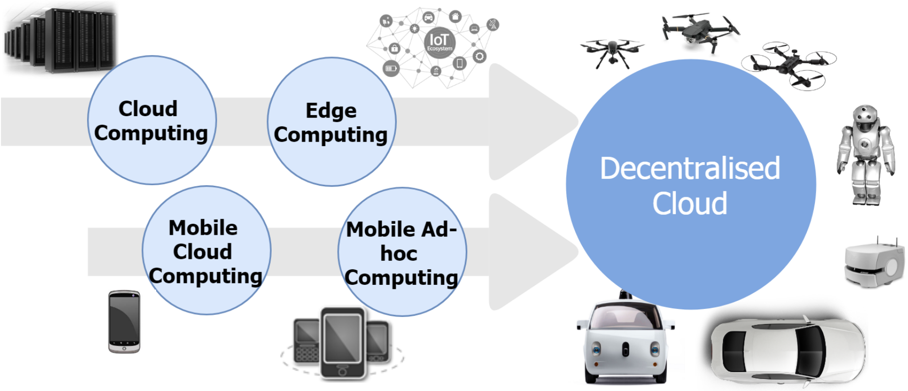


Figure 2.1: Cloud, Edge, Mobile Cloud and Ad-hoc Cloud Computing evolution paths.

here: the fact that these complex IoT devices have a number of constraints in their size and energy harvesting is also shared with Mobile Cloud computing works.

Additionally, the fact that often complex IoT devices are capable of moving raises novel challenges with regard to resource reliability, unstable connectivity and overall computing environment dynamicity, which for a number of years have been deeply analysed in the context of Mobile Cloud Computing. This reinforces the idea that future evolution of Edge computing has an intrinsic relationship with Mobile Cloud Computing.

Beyond existing Mobile Cloud Computing Cloudlet and Edge computing concepts relation, we claim that Mobile Cloud and Ad-hoc Computing concepts create novel forms of distributed and opportunistic computing which will become a key building block for the evolution of existing Cloud and Edge computing towards the Decentralised Cloud. As illustrated in Figure 2.1 evolution paths of these technologies have so far occurred in parallel, however we anticipate their convergence in the Decentralised Cloud Concept.

2.2 Setting the scene: Cloud, Edge, Mobile and Ad-hoc Computing Context

The movement towards Cloud decentralisation is a novel approach from a Cloud perspective. There is extensive research in the areas of Mobile Cloud Computing (MCC), Mobile Ad-hoc Computing (MAC) and Edge computing which can be explored so as to gain understanding of existing approaches and challenges this poses. Figure 2.2 provides a high level view identifying relations among these technologies. Table 2.1 details conceptual differences and similarities in their current approaches.

Cloud computing (Mell & Grance, 2011) initial developments revolved around on the Infrastructure as a Service, having AWS EC2 as its main representative. Nowadays Cloud computing is considered both a business and delivery model which permits the acquisition of a wide range of IT capabilities encompassing from infrastructure, development environments and security features to final user

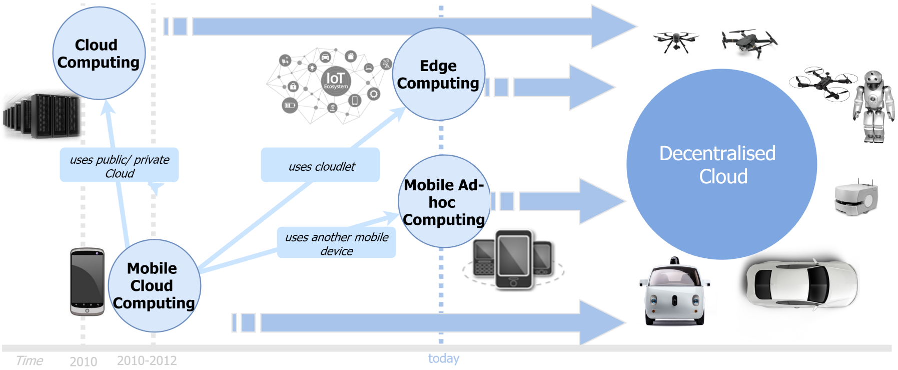


Figure 2.2: Relations among Decentralised Cloud models.

applications. Thus, nurturing the ambition of providing an infinite all-purpose elastic IT utility in which everything is open to being consumed by anyone, from anywhere “as-a-Service“.Drawbacks of the Cloud computing model have been identified in relation to vendor lock-in, inflexibility of SLAs, total cost of ownership, security and data protection, among others (Armbrust et al., 2009). The concept of Hybrid cloud and Multi-Cloud (Juan Ferrer et al., 2012; Petcu, 2013) refers to the seamless interoperability among diverse Cloud providers, focusing on specifically tackling vendor lock-in challenges. Cloud Computing supports elastic delivery of services which in the case of major Cloud providers are delivered from centralised Data Centres distributed across diverse regions all around the world. While this approach has been proven to be powerful for a very large number of scenarios, Internet of Things (IoT) and massive number of connected Things bring novel challenges to its development addressed through Edge and Fog Computing.

The emergence of Internet of Everything - the networked connection of people, process, data and things - is expected to exponentially grow the number of connected devices worldwide, from billions of units available today, to orders of magnitude of tens of billions of units expected to be deployed in the coming years. At present we are observing evolutionary forms of Cloud Computing, such as Edge and Fog, starting to break the Data Centre barriers so as to provide novel forms of computing embracing computing power and data resources increasingly obtainable everywhere. These are forcing existing Cloud computing environments which emerged as part of a centralisation paradigm to evolve to decentralised environments avoiding drawbacks of large data movements and latency, specifically found in IoT scenarios (Cisco Systems, 2016). These new forms of Cloud are making the Cloud concept create a more distributed approach in order to lead to better performance and enabling a wider diversity of application and services, complementarity to traditional X-as-a-service cloud models which is used as resource rich environment.

Major cloud providers such as AWS (“Amazon Web Services Greengrass”, 2017) and Azure (“Azure IoT Edge”, 2017) are increasingly featuring Edge Computing services, as a way to extend their offerings for IoT scenarios. In doing so, Edge computing has become an evolution of well-established Cloud offerings.

Both for Edge, Fog and Mobile Cloud Computing traditional Cloud models are perceived as the resource rich environment to be used in order to extend limited capacities of these environments.

In parallel to the hype around Cloud computing, mobile technologies experienced an unprecedented growth both in development and adoption. Mobile Devices and Cloud Computing have increasingly evolved in the concept of Mobile Cloud Computing (MCC). MCC is a research area which "aims at using cloud computing techniques on storage and processing of data mobile devices" (Guan et al., 2011). In traditional approaches to MCC, Cloud computing environments are used to overcome Mobile devices limitations. These limitations are often outlined in terms of battery lifetime as well as processing and storage capacity.

In order to overcome mobile devices limitations, three different approaches can be found in literature to augment limited mobile devices capabilities:

- Approaches which boost mobile devices capabilities with resources from Cloud environments by means of public or private environments. These approaches make the assumption that employed resources in the Cloud offer rich capacity and ensured availability.
- Approaches which rely on servers located close to the mobile device position, called Cloudlets.
- Approaches which are dependent on other mobile devices to increment their capacities (therefore relying on resources in principle subject to the same mobile's devices constraints and limitations). These approaches have been coined under the term Mobile Ad-hoc Cloud (MAC)(Yaqoob et al., 2016).

Mobile Cloud Computing Cloudlet concept (Satyanarayanan et al., 2009) is a precedent to Edge and Fog computing. It defines the concept of a proximal cloud that brings closer computing capacities so as to avoid latency to the mobile devices its serves. Diverse authors have drawn on this connection. Examples of these are (Satyanarayanan, 2017; Satyanarayanan, Simoens, et al., 2015) and (Bilal et al., 2018).

The forms of Mobile Cloud Computing (MCC)which consider other Mobile Devices to make use of their available resources recently have been classified as Mobile Ad-hoc Cloud (MAC)(Yaqoob et al., 2016). The concept of MAC develops a common umbrella term for a number of works both in MCC and other research environments which consider mobile devices as valid execution resources (Yaqoob et al., 2016). Historically, MCC motivation has been the need to extend Mobile Devices limited resources to richer execution environments. Fuelled by the increased capabilities of Mobile Devices, this research area aims to go beyond these approaches considering the Mobile Device a valid Cloud resource and therefore capable of taking part in Computing infrastructures. Although the concept had already been addressed in previous MCC works, it presents the characteristic of the opportunistic behaviour of the environments very much of interest for the development of decentralised Cloud concept.

The call towards the decentralisation of Cloud computing is present in a wide variety of works and under diverse terms. (Satyanarayanan, 2017) contextualises the current trend towards Cloud computing decentralisation in the context of alternating waves of centralisation and decentralisation which have affected computing since the 60's. In these, centralisation of computing has been prevalent in 60's and 70's through batch processing and time-sharing and from mid-2000's employing traditional centralised Cloud computing models; whereas alternating with

	Cloud Computing	**Mobile Cloud Computing**	**Mobile Ad-hoc Computing**	**Edge Computing**
Motivation	To provide IT services on-demand	To provide additional capacity to resource constrained mobile devices	To provide additional capacity to resource constrained mobile devices	To reduce latency in computation tasks of Data Generated by IoT
Client	Any application	Mobile applications	Mobile applications	IoT applications
Resource Nature	Steady servers in Data Centres	Mobile devices complemented with capacity on Cloud computing environments, Steady servers located in the vicinity (cloudlet) or other Mobile devices	Mobile devices complemented with capacity of other Mobile devices	IoT devices complemented with capacity of steady servers located in the vicinity of IoT data generation areas
Means to acquire additional capacity	Federation with other Clouds	Cloud, Cloudlet and other mobile devices	Other mobile devices	Cloud
Optimisation problem	Capacity, QoS	Energy, Capacity	Energy, Capacity	QoS (latency)
Representative Comm. Offerings	AWS, Azure, Google Cloud	None	None	AWS Greengrass, Azure IoT Edge, FogHorn
Standardisation	NIST, ETSI, SNIA, DMTF, OASIS, etc.	None	None	OpenFog Consortium, ETSI

Table 2.1: Cloud, MCC, MAC, Edge computing concepts comparision

decentralisation in 80's and 90's via the emergence of personal computing and in which Edge computing presents the last episode of this on-going trend.

Shi (Shi et al., 2016), among many authors, has explained the need of decentralisation motivated by the development of richer IoT devices which have changed their role from simple data consumers to rich data providers. Overall rich IoT devices are expected to generate such amounts of data that in the longer term it will become impractical to centralise all their processing.

Garcia-Lopez (Garcia Lopez et al., 2015) further elaborates the factors that call for placement of computing at the edge with the help of four elements: Proximity, bringing facilities to distribute and communicate information; Intelligence, due to the fact that IoT devices increase computing capacities at a rapid pace; Trust and Control, by permitting data sources to remain in control of generated data and application management; and Humans, making them the centre of all interactions. In addition, Garcia-Lopez recognises further research challenges to be addressed in Cloud computing for realising novel highly distributed Edge architectures and middleware which go beyond Hybrid Cloud developments and coping with specific challenges of decentralisation and "computation trade-offs between mobile terminals and cloud servers". These are expected to have to deal with issues affecting stability on the availability of edge devices, such as devices' churn, fault tolerance and elasticity aspects; all of them being core aspects of research in Mobile Cloud Computing in the last years.

A similar approach is taken by Varghese when analysing the future of Cloud computing in the next decades in (Varghese & Buyya, 2018). It precisely identifies MCC Cloudlet and MAC concepts as foundations for the evolution of Cloud Computing infrastructure towards the decentralised computing infrastructure in which resources are away from the Data Centre boundaries.

At the time of writing, there is still not a term that delimits the above mentioned highly decentralised computing infrastructure. Some authors such as (El-Sayed et al., 2018) refer to this just as Edge Computing, declaring that existing Edge Computing development just reflect an embryonic evolution stage of what it can become by utilising the incorporation to the concept of "smartphones, sensor nodes, wearables and on-board units where data analytics and knowledge generation are performed which removes the necessity of a centralised system".

Other authors prefer to define a specific term for this foreseen Edge capacity advancement. This is the case for Villari (Villari et al., 2016) who defines Osmotic Computing as "a new paradigm to support the efficient execution of IoT services and applications at the network edge". Osmotic Computing considers again distributed across Edge and Cloud application execution elaborating on MCC concepts to define its evolution requirements while acknowledging the need of reverse "mobile (cloud) offloading" mechanisms which move functionalities from Cloud computing to Edge devices. (Bojkovic et al., 2017) has coined the term Tactile internet for the evolution of Fog (Edge) computing that combined with developments in SDN and NVF able address requirements for ultra low latency and high availability required in scenarios such as "autonomous vehicles, haptic healthcare and remote robotics" among others.

Back in 2014, Lee (Lee et al., 2014) invented the TerraSwarm concept, as a set of technologies able to integrate cyber and physical worlds in a way that "Mobile battery-powered personal devices with advanced capabilities will connect opportunistically to the Cloud and to nearby swarm devices, which will sense and actuate in the physical world". These herald beginning of a close link which can be detected among MCC and MAC and the future of Cloud and Edge Computing. It is interesting to note that the consideration of Mobile device in TerraSwarm was also surpassing existing smartphone technology, but also considering Autonomous vehicles and Unmanned aerial vehicles (UAVs). While these, still today, seem futuristic scenarios, analysis of UAVs as "near user edge devices which are flying" was provided in (Loke et al., 2015). This was anticipating the use of mobile cloud computing cloudlet servers in the air on drones as "Data mules", able to bring data where it can be better processed, or by means of the development of "Fly-in, Fly-out infrastructure", able to provide punctual computing services in a specific location.

However, today specific implementations of these are starting to emerge, showing their potential to develop in the medium term. Some of the most noteworthy examples are as follows: (Jeong et al., 2018) who provides a Cloudlet mounted in a UAV that provides offloading capabilities to a series of static mobile devices and (Valentino et al., 2018) which develops an opportunistic computational offloading system among UAVs. All these works evidence that the nature of UAVs, and generally speaking robots and autonomous vehicles, share device characteristics with traditional mobile devices in the form that they present constraints in terms of computational and storage capacity, battery and energy supply limitations. Together with the fact of relying on unstable network links due to mobility, which drives to specific device reliability and volatility issues not yet explored in stationary resource environments present in Edge and Cloud computing today.

While specific needs of smartphones have driven the development of MCC, we anticipate that the emergence of rich IoT devices in the form of complex IoT Edge devices will push towards the development of Decentralised Cloud.

Whereas it is widely recognised that MCC Cloudlet concept is the precursor of Edge computing, further evolution of this concept will be rooted in other forms of Mobile Computing, which has relied on the interconnection of constrained devices to resource richer environments in traditional clouds, and more importantly, in the opportunistic formation of computing infrastructures among mobile devices and MAC.

This will be motivated by the on-going trend towards decentralisation but also by the increasing pressure to take advantage of all available computing capacity. As the evolution of Moore's law is progressively reaching its limits and computing demands will solely increase with the advent of more complex IoT devices and their expected data deluge generation. Parallel advances in Deep learning and artificial intelligence will intensify this need by multiplying the requirement for complex processing at the Edge.

All together it evidences the need for Cloud and Edge computing to drawn inspiration from and explore in depth evolutions that have happened in the context of MCC and MAC in order to address novel challenges that Decentralised Cloud is bringing to this context, removing the boundaries which have existed up to this point among these technologies employing resources that are analogous in nature.

A clarification should be done about the terms used in the rest of this paper. At the time of writing, there is still much controversy regarding the use of Fog and Edge computing terms. OpenFog consortium ("OpenFog Consortium", 2016) in its reference architecture ("OpenFog Architecture Overview", 2016) alludes to the fact that Fog Computing is often erroneously named Edge Computing, and argues it in the differences at levels of Cloud interaction, hierarchy and layers and aspects addressed. In particular indicates that "Fog works with the cloud, whereas edge is defined by the exclusion of cloud. Fog is hierarchical, where edge tends to be limited to a small number of layers. In additional to computation, fog also addresses networking, storage, control and acceleration.“ (“OpenFog Architecture Overview”, 2016) Fog Computing is a term coined by CISCO in its enlightening paper “Fog Computing and Its Role in the Internet of Things“ (Bonomi et al., 2012). In this publication, Fog computing is defined as a “highly vitalised platform“ between end-devices and Data Centre clouds which provides compute, storage, and networking services (see section 4 for details on definition). Recent publications of OpenFog Consortium blog (Kubik, 2017) extends this definition, to consider Fog Computing “a continuum or a range of computing that goes from the cloud, to the edge, to the devices“.

Currently, many authors are considering “Fog Computing“ a vendor specific term, and therefore opt for using “Edge Computing“ term. ETSI has also coined the term “Mobile-edge Computing“ (“Mobile-edge Computing”, 2016), which explicitly focus on the Network aspects of the technology. While the research and standardisation communities are currently debating the appropriate term to use, major cloud and technology providers have released related products, tagged as Edge Computing, to the market. These commercial products do not adjust to differentiation levels provided by OpenFog Consortium, instead, they consider Edge computing all computing environments outside Data Centre boundaries. The growing popularity of these products, evidenced by Google Trends “Fog Computing“ and “Edge Computing“ comparison of terms (“Google Trends, fog computing vs Edge Computing”, 2018), makes us opt for using the Edge computing term throughout this paper. However, as our work is a literature survey, both terms Fog and Edge Computing will be used as synonyms, making use of the term used by the referenced author in the different analysed studies.

In this section we have identified the diverse models of decentralised cloud we encounter in today’s literature including the specific relations among them. The upcoming sections provide a systematic literature review of works in the areas of Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing helping to identify their relations and existing developments as potential contributions to further evolution of Decentralised Cloud concept. These sections elaborate on the details of these different approaches including definition of existing challenges and approaches and analysis of existing works according to the defined taxonomies. Conlusions section entails the observation of significant gaps still to be covered by research in order to make the decentralised Cloud vision a reality.

2.3 Mobile Cloud Computing (MCC)

MCC is an area of research meant to connect Mobile Computing (Satyanarayanan, 1996; Satyanarayanan, 1993; Stojmenovic, 2012), Cloud computing (Mell & Grance, 2011) and even, certain aspects of networks management (Sanaei, Abolfazli, Gani, & Hafeez, 2012). There are manifold approaches and definitions, yet in general they all have the same principle at their core which is to apply to mobile’s devices compute and storage processes techniques from cloud computing (Guan et al., 2011). Some examples of these definitions are provided below:

– (Sanaei et al., 2014) defines MCC as “a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage, and mobility to serve a multitude of mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle“.

- For (R. S. Chang et al., 2013) MCC represents “an emergent mobile cloud paradigm which leverage mobile computing, networking, and cloud computing to study mobile service models, develop mobile cloud infrastructures, platforms, and service applications for mobile clients. Its primary objective is to delivery location-aware mobile services with mobility to users based on scalable mobile cloud resources in networks, computers, storages, and mobile devices. Its goal is to deliver them with secure mobile cloud resources, service applications, and data using energy-efficient mobile cloud resources in a pay-as-you-use model“.
- (Kovachev et al., 2011) describes MCC as “a model for transparent elastic augmentation of mobile device capabilities via ubiquitous wireless access to cloud storage and computing resources, with context-aware dynamic adjusting of offloading in respect to change in operating conditions, while preserving available sensing and interactivity capabilities of mobile devices“.

MCC has been recognised as a beneficial technology for diverse fields of mobile applications in (Hoang et al., 2013). By means of concrete application examples, it details mobile applications that take advantage of MCC in areas which comprise: Mobile commerce, using MCC as the mechanism that allows handling mobility in operations such as “mobile transactions and payments and mobile messaging and mobile ticketing“ (Hoang et al., 2013); Mobile learning, applying MCC in order to overcome shortcomings in terms of devices costs, available network, computing and storage resources, as well as, access to limited educational resources; Mobile healthcare, in which MCC is employed as a tool that permits efficient access to information making specific emphasis in the necessary security and data protection aspects; Mobile gaming, enabling these kind of applications to access resource richer environments. In addition to these, MCC is considered admittedly useful for content sharing, searching services and collaborative applications (Hoang et al., 2013).

2.3.1 MCC Challenges

Challenges in the scope of Mobile Cloud fall into four groups: Firstly, we can mention the ones inherent to the use of mobile devices. These are related on the one hand to the limitations mobile devices in resources and battery and, on the other hand, those associated to the ability to perceive context and location. In addition to these, challenges related to the different approaches favoured to deal with these constraints such as Network Connectivity, Security and Off-loading & Application Partitioning are detailed. These are represented in Figure 3 taxonomy.

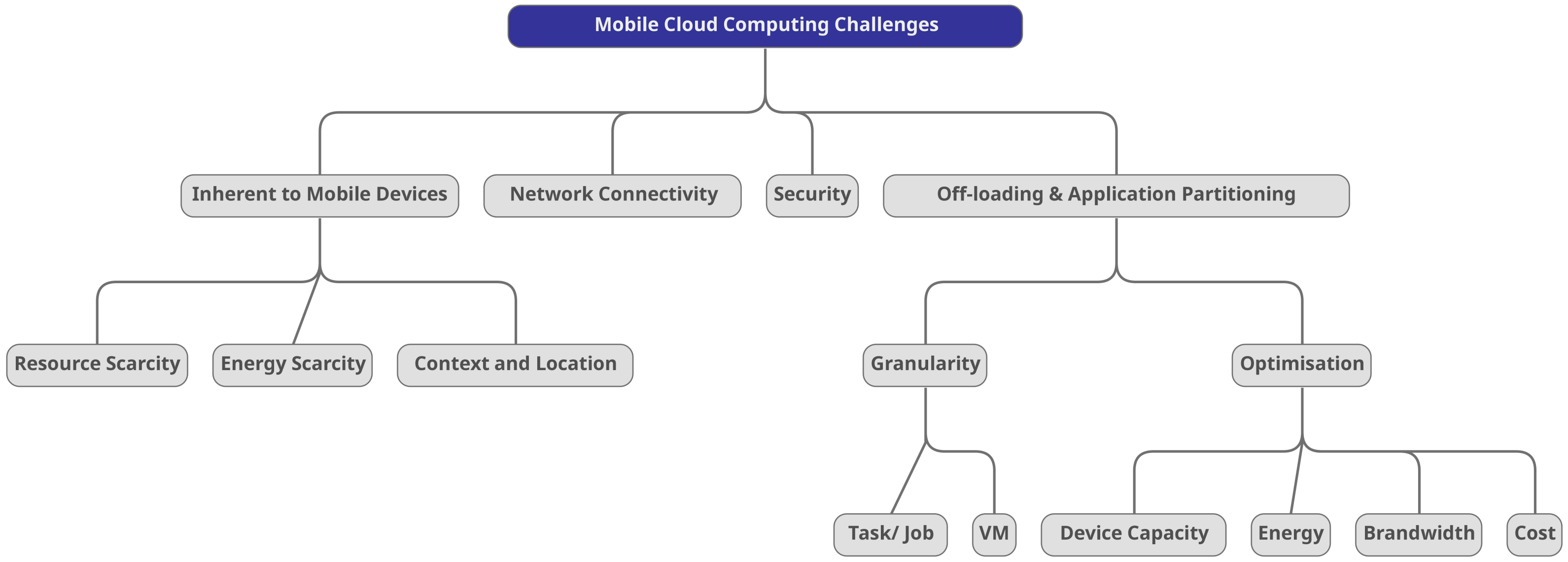


Figure 2.3: Mobile Cloud Computing Taxonomy.

2.3.1.1 Inherent Mobile Devices Challenges

Scarcity in Resource and Energy. While initial works in the area of Mobile Computing considered overcoming devices' limitations as the major issue for performance associated to resources hardware characteristics (J. Chang et al., 2005; Satyanarayanan, 1996). Authors today acknowledge the substantial augmentation of devices capacities in terms of CPU, memory, storage and others, such as the size of screen or associated sensors (Qi & Gani, 2012). Nevertheless, battery lifetime is still often perceived as a main roadblock due to the effect it has on mobile resource availability. With this regard, (Sanaei et al., 2014) acknowledges existing efforts to optimise, by means of applying offloading techniques, energy utilisation on the mobile device and the fact that this cannot always reduce energy. Other authors do not regard energy management and battery restriction as an issue for present-day Mobile Cloud Computing (Hoang et al., 2013; Zhong et al., 2012). Specifically (Hoang et al., 2013) presents MCC as a promising solution that can help to reduce power consumption in mobile devices without having to perform changes into the devices structure or hardware and taking advantage of software off-loading techniques.

Context and Location Guan in (Guan et al., 2011) underlines the fact that mobile devices allow the assessment of certain information from the device itself without the user's interaction. In the information that an be extracted without user interaction two types of contexts are identified: spatial context, related to location, position and proximity; as well as, social context, extracted from the user's or groups social interactions.(Sanaei et al., 2014) describes obstacles which radiate from the management of the social context owing to the exponential growth of this context due to multiple social interactions in diverse networks and social dynamism. The identified obstacles are related to storage, management and processing of these context data on resource constrained mobile devices.

2.3.1.2 Network Connectivity

The nomadic nature of Mobile devices and the fact that they rely on wireless networks as a challenge for Mobile Cloud in (Qi & Gani, 2012). Wireless networks are "characterised by low-bandwidth, intermittent and lower reliable network protocols is considered and as a factor that affects latency and therefore, unfavourably affects energy consumption and response time" (Sanaei et al., 2014). (Hoang et al., 2013) adds to this list availability issues and heterogeneity among different wireless networks interfaces applied. It explicitly cites as sources for availability issues, the aspects of traffic congestion, network failure and signal loss. In terms of heterogeneity, considers diversity on the radio access technologies, precisely determining the MCC needs with regards to continuous connectivity, on-demand scalability and energy efficiency. In order to address all these issues approaches based in local clouds or cloudlets have been developed. These are examined in detail in (Fernando et al., 2011; Gkatzikis & Koutsopoulos, 2013; Sanaei, Abolfazli, Gani, & Shiraz, 2012; Satyanarayanan et al., 2009). In this context, (Zhang et al., 2015) tackles the aspect of wireless intermittent connectivity among mobile devices and cloudlet environments, as a MCC key distinctive aspect. It develops a dynamic offloading algorithm which regards user's mobility patterns and connectivity to diverse geographically disperse cloudlets. In addition, it examines cloudlet's admission control policies based on user's distance to the cloudlet and cloudlet's coverage areas.

2.3.1.3 Security

(Fernando et al., 2013) concedes the fact that although many authors cite the need to provide the appropriate security context for Mobile Cloud Computing Services execution, the issue has been barely touched upon thus far. Specific analysis of Authentication and Privacy and Security issues are exhibited in (Alizadeh et al., 2016; Mollah et al., 2017).

(Shiraz et al., 2013) underlines that fact that "privacy measures are required to ensure execution of mobile applications in isolated and trustworthy environments while security procedures are necessary to protect against threads, mainly at network level". The analysis of privacy and security issues featured in (Gao et al., 2013) does not specifically concentrate on Mobile Cloud Computing issues, but rather reports well-known issues in the context of Cloud computing which involve the providers access to to user's virtual infrastructure or mobile physical threads associated with lending, lost or thieve of mobile devices or connection to public open network infrastructures.

Conversely, (Khan et al., 2013) provides a careful analysis and draws a detailed comparison of existing Mobile Cloud Computing security frameworks. Conclusions point to the fact that the majority of security frameworks overlook the trade-off between energy consumption and security requirements. It identifies hurdles which can be surmounted at the level of "data security, network security, data locality, data integrity, web application security, data segregation, data access, authentication, authorisation, data confidentiality, data breach issues, and various other factors"(Khan et al., 2013).

2.3.1.4 Off-loading & Application Partitioning

Many of Mobile Cloud Computing perspectives today revolve around application offloading and partitioning techniques in order to augment mobile device capacities (Fernando et al., 2013). Off-loading consist of moving part of the mobile computational workload to more resource-rich servers in heterogeneous Cloud models (Kumar et al., 2012). Research in Off-loading techniques (La & Kim, 2014) often contemplates a set of well-delimited phases which include:

Decision to offload Offloading has been viewed as a means to save energy and /or improve performance of mobile devices; however both feasibility and acquired benefits depend on factors such as available network link and amount of data to be transmitted. Considering the trade-off between offloading costs (commonly in terms of time,data transmission, economic costs, overall performance and energy) versus local processing costs, plays a key role in reaching offloading decisions.

Decision of application parts to off-load. The offloading granularity can be taken statically; this is predetermined in the mobile application execution flow at application development time; or dynamically, determined at runtime based on the execution context at a given time(Kumar et al., 2012; La & Kim, 2014). The granularity of

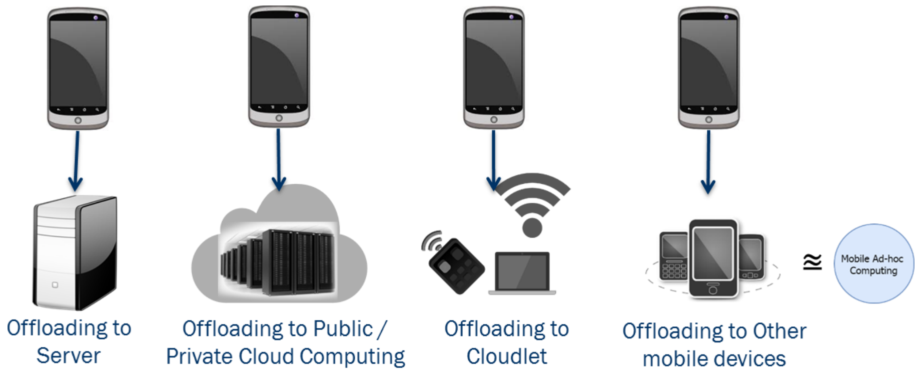

Offloading to Server
Offloading to Public / Private Cloud Computing
Offloading to Cloudlet
Offloading to Other mobile devices
≊
Mobile Ad-hoc Computing

Figure 2.4: Classification of Mobile Cloud Computing models.

parts of the application candidates to be offloaded ranges from offloading the complete application, so called coarse-grained methods; to fine-grained methods which consider specific application parts both at level of object, method, class, function and even tasks (Enzai & Tang, 2014).

Selection/Definition of infrastructures to off-load. Both specific framework analysis and literature surveys in the offloading topic consider this step, not as the selection of a computing infrastructure, but a specific server or surrogate selection in a pre-defined infrastructure (Enzai & Tang, 2014; Kumar et al., 2012; La & Kim, 2014). Very frequently, application partitioning mechanisms include mechanisms to optimize mobile device at the level of processor augmentation, energy savings, execution cost and bandwidth utilisation (Fernando et al., 2013; Gao et al., 2013; Qi & Gani, 2012; Shiraz et al., 2013).

2.3.2 MCC Models

Multiple papers tackle the issue of workload offloading from mobile devices to resource richer environments. These can be classified according to four main perspectives, depicted in Figure 2.4:

Off-loading to Server Under this classification we consider works that perform offloading to specific servers, which can be located or not in a Cloud environment. The pre-configured server has the mission to provide resources to alleviate mobile resource constraints and are by design limited to the initially defined server configuration. Analysis of articles that fit in this category is provided in Section 2.3.3.1.

Off-loading to Cloud Through use of private or public Cloud computing infrastructures, this classification considers the execution of off-loaded application parts often to virtual machine executing in a IaaS provider (Abolfazli et al., 2014). Works in this class are detailed in Section 2.3.3.2.

Off-loading to Cloudlet By means of using of Local computing infrastructures or Cloudlets (Satyanarayanan et al., 2009), works classified under this category (see Section 2.3.3.3) aim to reduce the overhead network latency derived from the use of distant traditional cloud infrastructures by using local infrastructures, cloudlets, closer to the mobile device location. (Satyanarayanan, Schuster, et al., 2015) further develops this concept by regarding the cloudlet as an intermediary step between the mobile device and the cloud, in a three-tier hierarchy in which the cloudlet is deemed to be a “Data Centre in a box“ set-up so as to “bring the cloud closer“ to the device (Satyanarayanan et al., 2014), therefore reducing latency. Conceptually, the idea of Cloudlet is the building block which sustains both Edge and Fog Computing. This is further developed in Section 2.5.

Off-loading to device Works in this category rely on using additional mobile devices capacity, commonly labelled as surrogates. These works are detailed in Section 2.3.3.4. Recently, under this standpoint a novel concept has been formulated through the development of Mobile Ad-hoc cloud (MAC) concept.

This classification serves us to structure the analysis of existing works in MCC in Section 2.3.3 presented hereby.

2.3.3 Analysis of Existing works in MCC

Drawing on the previously identified Mobile Cloud Computing Challenges and Models we define the taxonomy that is depicted in Figure 2.3. In the sections that follow we employ the Mobile Cloud Computing models for categorisation of existing works.

2.3.3.1 Approaches based on Off-loading to a Server

The model considers off-loading from Mobile device to a fixed external server, which can be or not hosted in a cloud environment.

2.3.3.1.1 MAUI, Making Smartphones Last Longer with Code Offload MAUI (Cuervo et al., 2010) targets reducing energy consumed by mobile devices while executing resource intensive applications. It offers fine-grained application off-loading at level of method. MAUI was defined by Microsoft research in 2010, being one of Mobile Cloud Computing precursor works, role in which is commonly referenced (Fernando et al., 2013; Hoang et al., 2013).

MAUI’s design goal is to overcome battery limitations of mobile devices. This work identifies the three most energy voracious categories of applications: video games and streaming, as well as, applications which focus on analysing data streams coming from mobile device’s sensors. By means of .Net code portability features, MAUI maintains two versions of the application to offload; one executing at the mobile device (equipped with Windows mobile in an ARM architecture) and one running at the server (x86 CPU). MAUI architecture presents components which execute both on the mobile device and the server. MAUI programming model, based C# and Microsoft .NET Common Language Runtime(CLR), allows developers to annotate methods as remotable. These annotated methods are instrumented at compilation time with the aim of allowing application state transfer when offloading. In order to minimize the amount of transfer of serialized application state, it uses an incremental approach, solely engaged in transmitting differences between mobile and remote states in different method invocations. At runtime the MAUI determines for all instrumented methods whether to execute it on the mobile device or remotely in the server before each execution.

Experimentation over MAUI's performance has been performed using four distinct applications, three of them pre-build and are currently running on Windows mobile phones (face-recognition, interactive video game and chess game applications), whereas a forth application was developed from scratch, a voice-based Spanish to English translator. For the first three, analysis of energy consumption and performance has been performed by comparing standalone execution of the application of the mobile application versus remote server execution based on a set of application metrics defined per each one of the mobile applications and considering several network conditions.

2.3.3.1.2 Cuckoo, a Computation Offloading Framework for Smartphones

Cuckoo framework (Kemp et al., 2012) targets application offloading for Android platform. Cuckoo design goals focus on providing a framework for mobile phones computation offload which allows energy consumption reduction, along with increased speed on execution of compute intensive applications for the Android mobile platform. The framework includes a programming model based on Java, conjoined with a Runtime environment. It allows mobile to server fine-grained method offloading which presents two optimization models: minimizing computation time and mobile device energy consumption. Server side execution requires any environment running a complete Java Virtual Machine, whether it is a dedicated server, a local cluster, a VM in a Cloud environment or any other capable environment.

To a certain extent, Cuckoo can be considered an analogous work in Java and Android platforms to previous .Net and Windows Mobile developments in MAUI. Having its main difference in the fact that Cuckoo permits to distinguish among code versions to be executed in the mobile device and the server (Kemp et al., 2012) bring new capabilities for user system configurability. This mechanism while being powerful in some cases has been identified as a drawback in previous mobile cloud computing surveys due to the need of providing two versions of the same application code (Fernando et al., 2013). By the use of Ibis High performance computing system Cuckoo acquires new capabilities for remote server configurability compared to other Server Off-loading existing works. Cuckoo permits dynamic deployment and interoperability with remote servers in diverse execution environments. This way, Cuckoo is able to consider Off-loading to Server model, in remote servers in diverse execution environments (dedicated server, a local cluster, a VM in a Cloud environment, etc.) in a transparent manner enabled by the interoperability layer that Ibis facilitates.

Cuckoo has been validated using two example applications: First, eyeDentify, an application which performs image pattern recognition, and simultaneously computing and memory intensive. eyeDentify was re-factored to use the Cuckoo programming environment. The second application was Photoshoot which is a distributed augmented reality mobile application.

2.3.3.2 Approaches based on Off-loading to Public / Private Cloud Computing

These approaches focus primarily on augmenting mobile device capabilities enabled by the use of more powerful resources in traditional date centre Clouds, both considering in private or public cloud environments and different levels of the Cloud stack (IaaS, PaaS and SaaS).

2.3.3.2.1 CloneCloud, Elastic Execution Between Mobile Device and Cloud CloneCloud presents a system which aspires to "augment mobile devices"(Chun & Maniatis, 2009) capabilities by means of offloading methods to device clones executed in a computational cloud. Vision was presented in (Chun & Maniatis, 2009) while its implementation is reported in (Chun et al., 2011). CloneCloud design goal is enable automatic transformation of mobile applications to profit from Cloud.

Significantly different from previous works Cuckoo and MAUI, CloneCloud is not dependent on the programmer in order to create application partitions. Instead, its purpose is to make application partitioning seamless and automatic for the programmer. In order to do so, it applies an offline method in which both static program analysis and dynamic program profiling are performed to define application partitions. Application partitions, in this case, are a choice of execution points where the application migrates a part of its execution and state from the device to the clone. Analyses can be executed considering several execution characteristics (considering CPU, network and energy consumption) leading to the creation of diverse partitions for the same application.

Static program analysis aims to identify "legal" choices whereby migration and re-integration execution between the device and the cloud are made possible. The system defines these migration points as method entry and exit points. "Legal" partitions are pre-computed and stored in a database. These are used in combination with dynamic application profiler to manage the distributed execution of the application across the mobile device and the device clone in the cloud.

CloneCloud is reliant on the concept of Application layer VMs, specifically in the Java VM available on Android devices, DalvikVM. This supports the migration of application pieces between the mobile device and the clone despite the differences in the CPU instruction set architectures, ARM and x86. Migration in CloneCloud is at level of thread and relies in a private Cloud environment based on VMware ESX.

2.3.3.2.2 ThinkAir ThinkAir (Kosta et al., 2011; Kosta et al., 2012) ambition is to simply developers tasks in migrating their applications to Cloud. In order to so, it presents a framework that facilitates method level computation offloading to Cloud environments.

Main novelties provided by ThinkAir adopt a more sophisticated use of Cloud computing environment directed at exploiting Cloud potential with regard to elasticity and scalability for Mobile Cloud benefit. ThinkAir provides on-demand cloud resource allocation in order to comply with specific requirements of mobile applications to offload at level of CPU and memory resources. Unlike CloneCloud, ThinkAir makes use of public commercial Cloud offerings and does not store predefined off loadable code partitions. ThinkAir relies instead on annotations provided by the developer to identify parts of code candidate to be off-loaded.

Furthermore, it enables parallelization by dynamically managing virtual infrastructure in the Cloud environment, therefore reducing both cloud server's side and overall application's execution time and energy consumption. The primary server in the ThinkAir architecture is a VM which clones of the mobile device replicating both data and applications (additional information about how these clones are synchronised and kept up-to-date is not present in the analysed works). This primary server is always set-up ready to be contacted by the mobile device. Other VMs distinct from the primary server, called secondary servers, are instantiated on-demand by the user. The primary server manages communications from the mobile, the life-cycle of these secondary servers, as well as task, allocation in case of parallelization; however no concrete details about this mechanism are readily available.

2.3.3.3 Approaches based on Off-loading to Cloudlets

(Satyanarayanan et al., 2009) formulated the concept of cloudlet as "a trusted, resource rich computer or cluster of computers that is well-connected to the internet and it is available for nearby mobile devices". In this concept, the mobile device acts as thin-client to services deployed in the cloudlet by means of VMs and that are accessible by wireless LAN.

As opposed to previously described approaches subject to distant servers or clouds, the overall aim of these models is to decrease the overhead network latency derived from the use of distant traditional cloud infrastructures. This is achieved by using local clouds or infrastructures, cloudlets, closer to the mobile device location. Proximity intends to ensure the predictability of the cloudlet's response time in order of magnitude of milliseconds. From this definition it derives the intrinsic linkage among cloudlet concept defined by Satyanarayanan in 2009 and posterior Edge and Fog computing definitions Bonomi et al., 2012; Shi et al., 2016. Generally speaking, the cloudlet vision defined by (Satyanarayanan et al., 2009) constructs scenarios where cloudlets shape "decentralised and widely disperse" computing infrastructures spread over the Internet. It is similar to enriching WIFI access points today with an easily deployable, long-lasting and self-managing "datacenter-in-a-box" resource. It is relevant to note, that (Satyanarayanan et al., 2009) circumscribe cloudlet to WLAN connectivity.

2.3.3.3.1 The Case for VM-Based Cloudlets in Mobile Computing While (Satyanarayanan et al., 2009) defined the concept of cloudlet it also provided an architecture in order to turn the concept into reality. Several authors, including recently developed Edge Computing area, do not dramatically differ in the concept articulation, but in its realisation: in MCC context developing the concept of workload offloading from mobile devices while in Edge and Fog contexts, responding to processing needs of IoT scenarios.. Some of these works are described in the sections that follow.

Overall design ambition of this work is to unveil potential of mobile computing as a mechanism which "seamlessly augments cognitive abilities of users using compute-intensive capabilities such as speech recognition, natural language processing, computer vision and graphics, machine learning, augmented reality, planning and decision-making". This ambition, articulated more than a decade ago, is today demonstrated ahead of its time and visionary by dint of existing Edge computing and Decentralised Cloud foreseen evolution. The architecture proposed in this paper (Satyanarayanan et al., 2009)is contingent upon "transient customisation of cloudlet infrastructure" in which, VMs are temporarily created, used and, afterwards, discarded from the cloudlet infrastructure in a dynamic manner and in order to provide a specific service to a mobile device located nearby. VM technology creates the necessary isolation and compatibility for cloudlet sustainability.

2.3.3.3.2 Gabriel Following the example described in previous work (Satyanarayanan et al., 2009), Gabriel (Ha et al., 2014) applies the Cloudlet concept to wearable devices in order to exploit its potential in Cognitive assistance processes. Gabriel relies on Cloudlets, with a view to reduce end-to-end latency while addressing battery and processing constraints of these wearable devices.

The concept is developed for the cognitive assistance scenarios employed include applications such as Face, Object, and Optical character recognition and Motion classifier. These require the interaction with wearable device (Google Glasses in this case) while placing high demands on both computation level capacity and latency requirements. The system design considers offloading from wearable devices to cloudlets and considering transiency among diverse cloudlets.

Also it takes the account that in cloudlets could have interactions to public/private clouds. Another notable aspect is that Cloudlets could be implemented with resource richer (not smartphones) movable devices such as laptops and netbooks. These bring different options for deployment of Gabriel framework itself, however are not developed in its architecture so to enable interoperability and seamless integration with a variety of execution environments. In Gabriel offloading normally occurs between the wearable device and the cloudlet located nearby. The wearable device discovers and associates to it.

In the absence of a cloudlet set-up, a proposed solution is to offload to cloud. This alternative workaround does not offer the cloudlet advantages incurring into the WAN latency and bandwidth issues in accessing distant Clouds initially avoided with cloudlets. In addition, this framework considers the situation of not having internet connection accessible, the an alternative solution proposed is the use of a mobile device or a laptop carried by the user as a direct device to offload. The vision proposed is that as smartphones increasingly come with more processing power, they can morph into viable offloading devices in the near future. Gabriel deploys each cognitive application in a separated VM in the cloudlet cluster. This cluster is also utilised in order to perform computational task parallelization required by the various applications.

2.3.3.4 Approaches based on Off-loading to other Mobile Devices

Hitherto, approaches regarding the mobile device part of the cloud are the least explored ones. The works under this classification significantly differ from previous MCC presented works. Both for Server, Public / Private Cloud and Cloudlet based MCC approaches, the mobile resource acts as a thin client and main motivation is to extend its limited capacities by acquiring additional capacity in resource richer environments. These resource richer environments are witnessed as infinite, in terms of the resources they can bring to the mobile application execution, neither presenting limitations in terms of battery and network instability under these approaches consideration. Here, the perspective changes. First, due to the consideration of mobile devices, which changes perception from just been seen as a thin client, to be considered a valid execution environment to complement capacity of other resources in its network. But also, from the view that the resource in which workload is offloaded presents the same volatility and instability characteristics than the resource which has originated the workload. The notable evolution which commenced a decade ago thanks to Moore's Law, has led to the increase of power and functionality of mobile phones (Triggs, 2020). Specialists expect this trend to continue up to a certain limit, as previously presented. Mobile battery is also an extended area of research both at industry and academia driven by requirements generated by the developments of wearable technologies.

Initial works driving to off-loading to additional mobile devices were presented in 2009-10 and coined under the MCC term. Starting in 2016 with (Yaqoob et al., 2016) some authors have used the term Mobile Ad-hoc Cloud computing to refer to similar approaches. These are presented in the upcoming Section 2.4.

2.3.3.4.1 Hyrax, cloud computing on mobile devices using Map Reduce

Hyrax (Marinelli, 2009) is "a platform derived from Hadoop that supports cloud computing on Android devices". Hyrax is constructed on the basis of a vision in which mobile computing is "an extension of cloud computing for which foundational hardware is at least partially made up of mobile devices" (Marinelli, 2009).

Hyrax's overall goal is to evaluate feasibility of mobile devices' hardware and network infrastructure to become a sort of cloud provider which uses local data and computational resources, analogous to traditional clouds. The envisaged type of clouds would be made of the opportunistic creation of networked connections of smartphones in which smartphones perform individual local computations in support of a larger system-wide objective which aggregates smartphone's local computations to meet goals of an overall application. The following principles guide the proposed mobile cloud computing infrastructure: "(a) each node is owned by different user; (b) each node is likely to be mobile; (c) each node is battery powered and (d) network topology is more dynamic" (Marinelli, 2009).

The following are understood as advantages of the approach: Avoidance of large data transfers to centralized remote services to perform computational jobs, instead of using local or vicinity capacity processing mobile multimedia and sensor data immediately; Enablement of more efficient access and sharing of data stored on smartphone devices through local area or peer-to-peer networks; As well as, distributed hardware ownership and maintenance.

Hyrax has based its work on porting Apache Hadoop 1.0 (Map Reduce)to be executed in the proposed Mobile Cloud infrastructure, rather than traditional commodity hardware as it is by definition intended. It is important to note that although mobile nodes are intended to be distributed, implementation of Hyrax utilizes an approach based on centralised management. Additionally, Hyrax to some extend oversimplifies the problem by relying solely on existing Hadoop fault tolerance mechanisms to overcome issues derived from use of mobile resources of the infrastructure. In addition, Hyrax does not take into account any of the application offloading and partitioning techniques for mobile application in previous works, instead it focuses on providing a already existing data analytics infrastructure in which worker nodes are mobile devices which offered functionality is equivalent to traditional clouds. Thereby, Hyrax is significantly divergent to previous MCC works however, completely in line with on-going and expected developments of Edge and Decentralised Cloud approaches, which almost a decade after still ambition similar goals.

2.3.3.4.2 A virtual cloud computing provider for mobile devices

Huerta-Canepa work on " virtual cloud computing provider for mobile devices" is described in (Huerta-Canepa & Lee, 2010). Its overall ambition is to overcome mobile resource limitations by simulating a cloud environment with other mobile resources available in the vicinity for situations in which connection to cloud is inaccessible or too

Works	MAUI, Making Smart-phones Last Longer with Code Offload	Cuckoo, a Computation Offloading Framework for Smart-phones	CloneCloud, Elastic Execution Between Mobile Device and Cloud	ThinkAir	The Case for VM-Based Cloudlets in Mobile Computing	Gabriel	Hyrax, cloud computing on mobile devices using Map Reduce	A virtual cloud computing provider for mobile devices
Resource Scarcity	X	X	X	X			X	X
Energy Scarcity	X	X	X	X				
Context and Location			X	X		X		X
Network	X		X	X	X	X	X	
Security								
Off-loading granularity	Method	JVM	JVM	Method	VM	VM	JVM	
Off-loading Optimisation	Energy	Energy	Energy	Cost		Latency		Energy, Time
MCC Model	Off. to Server	Off. to Server	Off. to Cloud	Off. to Cloud	Off. to Cloudlet	Off. to Cloudlet	Off. to Mobile Device	Off. to Mobile Device
Prog. Model / Language	Windows Mobile /.Net	Android / Java	Android / Java	Javascript / Java, C#		Android / Java	Java / Hadoop	PhoneMe, Java / Hadoop
Maturity	Prototype	Prototype	Prototype	Prototype	Architecture	Prototype	Prototype	Prototype
Use cases	Image Recog, Gamme	Image conv., Aug. Reality	Virus scan, image search, adver.	Image proc., aug. reality and video		Object recog., OCR	Video search and sharing	

Table 2.2: MCC Frameworks Feature Comparison

costly. This work is unique in MCC field by defining an infrastructure which is solely created out of mobile devices as an ad-hoc p2p cloud. The work provides remarkable inputs in relation to context management adapted to particularities of mobile devices. Specifically in this work partitioning of an application takes into account local resource availability and application resource needs. The selection of subrogates to which to offload and assign application partitions uses the amount and type of resources requested by the application execution and the amount of these resources available at candidate surrogates. This takes into account the mobile devices context defined as: social context, including relationships among users; location; and number devices in the vicinity. In addition, the works put forward a model for application partitioning that considers energy and time constraints; a failure prevention mechanism based on context; plus an adaptable trust mechanism that enables to open the platform to unknown nodes. (Huerta Cánepa, 2012) depicts the set of policies and processes involved in the proposed Context-aware offloading policy schema. The schema details the following steps: Monitoring, Partitioning, Selection of surrogate candidate and Offloading. Implementation of this architecture is reported to be based on Hadoop running on top of PhoneME. PhoneME is Sun Microsystems project to provide a JVM and Java ME reference implementation.

2.3.4 Features Comparison

Table 2.2 provides a feature comparison using the concepts defined in Mobile Cloud Computing Taxonomy, adding additional information about implementation status, maturity and use cases. In previous subsections we have analysed existing MCC works according to the MCC defined models for offloading: to server, cloud, cloudlet and mobile device.

Independently of this system architectural approach of all analysed studies, except of Hyrax, build on top of two main concepts: overall aim to optimise mobile device constrained resources and subsequent need for workload off-loading. From the analysed works only Gabriel (by use of wearable devices) is exploiting the MCC optimisation models and techniques for other available constrained devices than mobile devices, while these have huge potential for development in IoT and Decentralised Cloud context.

(Reinfurt et al., 2016) provides a systematic classification of IoT devices in form of patterns. In this, it is recognised that many IoT devices are mobile and are located off the power grid and recognises the need for these to optimise energy use, similarly to mobile devices addressed by MCC works. We claim that similarly to how MCC Cloudlet concept has recently been conceptually used in the development of Edge Computing concept. Tools and techniques for task off-loading and energy optimisation developed in the context of MCC will soon have to be employed in IoT and decentralised cloud context, together with the need of optimising IoT devices resources and taking advantage of all existing computing capabilities at the Edge.

According to this analysis we observe that the criteria most often used for optimisation of offloading decision are Energy and Execution time. The consideration of the Energy criteria is devoted to MCC traditional overall approach to preserve mobile devices resources. We foresee this need will remain with the application of MCC techniques to IoT context. It is noteworthy that so far consideration of security in MCC has been only marginally addressed. This is particularly critical while considering more advanced scenarios for MCC in Decentralised Cloud context, as mobile devices and, generally speaking IoT devices, act sources of data which will soon become critical to protect.

2.4 Mobile Ad-hoc Cloud Computing(MAC)

The concept of MAC has been only recently coined in (Yaqoob et al., 2016) in which is recognised as a novel area of research which is still in its infancy. In this work, MAC is understood as a new research domain that aims to "augment various mobile devices in terms of computing intensive tasks execution by leveraging heterogeneous resources of available devices in the local vicinity".

More concise definition is provided in (Yaqoob et al., 2017), "MAC enables the use of a multitude of proximate resource-rich mobile devices to provide computational services in the vicinity". Balasubramanian (Balasubramanian & Karmouch, 2017) further extends MAC definition by adding cooperation factors among participant mobile devices "A MAC is a pool of devices with high computational capabilities and is closer to the user. This low-cost computational environment is deployed over a network where all nodes cooperatively maintain the network". To the best of our knowledge, there is not yet a formal definition of MAC.

MAC motivation is to address situations in MCC for which connectivity to cloud environment is not feasible, such as absence or intermittent network connection (Yaqoob et al., 2016). This motivation was already the driver for MCC "offloading to mobile device" works, specifically central to (Huerta-Canepa & Lee, 2010), (Huerta Cánepa, 2012). It has to be noted that neither motivation nor MAC definitions denote substantial differences with previous MCC works instead; MAC appears as a novel term to denominate more recent works.

MAC is recognised to have its roots into MCC but also in opportunistic computing (Yaqoob et al., 2016). The definition of opportunistic computing (Conti & Kumar, 2010) provides additional considerations relevant for a system solely constituted by mobile devices. These are the concepts related to resource volatility and churn which can support further formal definition of MAC: "Opportunistic computing can be described as distributed computing with the caveats of intermittent connectivity and delay tolerance. Indeed, mobile and pervasive computing paradigms are also considered natural evolutions of traditional distributed computing. However, in mobile and pervasive computing systems, the disconnection or sleep device situations are treated as aberrations, while in opportunistic computing, opportunistic connectivity leads to accessing essential resources and information" (Conti & Kumar, 2010).

Kirby (Kirby et al., 2010) develops the desired features for ad-hoc clouds as: "An ad hoc cloud should be self-managing in terms of resilience, performance and balancing potentially conflicting policy goals. For resilience it should maintain service availability in the presence of membership churn and failure. For performance it should be self-optimizing, taking account of quality of service requirements. It should be acceptable to machine owners, by minimizing intrusiveness and supporting appropriate security and trust mechanisms" (Kirby et al., 2010).

Shila in (Shila et al., 2017) provides a distinction among mobile and static ad-hoc clouds. The latter, are including Edge computing and cloudlet environments and elaborating links among these novel cloud models and volunteer computing, as a way to optimise use of spare devices in mobile and other edge devices. Similar

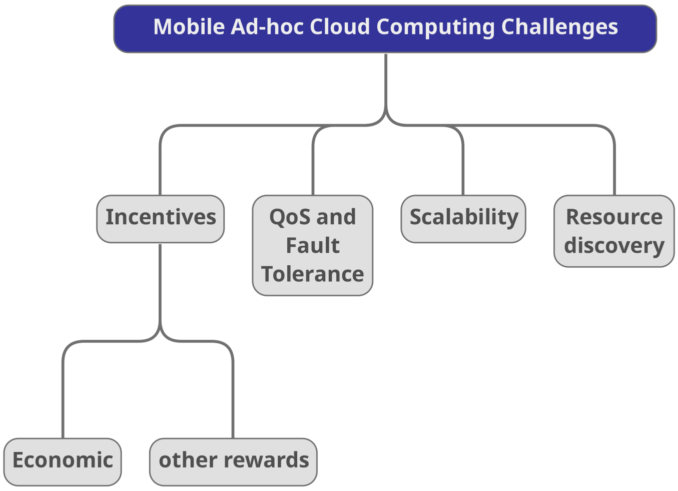


Figure 2.5: Mobile Ad-hoc Cloud Computing Taxonomy.

consideration is made by (Varghese & Buyya, 2018) considering this a as major trend for changing cloud infrastructures.

2.4.1 MAC Challenges

Challenges in MAC are inherit from MCC. However the consideration of Mobile devices as the single source of resources brings specific challenges to be considered in the context of MAC. These are depicted in Figure 2.5.

QoS and Fault tolerance As described in (Shiraz et al., 2013) mobile devices present specific characteristics with regards to resource availability (connectivity instability, battery limitation, communication bandwidth, or location variations). This makes it specifically relevant in the context of MAC the consideration of service management issues related to fault tolerance, availability and performance aspects. This work (Shiraz et al., 2013) highlights the importance of Fault tolerance mechanism considering the nature of mobile devices and its volatility. In addition, it remarks the need of incorporating additional aspects for QoS management in Mobile Cloud Computing which entail frequent loss of connectivity and low bandwidth and computational resources. Management of volatility of mobile resources and the availability issues derived from this fact is as a result the main identified challenge. Related work in the area for Mobile Cloud Computing based on Cloudlets recognizes as main problems limited and highly demand resources and mobility of users. (Yaqoob et al., 2016) reinforces the need of additional research of stability issues related to ad-hoc and distributed clouds. Similarly to some aspects of Service Management, few authors so far have analysed the problem of Admission control (the mechanisms to decide whether to accept or not a service to be executed on a cloud infrastructure). It is likely due to the fact that it is solely applicable in the context of MAC. In addition to this, expected autonomic nature of MAC, calls for management procedures which are self-managed. This autonomic management has to consider self-healing mechanisms so as to optimize provided QoS taking into account levels of fault tolerance and device's churn.

Scalability Mobile Ad-hoc Clouds could potentially sustain the provision of services over a massive number of resources with limited availability. Specifically on this aspect, authors such as (Hoang et al., 2013) only contemplate network QoS factors relevant for Mobile Cloud Computing, relying on local clouds and cloudlets as the simple solution for these issues. Particularly, (Hoang et al., 2013) identifies challenges in this area such as the distribution of processing, networking and storage capacity, in addition to the trade-offs management among cost and quality of experience. Both aspects, when extrapolated from MCC to MAC context, become critical in order to further develop this technology at scale.

Incentives Incentives for participation represent a key aspect for MAC and generally speaking to any volunteer computing system (Nov et al., 2010). Previous research in the area of volunteer Computing has demonstrated that temporal and voluntary resource donation is linked to different types of social, cultural and economic incentives with respect to service and data exchange, financial and collaboration aspirations. User's willingness to contribute is a key aspect for any contributory system. Although this area has been barely analysed in the MAC context, (Nov et al., 2010) presents motivations to contribute in the eScience area where most of the Volunteer computing work has been developed. Findings relate motivations mainly to "do good" and social contribution.

Resource Heterogeneity Generally speaking, Mobile Ad-hoc Clouds are particularly susceptible to the heterogeneity of devices. As resources set-in up the Ad-hoc Cloud environment are not confined to the data centre boundaries, but instead are extracted from sets of available resources, management frameworks have to consider device heterogeneity as key enabler.

Resource Discovery The dynamic behaviour of devices in terms or intermittent availability and its consequent possibility of resource churn in MAC, makes it necessary to take into account processes that permit the discovery of resources potentially available to join the MAC system. These processes for resource discovery in MAC have specific requirements with regard to the need to manage the environment dynamicity as well as to act in close relation with incentives mechanisms. Resource discovery methods for MAC could act in diverse logical network models including decentralised and centralised models. These methods could also consider diverse degrees of clustering and hierarchy depending on specific requirements in terms of scalability or fault-tolerance.

2.4.2 MAC Models

The analysis of existing literature in MAC, as well as general ad-hoc Cloud and opportunistic computing enables the definition of the following potential models for MAC (depicted in Figure 2.6.

Distributed Similarly to existing works in Contributory or Voluntary Computing, MAC could be based on the temporary resource donation, which is voluntarily

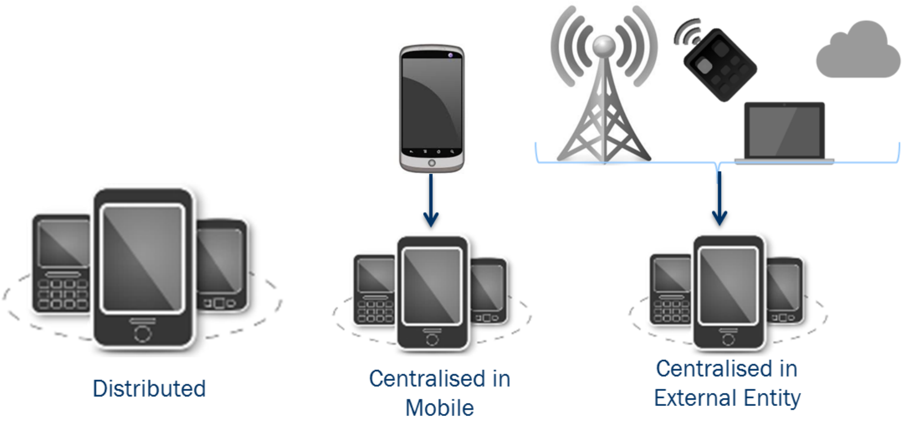


Figure 2.6: Classification of Mobile Ad-hoc Cloud Computing models.

contributed to set-up the ad-hoc mobile cloud. In this case, mobile resources would act at the same time as resource contributors and as resource users, by executing tasks or jobs in the MAC environment. Rooted in the contributory approach, mobile devices capacity is expected to be bestowed for an undetermined time period and can be disconnected at any time; as well as, it is decentralised and purely distributed, we can note the absence of any dedicated resource to its management.

Centralised in Mobile: These represent models in which one of the mobile devices taking part in the MAC does act as a Master for the Ad-hoc cluster, having the rest of devices as “surrogates“. This model is inherited directly from previous work in MCC in which the concept of surrogate was described.

Centralised in External This model considers external entities providing management features to the environment. This model categorises these cases in which the mobile device is deemed not to have sufficient resources to perform the ad-hoc cloud management, and other resource richer entities are selected as master. This model therefore only views mobile devices as “workers“ or “surrogates“. In the observed cases, the master election is a static decision, and not considering operational environments.

The taxonomy in Figure 2.5 for MAC challenges defines previously described characteristics for MAC. This taxonomy is used in Table 2.3 to classify existing works presented in next section.

2.4.3 Analysis of Existing works in MAC

This section presents a detailed analysis of previous works in MAC.

2.4.3.1 Dynamic Mobile Cloud Computing: Ad Hoc and Opportunistic Job Sharing

(Fernando et al., 2011; Fernando et al., 2012) elaborate on various aspects of dynamic mobile cloud computing framework. This framework aims to exploit the cloud when it is defined as “a cloud if local resources utilised to achieve a common goal in a distributed manner“. The aim of this work is to explore the feasibility of such local cloud in order to support mobility in mobile computing and associated concerns such as: sparseness and hazardousness of the resources in addition to limited energy source and connectivity. This framework aspires to respond to the following characteristics, being: “ a) Dynamic, in the way it can handle different resources and connectivity changes; b) Proactive, so that costs can be pre-estimated; c) Opportunistic, it makes use of resources as they are encountered; d) Cost-effective, in a manner that allows task distribution based on a cost model benefiting all participant resources.; e) Not limited to mobile devices, but able to manage low end devices such as sensors“ (Fernando et al., 2012). As opposed to previous works analysed it considers parallel task execution using simultaneously diverse surrogate devices, however details on the approach to do so, are not provided.

The system architecture is organised in a cluster, in which one of the end-user devices acts as master, with a set of associated surrogate mobile devices performing slave tasks. Although authors intention in this set of works is to handle diverse end-user devices in the IoT spectrum, experimentation performed focus on PCs and mobile devices.

2.4.3.2 MOCCA, A mobile Cellular Cloud Architecture

MoCCA (Mishra & Masson, 2013) is described as a “cellular Cloud architecture for building mobile clouds using small-footprint microservers running on cell phones“. MoCCA’s objective is to avoid costs incurred in the set-up of traditional cloud data centres by taking advantage of already existing infrastructure elements. MoCCA advances the idea of benefiting from already existing telecommunications and networking elements in GSM cellular systems in order to build its architecture. Thus, the resources included in the architecture are smartphones, base stations, base stations controllers and mobile switching centres. Five aspects are identified as main concerns for Mobile Cloud design in this work: (1) Connectivity, bandwidth limitation, lack of direct connectivity among mobile devices, and the need to consider frequent network disconnections; (2) Computational limitation, due to mobile device resource limitations; (3) Churn, due to users mobility and devices’ volatility; (4) Energy, with the approach of conserving energy in the mobile device; (5) and Incentives to users to participate with their mobile device in the Mobile Cloud infrastructure. The architecture proposed consists of two main parts: MoCCA Client and MoCCA manager. The latter, provides centralised control from the base station controller resource and executes from the base station. The MoCCA client is powered with an execution sandbox with stores function codes to be executed, in addition to Client controller and Audit and logging functions.

MoCCA has been evaluated with computer bound applications. The only notable issue regarding Mobile Cloud Design which has been evaluated is Energy consumption from data reception and transmission. The remaining concerns (connectivity, churn, computational limitations and incentives) have yet to be considered in their architectural design and evaluation.

MoCCA’s differentiation aspect from previous MCC and MAC works is that MoCCA adopts GSM cellular network infrastructure as part of the MAC. This fixed infrastructure acts as the MAC coordinator. The idea of using network equipment as part of the computing infrastructure at the Edge is now intensively examined as part of Edge computing research.

2.4.3.3 Ad-hoc Cloud as a Service

(Zaghdoudi et al., 2017; Zaghdoudi et al., 2015) present a "protocol an a preliminary architecture for the deployment of Ad-hoc MCC on top of MANET Ad-hoc networks". It addresses the need of solving dependence of mobile devices with remote cloud by exploiting capacities of surrounding devices. In these, two main entities are considered: Providers, offering nodes acting as resource providers; and Customers, that request resources. The resultant protocol, C-Protocol "governs the interaction and the communication among Ad-hoc nodes and provides the dynamic management of providers and customers" (Zaghdoudi et al., 2017; Zaghdoudi et al., 2015).

The proposed architecture presents two layers: The C-protocol layer, a meta-layer intended to provide required network services; The CloudSim layer: a simulation layer using CloudSim simulation aiming to model and simulate a data centre environment and virtualised infrastructure based on mobile devices. The protocol considers adding and members departure processes, as well as Customer inclusion. No specific details about potential implementation of these, such as mechanisms for customer or provider registry or fault tolerance, monitoring mechanisms, workload considerations are constituent of this work.

The originality of this work lies in the joint consideration of network and compute aspects (although the latter are not developed with full details) and specifically the joint consideration of MAC and spontaneous networks such as MANETs. Initial experimentation has used 9 laptops equipped with Windows and Linux operating systems simulating mobile nodes connected over WIFI Adapters. The objective of the experimentation was to analyse the feasibility of three metrics: Time to set-up, Time for customer to join and Time to add a provider in the MAC system.

2.4.3.4 MobiCloud

MobiCloud (Hammam & Senbel, 2014) is presented as a "reliable collaborative mobilecloud management system". which enables the efficient and collaborative use of available mobile phone resources. This work coins the novel term mobilecloud in order to refer to the overall objective of exploitation of computing capacities of mobile and field devices even when no internet connectivity is available. The detailed architecture comprises two types of nodes: a field control node, named Cloud Agent and participant nodes (mobile or field nodes). The Cloud Agent is the agent requesting to form a Cloud and provides centralized Cloud controller functionalities. When an application is submitted to the CloudAgent it localizes from the set of available registered resources those which match the defined application requirements.

The reliability of the resources is assessed by means of a Trust management system which takes into account QoS offered by the participant nodes.

Available nodes are priorised resting on: first, number of available CPUs, and then on, time employed in data transmission. The differentiation aspect of this work is declared to rely on the node reliability mechanism and its reputation system based on user's feedback. Other works in the past (Huerta Cánepa, 2012; Huerta-Canepa & Lee, 2010) have provided fully automated processes built upon collection of historical node behaviour. Evaluation of MobiCloud has been performed using an extension of CloudSim simulation and has included the homogeneous computing capacities of nodes, complete availability of all nodes and uniform distribution of connectivity speed. The metric evaluated in simulation has been application execution time.

2.4.3.5 mClouds

mClouds (Miluzzo et al., 2012) build on the vision future mobile devices will become core components of mobile cloud computing architectures and not just thin clients to cloud environments. It particularly elaborates in the assumption that "computation and memory will likely increase considerably while battery and network capacity will not grow at the same pace" with the overall aim of reducing saturation of cellular data networks (Miluzzo et al., 2012). The initial analysis of mClouds architecture is divided into two main aspects: distributed mCloud processing and specific resource discovery procedures; and Incentives management.

Distributed mCloud processing architecture comprises mDevs, mobile devices able to execute mTasks. An mTask is a part of a larger computing task that can be parallelised. Distributed mCloud processing advocates on a simple initial principle, execute locally whenever possible. For cases for which this is not feasible due to lack of resources in the task originator device (master), look for mobile resources to form a mCloud.

This work presents the interesting novelty of elaborating in incentives strategies for mCloud participation. Incentives mechanisms consider the mobile carrier as clearing house, in order to reduce network congestions at certain locations. mClouds is conceived as a commentary approach to previous MAC and MCC works developing tools and mechanisms for application partitioning and offloading.

2.4.3.6 Aura

Aura (Hasan et al., 2015) aims at providing IoT based Cloud computing models in which mobile devices, acting as clients, are able to offload computation tasks to nearby IoT devices. Therefore, creating ad-hoc cloud out of low power IoT devices in a specific location to which proximal mobile devices can outsource computation tasks.

Motivation for this approach is twofold: firstly, in order to provide a local computation environment that reduces latency and keeps data privacy; and secondly, with the intention of avoiding the costs of deploying data centre clouds located near to the client. The use of Aura is exemplified in a Smart building scenario. Compared to previous works, Aura brings the innovation of already considering IoT devices in the Smart Building scenario as part of the MAC system considering them not only as data sources but as valid MAC resources, depending on their specific characteristics.

A proof of concept of the approach has been developed for Aura with an Android mobile application for Mobile Agent implementation; Controller as a Desktop Java application; and IoT devices capabilities represented by MapReduce ported to Contiki IoT platform. A number of IoT devices were simulated with Cooja framework. The experimentation was conducted by offloading wordcount implemented in MapReduce for optimisation of execution time.

2.4.4 Features Comparison

Table 2.3 provides a feature comparison using the concepts outlined in Mobile Ad-hoc Cloud Computing Taxonomy, introducing additional information about implementation status, maturity and use cases. At model level, we observe that so far the preferred model in existing works is to provide ad-hoc mobile cloud functionality from an external entity. This external entity in the analyses works is offered from Cloud environments, IoT devices and even, Network equipment. Centralised management in a mobile that manages ad-hoc clouds in other mobiles acting as "surrogates" is also a model which is gaining popularity emerging together with the increment of computing capacities of mobile devices. In both cases, there is a single point of failure for these architectures due to centralised design. Complete decentralisation and distribution has been an area of study in Volunteer and P2P systems in the past. This model of management is feasible and performant, as demonstrated in previous volunteer and p2p computing works, and provides interesting features at levels of mechanism for handling complexity of volatile resources, high scalability and self-management foreseen as specifically of interest for the evolution of mobile and steady ad-hoc clouds.

Until now only some specific MAC works have gone beyond the smartphone as main source of resources. Tools such as Aura describe initial steps towards the inclusion of IoT in mobile ad-hoc architectures. In our view, future evolution of MAC in Decentralised Cloud will not only reinforce existing initial works addressing IoT devices but to focus its evolution on them, as available processing capacities in heterogeneous devices growth. The exceptional forecasted development on the number and complexity of IoT connected devices will force this evolution as a mandatory requirement. The overall computing available at the Edge of the network is growing in number of devices but also in their capacity, coming from diverse and heterogeneous sources in form of robots, drones and autonomous vehicles. At level of challenges addressed we observe consideration of location is yet to be addressed, as well as, QoS and massive scalability necessary in this context. As observable in Table 2.3, yet the attention to hardware heterogeneity in the management of MAC is not a reality in any of the analysed MAC works. This is in our view, another clear source of evolution in the coming years for MAC and Decentralised Cloud in general. Over the last decades, Moore's law has enabled the substantial computing capacity growth in microprocessors.

Recently, we are witnessing the emergence of built-in artificial intelligence processing units into mobile devices which are expected to soon power many other IoT devices.

The foreseen slow down progress expected for Moore's Law in the future will call for taking better advantage of all available compute resources, therefore forcing MAC systems and Decentralised Cloud to manage heterogeneity so to exploit all available compute sources.

2.5 Edge Computing

Cloud computing today has transformed into a massive centralised

infrastructure acting as a central keystone for compute power, storage, process, integration, and decision making in numerous environments. Following the pattern we have thus far in the existing IoT set-ups, generated sensor data would have to

Works	Dynamic Mobile Cloud Computing: Ad Hoc and Opportunistic Job Sharing	MOCCA, A mobile Cellular Cloud Architecture	Ad-hoc Cloud as a Service	MobiCloud	mClouds	Aura
QoS and Fault Tolerance						
Scalability						
Incentives	Economic				Economic	
Resource Heterogeneity						
Resource Discovery			X		X	X
MAC model	Centralised in Mobile	Centralised in External Entity	Centralised in Mobile	Centralised in External Entity	Centralised in Mobile	Centralised in External Entity
Prog. Env.	Java	Java				Android / Java
Maturity	Prototype	Prototype	Simulation	Simulation	Model	Prototype
Use Cases	Distributed Mandelbrot Set Generation	Cholesky Decomposition Fast Fourier Transform				MapReduce Word count

Table 2.3: MAC Frameworks Feature Comparison

be transmitted over the network in order to be centralised, processed and analysed in the Cloud.

With a view to cope with IoT proliferation this scenario has to change, providing an infrastructure which takes into account billions of devices connected at the edge and offering more rapid processing and decision making. Therefore, the idea under Edge Computing is to enable the decentralisation of the cloud, approximating computation and storage to the sources, at the edge of the network: avoiding unessential network transmission and getting data and computation at the right place and right time.

Edge computing paradigm (Bonomi et al., 2012) "extends Cloud Computing to the Edge of the network". Both Edge and Cloud manage computation, network and storage resources applying similar techniques such as virtualisation and multi-tenancy (Bonomi et al., 2014). However, Edge computing's main aim is to address the latency issues detected in the application of Cloud Computing to large IoT scenarios (Yannuzzi et al., 2014).

Edge computing is defined by Shi (Shi et al., 2016) as: "Edge computing refers to the enabling technologies allowing computation to be performed at the edge of the network, on downstream data on behalf of cloud services and upstream data on behalf of IoT services.". This work frames Edge "as any computing and network resources along the path between data sources and cloud data centres" (Shi et al., 2016).

The term Fog Computing has been instead proposed by Cisco (Cisco Systems, 2016): "Fog Computing is a paradigm that extends Cloud computing and services to the edge of the network. Similar to Cloud, Fog provides data, compute, storage, and application services to end-users. The distinguishing Fog characteristics are its proximity to end-users, its dense geographical distribution, and its support for mobility". Also from CISCO, Bonomi's in its introductory work "Fog Computing and its role on the internet of Things" (Bonomi et al., 2012) proposes the following definition for Fog computing: "Fog Computing is a highly virtualised platform that provides compute, storage, and networking services between end devices and traditional Cloud Computing Data Centres, typically, but not exclusively located at the edge of network".

The definition provided by (Vaquero & Rodero-Merino, 2014) does not confine technology choices to virtualisation and adds a cooperation factor: "Fog computing is a scenario where a huge number of heterogeneous (wireless and sometimes autonomous) ubiquitous and decentralised devices communicate and potentially cooperate among them and with the network to perform storage and processing tasks without the intervention of third-parties. These tasks can be for supporting basic network functions or new services and applications that run in a sandboxed environment. Users leasing part of their devices to host these services get incentives for doing so".

Overall Bonomi's approach refers to the fact that IoT platforms will, in the short term generate large volumes of data, which will stand in need of analytics platforms to be geo-distributed; in a way of "moving the processing to the data". Therefore, creating the need for "distributed intelligent platform at the Edge Computing that manages distributed compute, networking and storage resources".

Edge and Fog Computing are not devised as competitors to Cloud; quite the contrary, it is conceived as the perfect ally for use cases and applications for which traditional Cloud Computing is not sufficient. Further extended in (Bonomi et al., 2014) the Edge vision was created to "address applications and services that do not fit well the paradigm of the Cloud“. Edge approach is very much aligned with Mobile Cloud Computing works, as recognised in (Garcia Lopez et al., 2015; Satyanarayanan, Schuster, et al., 2015; Yannuzzi et al., 2014). When observing evolution of the market, again, the major Cloud provider, Amazon Web Services (AWS) appears as a pioneer in the area of Edge computing by its AWS Greengrass product (“Amazon Web Services Greengrass”, 2017). This has recently being followed by MS Azure Edge platform (“Azure IoT Edge”, 2017), as will be presented in Section 2.5.3.2.

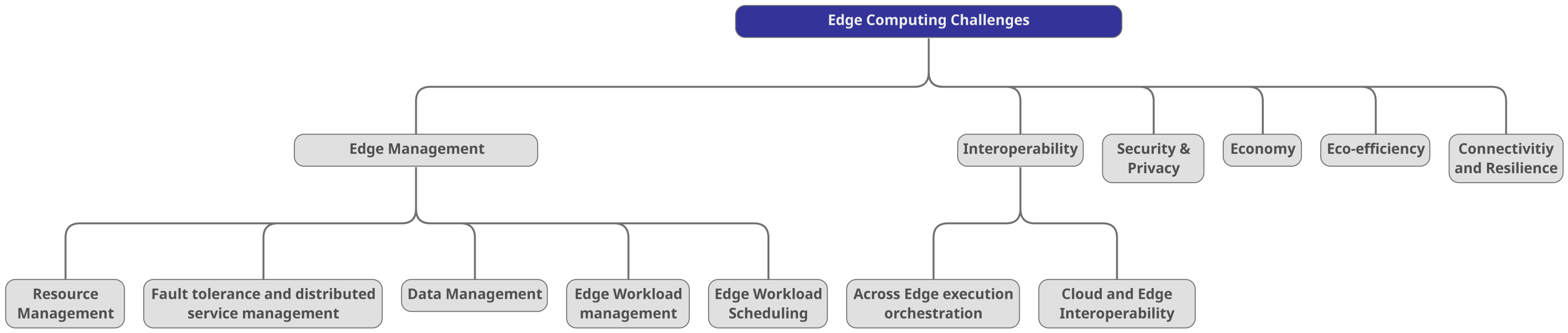

Edge Computing Challenges
Edge Management
Interoperability
Security & Privacy
Economy
Eco-efficiency
Connectivitiy and Resilience
Resource Management
Fault tolerance and distributed service management
Data Management
Edge Workload management
Edge Workload Scheduling
Across Edge execution orchestration
Cloud and Edge Interoperability

Figure 2.7: Edge Computing Taxonomy.

2.5.1 Edge Computing Challenges

Below we elaborate on a series of Edge computing challenges and characteristics necessary to be developed in order to make the described concepts a reality. These are also represented in Figure 2.7.

2.5.1.1 Edge management

Resource Management Management of massive number of small diverse devices and sensors in Edge computing set-ups will necessitate new management styles, potentially decentralised and able to scale to degrees that nowadays are unprecedented in existing architectures (Vaquero & Rodero-Merino, 2014).

Fault tolerance **and distributed service management**. Resource heterogeneity, scalability, fault tolerance, availability and performance are service management aspects still to be addressed in Edge computing. These are of specific interest due to the nature of devices and their volatility in addition to this need of including supplementary aspects for QoS management, scalability and heterogeneity in resources, integration of special devices including hardware accelerators, FPGAs and GPUs.

Workload management Encapsulation of edge workloads on top of Edge systems will have to accommodate diverse workload typologies and the different processors types where these workloads can be computed, for which the final encapsulation solution may vary. A system able to deal with various encapsulation approaches will be required to prepare the workloads depending on the final execution environment. Mechanisms adapted to balance between high-performance processor and low power processor according to the final objectives of the workload should shortly be taken into consideration.

Workload Scheduling Workload or task scheduling in Edge and Fog computing has to take into account specificities of the Edge devices, such as energy constraints and QoS (usually in terms of latency optimisation). Diverse works have recently analysed the problem from diverse perspectives. Some works handle it as a joint optimisation problem among the Edge and Cloud resources: with the aim of addressing different application classes (Bittencourt et al., 2017); focusing on performance and cost optimisation (Pham & Huh, 2016); and aiming to optimise delay and power consumption (Deng et al., 2016). Others, such as Bitam (Bitam et al., 2018) devises it with the innovative approach of bio-inspired optimization.

Data management Hitherto, data intensive applications have been the key motivation spreading Edge computing need. Novel systems able to manage data scattered on an Edge heterogeneous and distributed environment needs to deal with the intricacies of the underlying complex infrastructure composed by smart devices, sensors, as well as traditional computing nodes. Conversely, developers must focus on establishing the relevant data, which is the necessary to keep, their format and quality, and how to process them, avoiding details concerning how to gather data, where to store or process them ("DITAS, Data-intensive applications improvement by moving data in mixed cloud/fog environments", 2017).

2.5.1.2 Edge Interoperability

Orchestration across Edge. Edge set-ups are envisaged to be spread covering wide geographic areas. For serving applications and services that make use of these distributed set-ups, mechanisms for deployment, provisioning, placement and scaling service instances across execution zones in the distributed Edge set-ups are necessary.

Interoperability with Edge and Cloud. Current status of Edge computing developments very much relies on specific vendor solutions. In order for these to interoperate among them and with traditional clouds, new standards would have to appear to manage the expected scale of edge set-ups and the interoperability of devices and sensors.

2.5.1.3 Economy

Cloud computing has been recognised as a bridge between distributed systems and economics. Cloud computing providers offer a number of services to users using pricing schemes relying on incurred resource consumption. Existing commercial Edge computing environments, although based on simple devices, are being deployed in complex economic models which combine pay per use and licensed based (see Section 2.5.3.2). Further investigation is vital for designing models ready to cope with challenges and diversities of existing Edge Cloud models.

2.5.1.4 Eco-efficiency

A significant challenge associated with Edge deployments is potent power provisioning for locally deployed infrastructure. While substantial advances have been made for data centre and Cloud Energy Efficiency, particular challenges remain in order to optimise energy consumption and availability of energy sources in edge environments. It is important to note that the diversity of resources and potential energy sources potentially involved in Edge computing provisioning add additional challenges to this matter. Another environmental concern linked with Edge computing is the lifecycle of all devices which are disseminated. Approaches for device management of objects that incorporate a battery and matter potentially harmful to the environment would have to be considered in the future.

2.5.1.5 Security and privacy

Edge computing, similarly to traditional cloud, is viewed as multi-tenant, and therefore actual set-ups will require of concrete isolation mechanisms so as to avoid security and privacy concerns.

2.5.1.6 Connectivity and Resilience

Resiliency is also a core characteristic required for Edge computing set-ups, notably for mission critical IoT applications. There is the overall need for these applications to continue providing their services from the Edge even when network links to Cloud are down or seriously overloaded. Diverse techniques are being studied in order to provide lack of connectivity resilience capability, among them fault tolerance systems across diverse Edge installations in a close location and techniques for

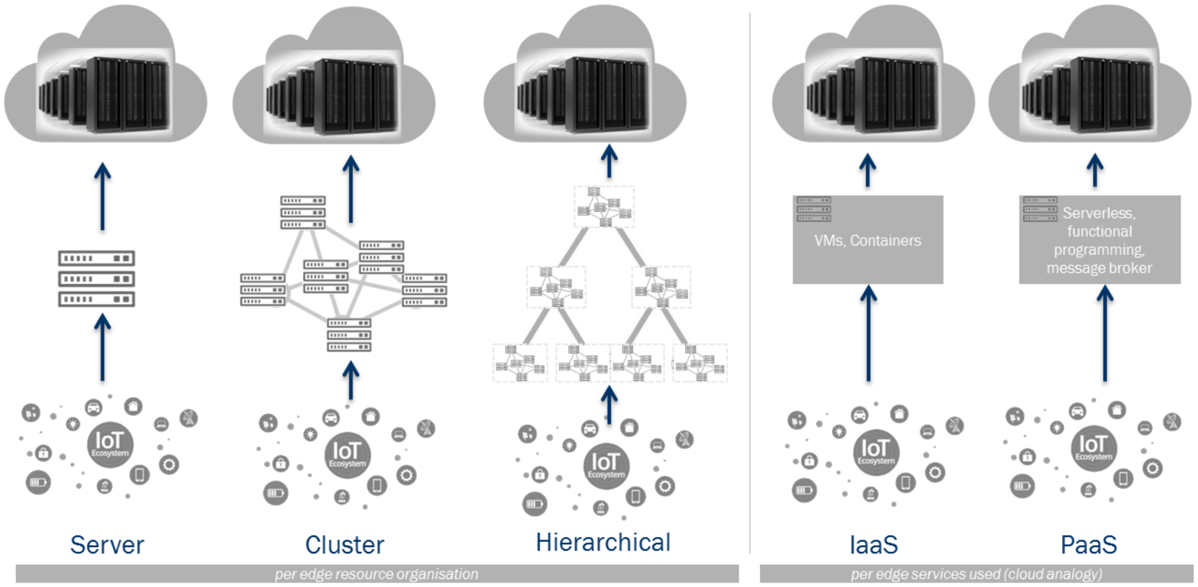

VMs, Containers
Serverless, functional programming, message broker
IoT Ecosystem
Server
Cluster
Hierarchical
IaaS
PaaS
per edge resource organisation
per edge services used (cloud analogy)

Figure 2.8: Classification of Edge Computing models.

unconnected Edge limited operation.In addition to this, it is important to remark that a wide area of research exist (which is considered out of scope in this Thesis) addressing specific research challenges in future networks in relation to 5G and SDN/NFV and Edge computing from Telco perspective..

2.5.2 Edge Computing Models

Existing approaches to Edge Computing can be classified according to the following criteria. These different Edge models are depicted in Figure 2.8.

Edge Server approaches are those which consider the Edge environment a device, which we name server, that provides computing and storage capacities to a series of Edge sensors and other resource poorer devices that are connected to it in a locally close environment. These so-called “servers“ can be represented by devices which range from Raspberry Pis to servers, but so can devices such as connected cars, network equipment, or other rich smart IoT devices, as long as they provide a minimum computing and storage capacity. With this regards project HEADS has provided the following classification (HEADS Project, 2016) among devices which comprises: Tiny, Small and Large. These can be described as: Tiny: Very limited devices (8 and 16 bit micro controllers with less than 64kB program memory and 4kB of data memory). Example of this type of device is Arduino UNO; Small: Devices with a specific OS and restricted hardware characteristics (less than 128kB program memory and less than 64kB data memory); Large: Devices supporting general purpose OS. Examples of these are: Raspberry PI and Android. Edge Server approaches are the ones we encounter today in commercial products such as Amazon Greengrass, and Azure IoT Edge using the so called “Large“ devices. Also from equipment vendors such as Dell we found pure and traditional servers to be deployed (Dell PowerEdge Series).

Edge Cluster approaches are those considering sets of the previously so-called server devices that are coordinated by a node considered the cluster master. This clustered approach could be considered at diverse granularity levels in view of the nature of the proposed scenario and the compute/storage requirements. An exemplification of the concept could be performed in a smart home scenario considering that all “smart“ enough devices, servers, aggregate their capacity in order to provide compute/storage capacities to other more resource constrained home appliances.

Hierarchical classification considers layered configurations of Edge clusters. The layered approach could be construed according to diverse criterion. These include: layered approaches based on increasingly resources capabilities or location (aggregating at diverse levels i.e. resources at home, neighbourhood and smart city).

Making an analogy with existing Cloud offerings we could also classify Edge approaches as:

IaaS Those offering Compute and Storage capacities in diverse virtualisation formats including VMs and containers.

PaaS offering access to programming environments (the more advanced ones providing Serverless and functional programming environments such as AWS Lambda), ML tool-sets as well as software capabilities such as message brokers to facilitate development of applications on top of these environments.

2.5.3 Analysis of Existing works in Edge Computing

2.5.3.1 Existing works in Research environment

2.5.3.1.1 Fog computing, a platform for internet of things and analytics Fog computing was introduced in (Bonomi et al., 2012). (Bonomi et al., 2014) enhances this initial work in order to propose a Fog architecture including new requirements that IoT scenarios pose on Fog Computing with regard to big data analytics. Overall the approach is based on the fact that IoT platforms will, in the short term generate large volumes of data, requiring of analytics platforms to be geo-distributed; in a way that "moving the processing to the data". Thus, creating the need for "distributed intelligent platform at the Edge (Fog Computing) that manages distributed compute, networking and storage resources".

The proposed high level architecture has the following three key objectives: transparency, heterogeneity (of both resources and applications) and distributed orchestration. Transparency refers to the ability to manage in an abstract manner resource elements at edge, cloud and network. Heterogeneity is related to the diversity of previously mentioned resources but also to need of supporting multiple applications from diverse sectors. Finally, orchestration has to be driven by defined policies that consider scalability at local and global levels. Bonomi's work coined the term Fog computing. Although cloudlet concept is not specifically referenced in this work diverse authors have recognised its direct links in spite of different motivation for decentralisation: IoT infrastructure scalability, for fog computing; versus mobile applications performance for cloudlet(Satyanarayanan, 2017).

2.5.3.1.2 ANGELS for distributed analytics in IoT ANGELS stands for "Available Network Gateways in Edge Locations for Sensors" and it is presented in (Mukherjee et al., 2014). ANGELS presents on-going work and explores the idea of using smart edge devices (sensor gateways, personal laptops, play-stations, and smartphones) as envisaged in the Fog paradigm in order to perform parallel execution of data processing jobs in IoT, using idle capability of these devices. Overall ambition of this work is to take advantage of unused computing capacity at the edge of the network at homes and around these, in order to cope with demands for data analytics computation expected from the development of IoT systems. This architecture targets the class of applications which presents a data parallelization approach: namely, applications capable of processing data divisible into several subsets, partitions, which can be processed in parallel, similar to the MapReduce approach.

So far this architecture is working under the assumption that edge devices are available. Next steps detail the consideration of dynamic availability patterns of edge devices. A new element of ANGELS is the contributory/volunteer computing element it brings, by means of taking advantage of idle of smart edge devices. However, it recognises that due to Edge devices resources constraints and their mobility, edge devices will have to be complemented with fully powered resource richer servers.

2.5.3.1.3 Mobile Fog Mobile Fog (Hong et al., 2013) presents a "high level programming model", or a PaaS, "for applications that are geographically distributed, large scale and sensitive to latency"(Hong et al., 2013). Authors position this work as an alternative for Cloud PaaS which focus on web applications, by developing a solution that specifically addresses needs of data analytics for IoT.

The objectives of Mobile Fog Programming model are: to ease application development on highly distributed heterogeneous devices; and to support scalability both at Edge and Cloud. In this work Edge devices resources considered go beyond typical mobile phones, but also considering connected vehicles. In Mobile Fog an application is a group of distributed processes which have to be assigned into a set of disperse computing instances in edge devices, and fog or cloud environments. It is considered a physical hierarchy of devices in which a process in an edge device is a leaf, and processes in the edge cloud are intermediate nodes and processes in cloud are considered the root. In this set-up each Mobile Fog Node manages workload from a specific geo-spatial location. Scalability management is performed through scaling policies that determine behaviour reliant on monitoring metrics such as CPU or bandwidth. Scalability mechanisms address instances at the same network level. Further work it is expected in runtime systems implementation and process placement algorithms. This work recognises to be complementary to fog architecture presented by (Bonomi et al., 2014; Bonomi et al., 2012) by addressing on programmability aspects in Fog context.

2.5.3.1.4 Nebula Nebula (Chandra et al., 2013; Jonathan et al., 2017; Ryden et al., 2014) is presented as a "dispersed edge cloud infrastructure that explores the use of voluntary resources for both computation and data storage". Nebula motivations are: to reduce data upload to traditional clouds by offering disperse computing environments and to eliminate overhead of virtual infrastructure instantiation in Clouds. Nebula relies on volunteer computing mechanisms as tools that allow widely distributed environment. While supporting distributed data intensive applications, Nebula deems data movement and origination problems, considering geographical distributed execution. In order to do so, scheduling of computing has to take into account execution time, but also data movement costs. Nebula system architecture includes the use of dedicated servers for central platform level operations, together with a set of donated nodes both providing computation or data storage resources.

Data Nodes donate storage space in order to store application files. They provide operations to get and store data. Compute nodes, offer computation resources to the environment. With a view to maintaining isolation among the donated resources and applications executed by means of Nebulas, it employs NaCl sandbox provided by Google Chrome browser. By means of this sandbox, Nebulas orchestrates the execution of NaCl executables into the contributed resources. Evaluation has been provided for Nebulas MapReduce Scheduler comparing it to current Volunteer computing models BOINC and MapReduce-tuned BOINC. This evaluation has employed an experimental set-up using 52 Nodes in PlanetLab using a Word Count MapReduce Like application. Similarly to ANGELS (Mukherjee et al., 2014), NEBULAS develops the idea of volunteer contribution of Edge resources, however, elaborating by-design management of fault-tolerance to edge devices churn and volatility.

2.5.3.1.5 Resource Provisioning for IoT Services in the Fog (Skarlat et al., 2016) main objective is to provide both theoretical and practical foundations for resource provisioning in Fog environments. It provides a systematic classification of Edge resources. This classification comprises the following classes for resources: fog cells, single IoT devices that control a series of other IoT resources while providing virtualised resources; and fog colonies, described as micro-data centres built-up from a series of fog cells. In the proposed architecture: The Cloud-Fog control middleware is the central unit that supports the management of underlying Fog colonies. The management of fog colonies incorporates execution of fault tolerance processes over fog cells as well as novel device discovery, and re-organisation of colonies if needed; Fog Orchestration Control Node supports a fog Colony constituted by diverse Fog Cells; and Fog Cells are software components running on Fog devices. Both the Fog orchestration control node and Cloud-Fog control middleware need to implement placement optimisation for tasks execution. The selected optimisation criterion in this work is twofold, first to optimise resource utilisation at fog cells and secondly to minimise delays in propagating data to cloud. This hierarchical architecture is more complex than MobileFog's one, developing various Fog levels. Evaluation of the proposed model has been performed using an extension of CloudSim simulation framework for Fog Computing, resulting in 39% delays reduction.

2.5.3.2 Existing products in the market

2.5.3.2.1 Azure IoT Edge Azure IoT Suite Reference architecture (*Azure IoT Reference Architecture*, 2016) considers three central aspects for a typical IoT solution: device connectivity, data processing, analytics and management; and presentation and business connectivity. Recently Azure has announced the availability of Azure IoT Edge ("Azure IoT Edge", 2017) as Open Source ("Azure IoT GitHub", 2017). The provided open source software can run on Windows and Linux/Mac powered devices. IoT Edge modules are executed as Docker compatible containers. The IoT Edge Runtime provides monitoring and workload execution functionalities at the Edge.

It allows data pre-processing on-premises before sending it to Azure Cloud environments. The Microsoft services which can run on these devices include Azure Machine Learning, Stream Analytics Azure Functions, Microsoft's AI services and the Azure IoT Hub. Azure IoT Hub component contains device registry and identity store, as well as, device-to-edge and edge-to-device messaging features, acting as the entry point to access the rest of IoT suite services at Edge side. Azure IoT Hub presents an SDK that allows interoperability with custom gateways and simplified programming, Stream Analytics component offers real-time event processing so to support stream data analysis by processing telemetry, data aggregation, and event detection. On the Cloud side, Azure Storage offers long term data and object storage. This can be used in conjunction with Azure Web Apps and Microsoft Power BI, so as to have data visualisation means. At time of writing Azure IoT Edge can be used free of charge while associated use of Cloud services is billed based on usage.

2.5.3.2.2 AWS Greengrass AWS Greengrass ("Amazon Web Services Greengrass", 2017; "Amazon Web Services Greengrass FAQs", 2017) offers an Edge computing platform which propounds local computing using AWS Serverless technology (AWS Lambda), messaging, data catching sync and ML inference while providing interoperability with AWS IoT Cloud services. It is a software stack available for any ARM and x86 device with minimum required capacity (1GHz of compute, 128MB of RAM plus additional resources for workload and message throughput). At time of writing, AWS Greengrass documentation details that compatibility tests have been validated with more than 40 devices. In addition it offers direct communication and operation with Amazon FreeRTOS micro-controllers. The software stack is divided into three main pieces: AWS Greengrass Core, AWS GreenGrass SDK and AWS IoT Device SDK. The Greengrass core allows for: local deployment of applications using lambda functions developed in Python 2.7, Node.JS 6.10 and Java 8; enables secured local messaging based on OPC-UA protocol; provides device management and device clones; and authentication and authorisation in device to cloud communication. AWS Greengrass SDK permits Lambda functions to interact with Core services. The extended IoT Device SDK endowed with Greengrass offers an extension to existing AWS IoT Device SDK so as to support constrained devices (supporting TLS) to communicate with Greengrass core. In addition, devices can use Greengrass discovery API to locate and manage secure communication to Greengrass core. A very interesting feature added recently is Greengrass ML. This feature allows ML models that have been developed and trained in the cloud, to be deployed and executed locally in the Greengrass core equipped device. This is reported to support GPU utilisation for devices which have it present.

It is important to remark that AWS Greengrass supports the possibility to work offline (without Internet connection to the Cloud) performing synchronisation process when connectivity is ensured to the device. Logically, this has to be limited to the resources available on the device powering the AWS Greengrass Core, albeit no concrete information is presently found in the product information. Pricing for AWS Greengrass considers a combination of devices installed plus the usage of Cloud services these make. The price for devices can be charged monthly or with a fixed yearly amount.

2.5.4 Features Comparison

In Table 2.4 we present a comparison of features among all analysed architectures. The analysis compares features considered by research works and commercial offerings. In this analysis, the observed maturity of market developments possesses a remarkable nature. These today are considering advanced capabilities with regard to Data management, Edge workload execution models adapted to the last trends on the market and even consideration of machine learning frameworks. At the same time analysed works in research elaborate on conceptual approaches and future requirements while existing implemented architectures are yet scarce. It is interesting to note that as previously introduced OpenFog architecture limits Edge computing to intermediary nodes among IoT devices and Cloud, while considering Fog, as the computing continuum that embraces end to end management from IoT devices to Cloud. However, from the provided descriptions, it is clear that commercial Edge computing offerings, and specifically Amazon Greengrass, go far beyond providing an intermediate computing layer. Instead, these develop end to end solution for both IoT devices, computing at the edge and rich cloud services, commercial products make reality the computing continuum concept nevertheless exposing its adopters to strong vendor lock-in.

According to current developments it can be the case that instead of research works feeding industry products with advanced features and ideas, it is research lagging behind industrial developments. This is to some extend corroborated by initial experimentation done in commercial offerings with rich IoT devices in connected vehicles (Barr, 2018a; Rec, 2018) which represents a clear initial step towards the realisation of Decentralised Cloud concept defined by this work. This experimentation while exploiting the inference of ML at the edge recognises the need of edge groups of devices and its communication. As it happened in the area of Grid, and the successful application of Cloud utility models in the market almost a decade ago. Nowadays opportunities in research are apparently in scheduling, orchestration and optimisation problems instead of basic capabilities already being tackled in interesting approaches by commercial developments of major Cloud providers. These commercial offerings are advancing at impressive rapid pace and getting quickly into quite mature stages, research and standardisation works have yet to achieve.

Works	Fog computing, a platform for internet of things and analytics	ANGELS	MobileFog	Nebula	Resource Provisioning for IoT Services in the Fog	AWS Greengrass	Azure IoT Edge
Nature	Research	Research	Research	Research	Research	Comm.	Comm.
Edge Management: Resource management	X	X	X	X	X	X	X
Edge Management: Fault Tolerance				X	X		
Edge Management: Workload Mng.	X	X				X	X
Edge Management: Workload Scheduling					X		
Edge Management: Data Mng.		X		X		X	X
Edge Interoperability: Orchestration across Edge							
Edge Interoperability: Interoperability with Edge and Cloud						X	X
Economy						X	
Eco-efficiency							
Sec. and Priv.						X	X
Connectivity and Resilience						X	X
Edge Computing Model	Cluster	Server	Hierarch.	Cluster	Hierarch.	Server	Server
Edge Offering	IaaS	PaaS	PaaS	PaaS	IaaS	PaaS	PaaS
Prog. Model / Language				JavaScript		Lambda, containers	containers
Maturity	Arch.	Arch.	Arch.	Simulation	Simulation	Product	Product

Table 2.4: Edge Frameworks Feature Comparison

2.6 Conclusions

In this chapter we have presented the current state of the art and research challenges for future Decentralised Cloud models. In these we observe that Mobile Cloud Computing has already developed a number of valuable tools and techniques that can significantly influence the future evolution of Cloud models. Specifically, in relation to Cloudlet and Edge Computing and Mobile Ad-hoc Cloud.

Building its routes in Cloudlet concepts we observe that still Edge Computing research is very much in a conceptual state. At the current state of development, multiple works have elaborated on diverse conceptual approaches for it, however, very few architectures do elaborate on management of specific aspects and still research gaps are appreciated in research challenges such as: across Edge execution models, Economy, Connectivity and Resilience.

Interestingly, while the research community is still debating the most appropriate term to use (Edge/Fog), major cloud providers are already launching significantly mature products to the market even exploiting aspects such as ML inference at the Edge. This gives a clear indication on how promising Edge Computing developments are and the need for future research works to take into consideration commercially developed products in order not to re-invent the wheel.

At the same time, expected gains in complexity of the connected complex IoT devices will designate specific requirements to Decentralised Cloud Computing evolution (Juan Ferrer et al., 2017).

These environments are initially taking form in the evolution of Ad-hoc Clouds enabling smart collaboration among mobile devices. These build their routes in ad-hoc networks and opportunistic computing. Further evolution of this concept is expected to enable the creation of dynamic ecosystems, meshes or swarms of complex IoT Edge devices in fully distributed and decentralised manner in the so-called Decentralised Cloud.

At the same time we are witnessing to very significant advances in AI and deep learning technologies which fuelled by the unstoppable data availability collected from complex IoT Edge devices will soon increase computing demand by several orders of magnitude.

While the relation among these technologies is starting to be tackled by both research and commercial efforts ("Amazon Web Services Greengrass", 2017; Morshed et al., 2018; Satyanarayanan, 2017), it further calls for development of Decentralised Cloud environments as ecosystems of complex IoT devices in which resources capacities are complimented by connection to other objects in the community. These have to be designed to allow the dynamic creation of dynamic devices eco-systems encompassing IoT devices, cyber-physical devices, edge and clouds, each of these adding to the collective capability and insight, in a future computing continuum which will act as the backbone in which to build collective intelligence.

CHAPTER 3

Concept and Architecture

3.1 Overview

As it derivates from the state of the art analysis presented in previous section Chapter 2, Edge Computing currently reflects one of the major IT trends, originated at the intersection among four main IT developments: Internet of Things (IoT), Cloud Computing, Networks and more recently, Artificial Intelligence (AI).

Edge computing has emerged with the objective to bring Cloud computing capacities at the "Edge" of the network to address latency issues present in IoT scenarios.

Edge computing serves the purpose of providing a compute environment located in the vicinity of data generation sources able to prevent latency issues detected in accessing Cloud services. Edge Computing brings together networking with distinctive cloud principles to define distributed computing platforms in charge of meeting the specific needs of IoT(Bonomi et al., 2012).

This has indubitably established an initial step towards the decentralisation of Cloud computing by initiating its transformation from the provision of services in dedicated datacentres for which resources were perceived as unlimited to a more decentralised approach in which these cloud services are presented in combination with stationary Edge devices (Cisco Systems, 2016).

This approach is today materialised in existing market offerings which provide initial IoT data filtering and pre-processing with integrated synchronization with cloud services by major providers such as AWS Greengrass("Amazon Web Services Greengrass", 2017) and Azure IoT Edge("Azure IoT Edge", 2017).

Today's Edge computing environments are principally considered stationary dedicated Edge computing devices. However, Connected IoT devices are widely available at the same time as they have incorporated noteworthy compute resources. For some time now, IoT environments do not merely include simple sensors with 8-bit microprocessors(D. Chen et al., 2016), but they are increasingly composed by complex devices which are assemblies of non-negligible computing and storage resources aggregated together with diverse sensors and actuators. Examples of such devices are robots, drones and autonomous vehicles. Moreover, innovative compute devices are being released on the market including application specific processors for AI processing to facilitate embedding compute intelligence into all kinds of IoT devices.

The growth in the complexity of IoT devices is calling for Edge computing to take advantage of all compute and storage capacity available in a specific location in all kinds of stationary and IoT edge devices.

The particular characteristics of the IoT devices which partake in this infrastructure pose special challenges to be addressed in the Ad-hoc Edge Cloud architecture in relation to its overall resource and service management practices. The aim of this section is to analyse the unique challenges that the use of IoT devices prompt and present the Ad-hoc Edge Cloud architecture proposed in this Thesis to address them (Section 3.5). In addition, we elaborate on specific use cases that could benefit from the Ad-hoc Edge Cloud concept.

3.2 Definition of Ad-hoc Edge Cloud Concept

The unprecedented growth we are witnessing nowadays in the number connected devices calls for harnessing the idle capacity at the Edge of the network. Connected devices are not only omnipresent, but also significantly gaining complexity (such as robots and autonomous vehicles).

The overall idea of Ad-hoc Edge Cloud is to exploit available capacity at the Edge in order to create on-the-fly and in an opportunistic manner distributed compute infrastructures, Ad-hoc Edge Clouds. These Ad-hoc Edge Clouds meet the demand for extracting value out of the overwhelming data availability collected from IoT Edge devices and respond to the growing processing demand at the Edge based on the technological advances of AI and Deep Learning.

To do so, Ad-hoc Edge Cloud takes into consideration the specificities of Edge devices, their characteristics and the non-dedicated inherent quality of the infrastructure by developing mechanisms to handle their expected massive number, the availability of dynamic resources (due to mobility and resources constraints) and the heterogeneous nature of IoT Edge resources.

Ad-hoc Edge Cloud develops the idea of Decentralised Cloud (see Section 2.6), by creating dynamic ecosystems of IoT Edge Devices in a completed distributed and decentralised manner. Hence Ad-hoc Edge Cloud relies on decentralized management distributed among all participant resources, which avoids a single point of failure for the infrastructure and offers inherent mechanisms to scale. Likewise, this fact also ensures the autonomy of the Ad-hoc Edge infrastructure, by eliminating the reliance on external management layers, for instance, located at the cloud which can hinder its operation in case of unreliable connectivity.

The target edge devices for this Ad-hoc Edge Cloud constitute limited resources in terms of computational power and storage capacity. These edge devices can be increasingly regarded as non-trivial assembles of various assorted sensors with a set of storage and compute resources. The manufacturers of these kind of devices, such as drones, robots or automobiles, are submitted to pressure to provide more AI-enabled intelligent capabilities at commercially viable costs. The mobile nature of these devices makes them subject to restricted capacity batteries for energy supply.

Enhancing edge devices' on-board computational and storage resources, increases their cost and energy demand, therefore resulting in reduced device autonomy. Hence, any framework aiming to operate in such environment needs to consider these limitations. Thereby, Ad-hoc Edge infrastructure management components will require very lightweight implementations which do not hinder the normal functioning of the target edge devices.

Distributed and decentralised management is also fundamental in order to handle the uncertainty on resource availability in the Ad-hoc Edge Cloud. As previously introduced, the dynamicity present in resource availability is commonly referred to as node churn. It refers to the volatile behaviour of resources concerning their accessibility to be part of the system. Node churn describes the dynamic behaviour of resources appearing and disappearing from the system. Node churn contemplates the "change in the set of participating nodes due to joins, graceful leaves, and failures" (Godfrey et al., 2006).

Node churn is motivated by many different factors, such as a change of edge device locations, they can deplete the battery or trigger an occasional loss of connectivity. Node churn stresses the usefulness for the Ad-hoc Edge Computing infrastructure to manage uncertainty on resource availability. Edge resources can suddenly appear or disappear from the environment triggered by a change of edge device locations, they can deplete the battery or be affected by many factors which influence the availability as part of the infrastructure.

Edge resource node churn and resource volatility also heighten the importance to provide a decentralized management system to the Ad-hoc Edge Cloud, in order to avoid a single point of failure. It equally aims at ensuring its autonomy by not relying on external management layers located i.e. at the Cloud level, potentially hampering the infrastructure operation in case of losing external connectivity to the cloud. Thereby, the Ad-hoc Edge Computing infrastructure management processes need to be managed in a distributed manner among all available Edge resources.

3.3 Ad-hoc Edge Cloud Resources Characteristics

A significant factor which differentiates resource management in Ad-hoc Edge Computing is the high degree of heterogeneity of the edge devices likely to be involved in the infrastructure. Targeted Edge devices range from dedicated stationary Edge devices to rich IoT Edge devices which adjust to certain characteristics (see additional details below in this section).

In this context, diversity undoubtedly stems from the variety of capacities of the edge resources intended to be used, as well as from aspects such as: supported operating systems and processing architectures (i.e. CPU and GPU). It is imperative for our thesis to handle the diversity of edge devices engaged in the infrastructure.

In addition, the mobility of the edge devices which form this computing infrastructure leads to a significant breakdown of existing resource management practices in Cloud and data centre, as far as connectivity instabilities are concerned and their expected massive number. Furthermore, they are affected by specific factors, namely the prerequisite of energy optimization and battery lifetime, which can exert negative influence on the availability of these resources.

In view of the above, edge resource management in Ad-hoc Edge Computing infrastructure reflects the requirement to support the operation in a massive number of heterogeneous constrained devices and presents the requirement of being able to manage dynamic behaviour in relation to resource availability.

Therefore, the four main distinctive aspects which describe the kind of infrastructure resources Ad-hoc Edge Cloud rely on are: its scale in terms of constituent devices, their heterogeneity, their potential mobility and their restrictions in terms of the battery and overall capacity. These four characteristics are crucial to understand the dynamic availability of resources considered by this Ad-hoc Edge

Cloud and which significantly differs from typical resource management practices in Cloud computing.

Massive scale There is a plethora of disquisitions regarding the number of expected devices worldwide (Nordrum, 2016). While it seems very hard to offer a precise estimation of the number of connected devices in the coming years, the reality is that all trends reflect a massive growth in the number of connected devices in a wide diversity of scenarios (Petrov, 2019). This has constituted one of the main drivers to the emergence of Edge computing and is expected to continue. While Ad-hoc Edge Cloud is obviously not addressing all kinds of connected devices, it is an important factor for its design and development the consideration of scale in the number of devices able to participate in the Ad-hoc Edge infrastructure. Specifically, the expected scale in the number of devices which punctually can be part of the generated compute infrastructure raises the need to employ new management styles able to cope seamlessly with situations with variable and massive number of devices.

Heterogeneity The cloud computing model is sustained on economies of scale. These are enabled in large public cloud providers by the capacity of automation over standardised and homogeneous huge farms of servers which provide compute, storage and networking resources to the specific cloud services

Homogeneity of resources which provide a specific service in these large data centre set-ups leads to reduced operating costs with the help of standardised management practices and automation.

Edge computing environments, on the contrary, are characterised by their heterogeneity. The expected massive growth in connected IoT devices together with the wide variety of use cases in which these can be employed, brings diversity at several levels. Devices at the Edge can range from simple sensors able to capture data (i.e. a temperature sensor) to complex aggregations of sensors and actuators embedded together with high performant compute and storage resources, such as an autonomous car(Taivalsaari & Mikkonen, 2018). Edge devices can be stationary wired powered devices for which size is not a critical design issue or constrained battery powered mobile devices with strict requirements for optimisation of devices autonomy.

Additionally, the increasing computing demands for these devices to provide more intelligent features is prompting the emergence of innovative sets of compute devices which can be embedded and employed into IoT environments and devices. In recent years the use of Raspberry Pi (“Raspberry Pi”, 2019) for this purpose has grown in enormous popularity (Johnston & Cox, 2017). However, nowadays the rise of AI and its demanding compute requirements, is generating the appearance of embeddable devices, AI accelerators, designed specifically for the execution of AI at the edge (Tang, 2019) by means of providing specific purpose hardware micro-processors and computer systems such as Intel movidius (“Intel Movidious”, 2019), NVIDIA’s Jetson systems (“NVIDIA Jetson”, 2019) and not long ago, Google Coral (“Google Coral (beta)”, 2019) to cite some noteworthy examples.

In this context, it is crucial for our analysis to be able to cope with all diversity arising from the medley of capacities of the edge resources which can participate in an Ad-hoc Edge Cloud including different processor architectures (CPU, GPU, TPU, FPGA) in addition to other considerations such as a variety of operating systems in target edge devices can operate (i.e. Linux, Raspbian, Robot Operating System) as well as connectivity protocols and technologies which need to be supported.

We aim to adopt Edge devices categorization provided by HEADS (HEADS Project, 2016) project which classifies Edge devices as: Tiny (8 and 16- bit microcontrollers), Small (between 64-128 Kb memory) and Large ("devices running a general operating system like Linux or similar" (HEADS Project, 2016)). Large classification encompass devices such as "Arduino Yun, Raspberry Pi, Android, and iOS" (HEADS Project, 2016). Devices executing general purpose operating systems (classified as Large in this categorisation) represent the specific target of Ad-hoc Edge Computing.

Long term feasibility of this approach is evidenced by increasing support for operating-system-level virtualisation, containerisation in typically- considered largely constrained environments. This is demonstrated through the increased availability of containerisation technologies more lightweight than Docker ("Docker: Enterprise Container Platform for High-Velocity Innovation", 2019) or LXC ("Linux Containers", 2020), such as Unikernels (Madhavapeddy et al., 2013; "Unikernels, Rethinking Cloud Infrastructure", 2019), Kata Container ("Kata Containers", 2019) and gVisor ("gVisor, Container Runtime Sandbox", 2019).

Resource Limitations As previously presented, heterogeneity serves as a salient characteristic of this environment. However, generally speaking, target IoT Edge devices for this analysis are restricted in terms of computational and storage capacity. The motivation for this, is the condition for IoT Edge devices producers to achieve the appropriate trade-off among cost, energy consumption and performance in the devices being launched to the market (D. Chen et al., 2016). Providers of intelligent IoT Edge devices such as smartphones, automobiles, robots or drones require the right balance among rich functionalities, appropriate energy consumption to optimise device's autonomy and overall profitable solutions' costs. In this complex environment, IoT Edge resource providers tend to choose optimal cost solutions with lower performance, reserving the option of higher performant processor architectures, as those described in the previous section, for cases from which they derive a significant competitive advantage for their devices. In this manner, a solution reliant on this kind of devices must anticipate limitations in terms of available compute and storage resources together with constraints in batteries as energy supply.

Mobility Another source of differentiation between Resource management in the context of Ad-hoc Edge Cloud and the traditional data centre resource management in Clouds is the potential mobility of the devices participating in the infrastructure.

Mobility of resources involves using unreliable network links for the Edge device connectivity and its consequent, resource volatility and lack of reliability. These issues are key aspects of this study and have remained unanalysed in the present Edge and Cloud computing stationary resources environments. Node churn, the term commonly employed to describe the dynamicity in resources appearing and disappearing of the system, is instead an area widely studied in P2P (Stutzbach & Rejaie, 2006) and to a lesser extent in Mobile Cloud Computing (Juan Ferrer, Marquès, et al., 2019) areas.

Furthermore, mobility of devices entails battery-powered environments and must optimise battery life in order not to compromise the autonomy of devices and their availability to the Ad-hoc Edge computing infrastructure. Hence, mobility of devices prompts two main matters to be considered as part of this study volatility in the node availability in the infrastructure, node churn, and resource limitations.

3.4 Motivational Use cases

The following use cases aim to exemplify use cases in which Ad-hoc Edge Cloud infrastructure can demonstrate its benefits and value.

Social computing Jordi has recently installed in his mobile phone the "ShareYourMobile" application. Similarly to Volunteer computing in the past ("SETI@home", 2020), "ShareYourMobile" application allows sharing of mobile devices by donating computing cycles in your mobile to other users in exchange of service credits. This mobile application has been built on top of "Ad-hoc Edge infrastructure" open source framework by exploiting its sharing model enablers which take advantage of compute capacity deployed in the location.

Jordi is today taking a high speed train from Barcelona to Madrid to attend a conference. From the moment in which the train left Barcelona central station, he is playing the "LastUltimateSuperGame" which requires the very last image rendering technology he is streaming directly from a Cloud server, as he does not have a last generation mobile. Once he is halfway he's 4G internet connection starts to slow down. The image quality he is getting is poor now and cannot continue playing. It is a pity, as he was about to get a "SuperChampion badge" that unveils new features in the game. At this point in time Jordi decides to look if available capacity to execute the game in any other user of "ShareYourMobile" is around. There it is, luckily Ana is on the train. Ana is eager to get credits for getting free compute capacity for a project she is carrying out. But today she is watching the movie on the train screen, so not using her iPad at the moment. She has a "veryverylasttechnology" iPad equipped with a 8-core GPU which supports quick image rendering. When receiving a request to use her device capacity from the "ShareYourMobile" application, Ana decides to accept. Configuration of the "LastUltimateSuperGame" changes automatically to use local resources automatically discovered in Ana's mobile instead of cloud services, and continues playing. By the time they approach Madrid, Jordi realises the internet connection is good again, he already got his "SuperChampion badge", and does not want to spend all his credits in "ShareYourMobile" now. So, he indicates the application to stop using shared resources, and the configuration of the game changes to Cloud rendering and sends a thank you message to "ShareYourMobile".

Smart Home Increasingly smart appliances such as virtual assistants and light bulbs are commonly used at home. Edge computing is expected to enable ever more interactive devices. Current market home devices typically respond to voice commands over smart devices, edge-enabled smart devices are expected to respond to conditions in the home through extensive use of sensors and controllers deployed in various areas of the house while keeping this information in close environment so respecting privacy and security.

Progressively at home we account for a series of mobile and fixed devices, including mobile phones, tablets, connected TVs and Laptops. Developments such as the ones presented in Ad-hoc Edge infrastructure context could allow the creation of "Personal" or "Home" Edge infrastructure in which our own devices could rely in order to get additional computing and storage capacities. This scenario will exploit the usage in single tenant personal context, as all available resources will belong to the same user, but will benefit from the dynamicity and churn management. The emergence of these "Personal" or "Home" Ad-hoc Edge infrastructures could even create novel opportunities in the area of Smart Home connected with Assisted Living context, by providing innovative services to patients at home depending on both home and patient health conditions.

Industrial plant Industry 4.0 main goal is to improve efficiency, flexibility and security in Industrial Automation by means of widespread adoption of IT in the Industrial Automation operations. A fundamental component of Industry 4.0 so far has been the application of Cloud technologies to the Industrial domain. Industry 4.0 is expected to require of bringing together industrial robots to expand their capacities with computing power at the Edge and Cloud, enabling devices and production assets to become smarter. Emerging next-generation industrial robotics trends point in the direction of using small, general purpose cheap onboard processors and software defined robots as means of achieving new skills, cognitive capabilities, and greater flexibility. Edge computing contributes to the reduction on volume of data traffic from and to the robot to the resource rich computing environment, by reducing distance of transmitted data, shrinking latency, and overall improving quality of service. In this way function will cease to be solely defined by their mechanical parameters, but also by their software, processing and communication capabilities as they become IoT complex devices. In the long term, this could allow the formation of Ad-hoc Edge infrastructures among all computing elements in the industrial plant including industrial robots, specific equipment such as PLCs and dedicated Edge devices. This has the potential to bring improved flexibility on the software of industrial appliances and to take advantage of available idle capacity in the location. This case considers the single tenant aspect of the envisaged framework and exploits decentralisation approach while improving reliability though avoiding single points of failure in the industrial plant.

UAVs for inspection Unmanned Aerial Vehicles (UAVs) provide a cost-effective solution for infrastructure inspection. Infrastructure inspection activities are relevant in diverse and heterogeneous vertical sectors such utilities, agriculture and logistics. Inspected Infrastructures can include roads, bridges, pipelines, electrical and water grids and other facilities. UAVs generate massive business opportunities enabled by their capacity of capturing valuable data. However, the challenges remain at level analysing gathered data and making UAVs' data, actionable information pieces to be ingested into the rest of IT systems. UAVs' data processing poses big data and IoT challenges by the requirement for analysis of non-standard IoT data including imagery and videos files and streams. This makes necessary specific collaboration mechanisms among UAVs fleets in order to increase flights coverage areas, combined with exploitation of hardware heterogeneity for timely data processing combining compute capacity at UAV, Edge and Cloud.

Connected vehicles Increasingly Connected autonomous and semi-autonomous car is considered the ultimate Edge device for advanced edge computing scenarios. Autonomous cars provide the combination of enormous amounts of sensor data, critical local processing power, and the overriding need to get advanced data analysis tools in richer computing environments. Assisted or autonomous drive requires of a wide range of different computing elements and sensor data to be processed and analyse under ultra-low latency requirements. In addition to this, specific benefits can be easily observable though coordination of vehicles flees and knowledge sharing scenarios, as well as, novel business possibilities services for car OEMs and other one tier suppliers in combination with Smart city services or even involvement of smart road infrastructures.

3.5 Ad-hoc Edge Cloud framework

Figure 3.1 presents the proposed architecture for Ad-hoc Edge Infrastructure. The architecture is structured in two main contexts which are present in all participant devices in the Ad-hoc Edge Infrastructure. Contexts represent separations of concerns:

Edge Device Context entails tools and mechanisms for the management of a particular node of the infrastructure. It enables IoT Edge resources to execute services or parts of them. Node Manager allows handling a node as part of the infrastructure. It allows for the unified description of the specific resource static characteristics and with support of Node monitor, of its dynamic characteristics. Finally, the component executor can manage the life-cycle of services components to be executed in the Edge node with support of OS workload virtualisation tools such as Docker.

Ad-hoc Edge Cloud Context designates the components devoted to the management of the overall infrastructure cluster distributed among all participant resources. The core component in this layer is the Logical connectivity layer which is based on etcd("Etcd Discovery service protocol", 2019) and handles the distributed indexes to manage nodes in the infrastructure, services and monitoring information as a distributed system with no central management. Etcd distributed storage allows Resource management, Admission control, and Service management to have a unified view of the status of the infrastructure utilising its out-of-the-box replica and data distribution mechanisms.

It is noteworthy to highlight that contexts do not represent layers. This is performed with the objective of evidencing that the architecture is not built in a stack-like manner, but rather that modules and contexts can be executed in any of the Edge Device nodes. Contexts define the separation of concerns with regard to the management of the particular Edge node, Edge Device Context; and Ad-hoc

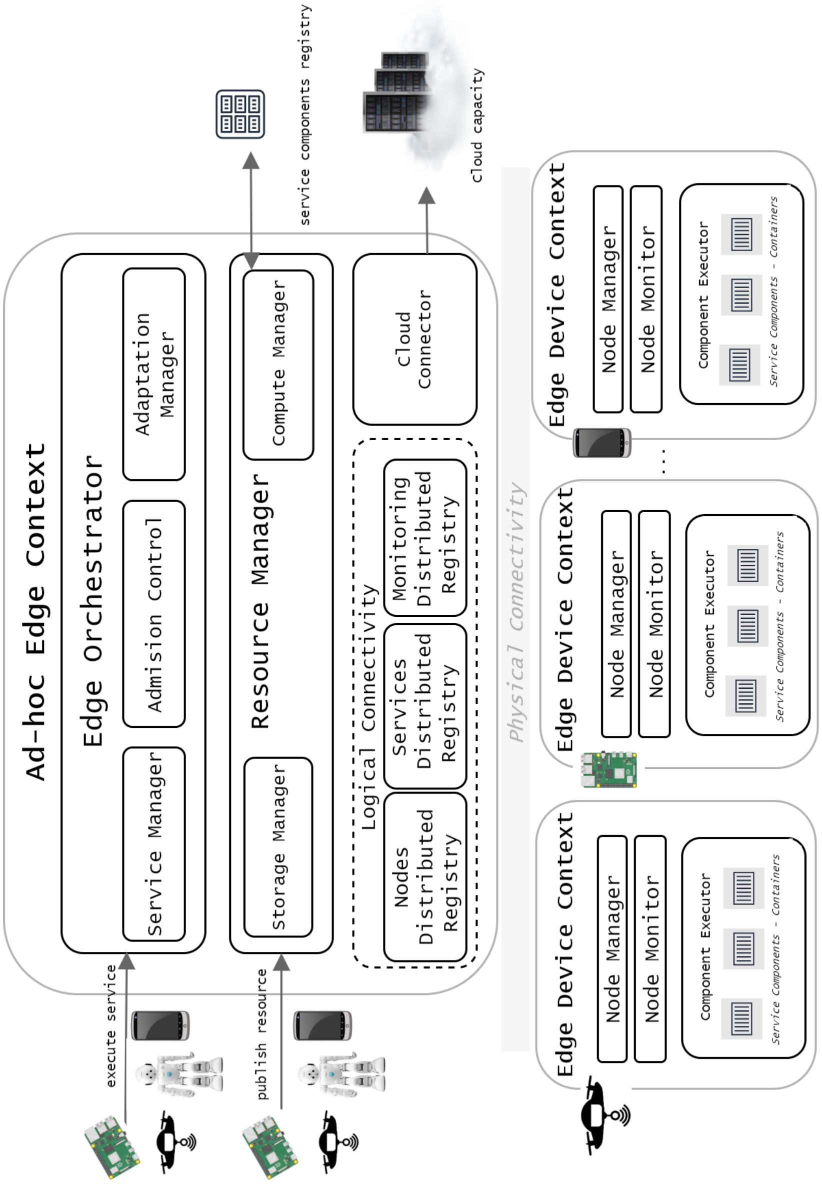


Figure 3.1: Ad-hoc Edge Cloud Architecture

Edge Cloud Context, denoting the overall infrastructure cluster formed out of all participant Edge devices.

3.5.1 Edge Device Context

The Edge Device context modules enable the Edge resources to execute services or parts of them (services' components).

Node Manager The Node manager component oversees the interface in order to manage a node, handling the resources at the Edge device. The main aim of this component is to deliver a unified description for the Edge resource characteristics and capabilities, and their offered capacity to the rest of the infrastructure. The Node manager develops an abstraction layer for all types of resources in the infrastructure which permits heterogeneous resources to be handled uniformly in the Ad-hoc Edge infrastructure.

Component Executor This component provides the means to perform actions related to the life-cycle of application components. The workload virtualisation is based on containers, which facilitate the unified execution in heterogeneous execution environments in a variety of Edge devices. The most popular containerisation system, Docker ("Docker: Enterprise Container Platform for High-Velocity Innovation", 2019) is now present in diverse constrained environments such as Raspbian Raspberry Pi Operating System ("Docker comes to Raspberry Pi", 2016; "Raspberry Pi", 2019), or Robot Operating System ("ROS, Powering the world's Robots", 2019) as well as specific network devices (Vizard, 2017). Docker is increasingly being complemented with even more lightweight implementations of the containerisation technologies among which are included Unikernels (Madhavapeddy et al., 2013; "Unikernels, Rethinking Cloud Infrastructure", 2019) , Kata Container ("Kata Containers", 2019) and gVisor ("gVisor, Container Runtime Sandbox", 2019). This permits us to predict the feasibility of our approach in even more constrained execution environments to those able to support Docker at this point in time.

Node Monitor It collects monitoring information about the status of the Edge node. The compiled parameters include the following aspects: physical infrastructure (memory, CPU usage and available storage) bandwidth (connection type and transmission rate), as well as, battery level status. While bandwidth parameters provide a clear understanding of the quality offered by the node to the rest of the Ad-hoc infrastructure, the battery level is particularly important as a factor indicating probability of churn, therefore with the potential of affecting the availability of the node in the infrastructure.

3.5.2 Ad-hoc Edge Context

Modules in the Ad-hoc Edge Context offer the functionalities which enable the overall infrastructure management. They warrant the handling of all participant IoT Edge resources as a cluster. A crucial module in this context is the Logical Connectivity Layer. It creates and maintains a distributed registry among all participant Edge nodes, allowing for building distributed indexes. The Logical Connectivity layer offers the mechanisms to handle two of the main challenges in the Ad-hoc Edge Infrastructure: distributed management over all available nodes, in order to manage scale; and resource volatility, owing to the probability of node churn. Three distributed indexes are illustrated in this work: Nodes Distributed Registry, which supports the storage of the information about the physical resources of Edge nodes added to the system; Services Distributed Registry, granting access to information of services in the system; and Monitoring and Accounting Distributed Storage, which collects information of node and service execution status and resources consumption.

The enabling mechanism for these modules are distributed key stores. These store the correspondence between a key and a value, similarly to traditional hash tables, running in a distributed system in which look up and storage are scattered among nodes with no central management. Instead, each node maintains a portion of the information along with pointers to the ranges of keys available in other nodes of the distributed storage. Any read, write or look-up operation has to handle, by dint of distributed storage mechanisms, the operation at node level. In Ad-hoc Edge Cloud infrastructure, each Edge node is responsible for a certain part of the overall information system. Each node stores information about resources, services components in execution and monitoring information, therefore without a centralised management as a single point of failure. Distributed key stores offer data distribution and replica mechanisms that supports across node synchronisation and information recovery in the case of a node abandoning the system, and permit the Ad-hoc Edge Infrastructure to manage node churn at information level.

Resource Manager Resource manager component allows resources to be published into the Ad-hoc Edge Infrastructure. It provides the resource specification in terms of device characteristics and capacity. The registration of the node is obtained by means of its incorporation to the Nodes Distributed Registry. The publication of a resource incorporates it to the set of resources to be used in the Ad-hoc Edge Infrastructure. This requires the new node to be bootstrapped into the distributed storage and the generation and management of a cluster among the participant resources, making use of distributed storage capabilities. The Resource Manager also plays an essential role at service deployment time. Resource Manager offers the interface to the Edge Orchestrator to locate nodes selected as part of Admission control process. Within the Resource Manager, the Compute Manager component is responsible for controlling Compute resources in the Edge infrastructure; while Storage Manager handles Block storage resources. The complete Resource management process together with technology choices and evaluation of distributed storages as the enabling mechanisms for Resource Manager and Nodes Distributed Registry are provided in Chapter 4.

Admission Control Admission control supplies the necessary mechanisms which allow the decision making regarding the acceptance or rejection of a service to be executed into the Ad-hoc Edge infrastructure. The challenge of the Admission control component is to select the set of resources that offer sufficient capacity to execute the service, but also to favour those which offer a more stable execution environment in order to handle the environment dynamicity due to node churn. The Admission Control receives the requirements of the service to be executed from the Edge Orchestration in terms of CPU, memory and storage of its components. The Admission Control obtains up- to - date information about available nodes in the Edge infrastructure. Admission Control later performs a filtering and prioritisation process among the available Edge nodes in order to select the candidates able to host a service. Filters represent the set of parameters connected with the capacity of the node. These help to determine whether the Edge device host is able or not to host the workload for a minimum period of time. The filter parameters cover: Capacity, as function of (Memory, CPU and Storage) and available percentage of battery. Once the initial filtering process is performed, the remaining nodes are prioritised according to Ranker parameters. Rankers are viewed as the parameters which measure the quality of the node and its stability. They aim to determine or estimate the QoS provided by the edge device host. Among them, we intend to select Edge devices endowed with longer connection times and better battery levels with the aim of minimising node churn. The result of the admission control process is the assignment of each Service Component to one or a set of Edge Nodes.

Service Manager This module empowers the management of service execution in remote nodes via the Node Manager interfaces interacting with Services Distributed Registry and Monitoring Distributed Storage. Thus, it obtains information regarding the status of nodes and services controlling their availability and performance. The Service Manager is in charge of performing the operational actions in the operational lifecycle for the complete service. The Service Manager locates, via the Resource Manager, the node(s) responsible for the execution of a certain service component. Once located, it interacts with the correspondent Node Manager to implement the required operational action on each node. In case of remote node failure (i.e. due to node churn), it handles, together with the Adaptation Engine, the service re-collocation in another available node. In addition to this, the continuous monitoring process can entail other adaptation actions as a result of the necessity to scale-up or down the number of instances of a component of a service.

Adaptation Engine This module works in close cooperation with the Service Manager which performs the necessary adaptation actions suggested by it. In the event of adding a new Service Component instance to the execution of a service, it relies on the Admission Control which performs the placement decision for this new component.

Edge Orchestrator This component constitutes the entry point to execute a service in the Ad-hoc Edge infrastructure. It receives the Service template which provides a deployment specification of a service to be executed in the Ad-hoc Edge infrastructure. The Edge Orchestrator coordinates via the Service Manager and Node Manager the deployment of the different Service components interacting with Admission Control to obtain placement alternatives. The Service Manager observes service execution interacting with Adaptation engine for ensuring proper service execution.

3.5.3 Ad-hoc Edge Cloud Architecture Flow of Events

A request to execute a service raises the Ad-hoc Edge cluster instantiation. The initiating device utilizes nodes distributed registry reliant on etcd(“Etcd Discovery service protocol", 2019) in order to dynamically discover other available nodes in the specific location. Chapter 4 provides full details of this process.

It is important to note that the Ad-hoc Edge Cloud architecture does not consider at this stage the physical location of the nodes that constitute the infrastructure. Overall, location is regarded in this work as conceptual mechanism that permits that a set of compute resources are gathered together to form a cluster under a pre-defined networking set-up which make them accessible to each other. While the consideration of physical location of the IoT Edge nodes that constitute an Ad-hoc Edge infrastructure can be of upmost interest in certain usage scenarios, it also brings a number of research questions such as the determination of the physical areas in which an Ad-hoc Edge cluster operates, the management of Ad-hoc Edge Cluster overlaps physical locations and physical node discovery mechanisms which mainly operate at the networking level and that are considered beyond the scope of this work.

The Admission Control mechanisms determine the acceptance or rejection of the initiating service considering the placement options it can detect from the characteristics of the service to be executed and the current status of resources currently available to the infrastructure. This process is presented in Chapter 5. In case a feasible placement option has been identified by the Admission Control process, the different service components are instantiated into the correspondent Edge resources using Service management and Edge Device Context Components in each node.

Having reached that point, Service Management processes take responsibility for monitoring the availability and performance of both resources and executing services. In the event of remote node failure for instance, due to node churn, the Service Manager manages the reallocation of the service component to a different available node by executing the Admission control process for the specific part of the service. What is more, it can handle additional deployment adaptations such as carrying out the addition or removal of services' components instances raised by performance observation and elasticity monitoring. This process is illustrated in Figure 3.2.

3.5.4 Conclusions

Over the last decade the growth of connected devices has undergone an extraordinary surge. At present time, connected devices aside from being massively available ubiquitously, have also acquired high levels of sophistication which warrant significant compute and storage resources. Therefore, computing ceases to be available on determined dedicated stationary compute devices, to become widespread and permeating a substantial amount of devices. At the same time, the popularity of AI is rapidly evidencing the necessity of performant data analysis at the Edge. The aforementioned aspects uncover the growing demand for employing all processing capable resources at the Edge of the network.

In this chapter we have defined the concept of Ad-hoc Edge Cloud, as the mechanism that permits us to form dynamically computing infrastructures able to take advantage of increasingly available compute capacity at the Edge of the network. In addition, we have explored which are the specific characteristics of the IoT Edge devices in order to participate in such compute infrastructures and we have identified a set of Motivational use cases which aim to exemplify potential uses of Ad-hoc Edge Clouds.

Finally, we have presented the envisaged Ad-hoc Edge Cloud Architecture developing a framework in which increasing compute capacity at the Edge can be exploited in a distributed manner by enabling ad-hoc formation of Edge infrastructures created out of participant edge devices. This architecture serves as guiding principle for the rest of this thesis in which we elaborate on the Resource management and Admission Control processes and mechanisms.

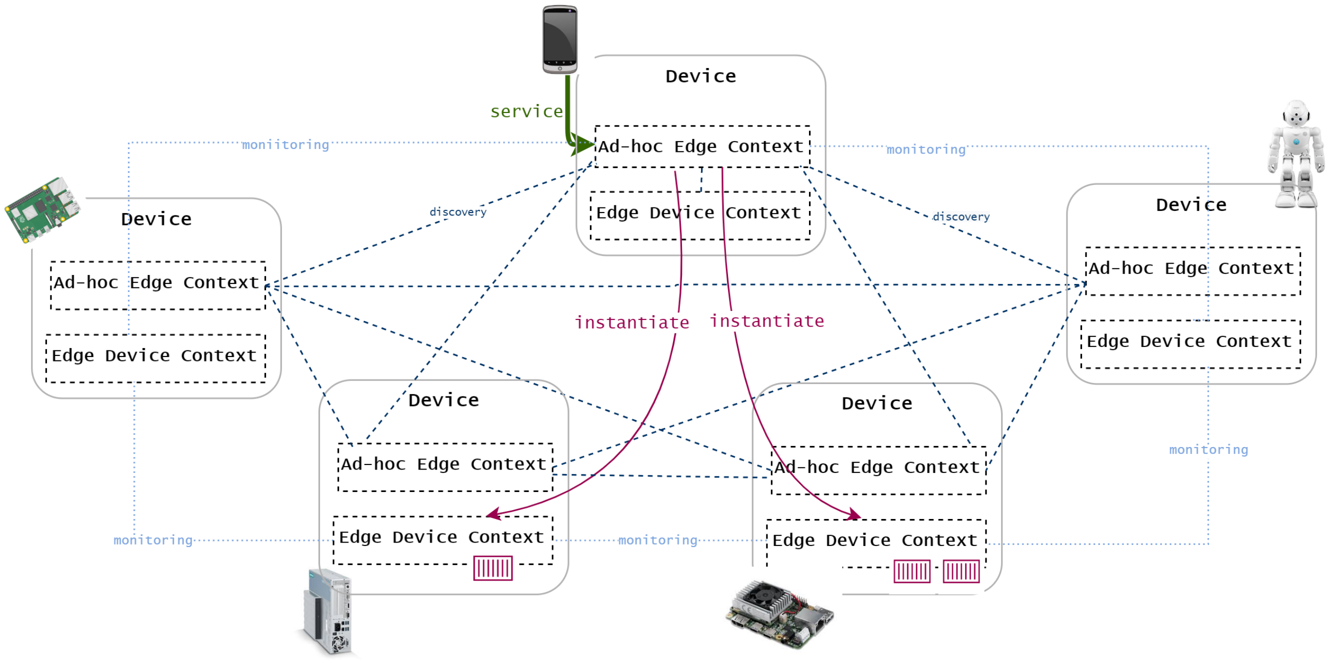


Figure 3.2: Ad-hoc Edge Cloud Architecture Flow of Events

CHAPTER 4

Resource Management

4.1 Overview

The overarching goal of the Ad-hoc Edge Cloud is to dynamically form ephemeral compute infrastructures by harnessing heterogeneous non-dedicated resources accessible at a certain location at a specific point in time.

As previously introduced in Chapter 3, IoT Edge resources form Ad-hoc Edge Clouds. These IoT Edge resources are characterised by four particular factors which distinguish Resource management in this context from existing studies in Edge and Cloud computing: Massive scale, Heterogeneity, Resource Limitations and Mobility (see Section 3.3 for complete details). All these factors lead to a high degree of volatility of resources owing to probability of node churn. Node churn is the term used to describe the volatile and dynamic availability of IoT Edge resources which constitute the Ad-hoc Edge infrastructure. Node churn particularly affects Resource management processes in Ad-hoc Edge Cloud.

The study of the consequences of node churn is crucial to the developments put forward in this chapter. Node churn determines the mechanisms defined for handling the participation of IoT Edge devices in Ad-hoc Edge Cloud infrastructures, but more importantly it calls for using fault tolerant distributed storage systems and distributed consensus algorithms as the key building block for Resource management in our work.

This chapter starts by presenting the defined protocol for which IoT Edge resources will be provided to the Ad-hoc Edge infrastructure, enabling available IoT devices to take part in Ad-hoc Edge Cluster infrastructure and through its entire lifecycle. Afterwards, it details the mechanisms defined in Ad-hoc Edge Cloud for cluster instantiation and management. Finally, it presents the evaluation of two main aspects in terms of handling resources in this context: measurement of the ability to scale in terms of the number of nodes which take part in the cluster, as well as, the reliability to support certain degrees of node churn.

4.2 IoT Device Availability Protocol

The protocol for IoT Device availability as a service host in the Ad-hoc Edge infrastructure entails the complete resource availability life-cycle. It is depicted in Figure 4.1 and described in the subsections below. This protocol describes a particularisation of the general protocol for resource collaboration in P2P described in (Bandara & Jayasumana, 2013).

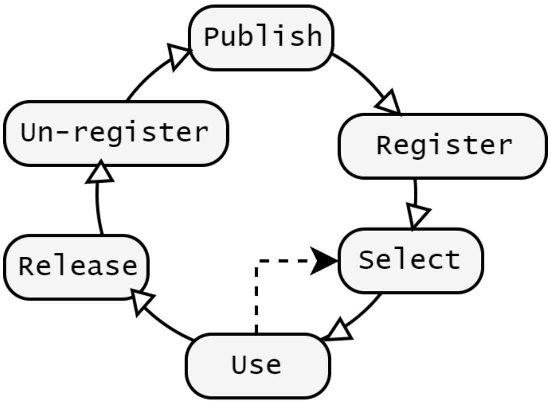


Figure 4.1: Resource Availability Phases in Ad-hoc Edge Cloud infrastructure

4.2.1 Publication

In this phase an IoT Edge Device renders itself available to the Ad-hoc Edge infrastructure. Devices are described by means of resorting to a defined resource specification language able to support heterogeneity on the typologies of resources to engage in the infrastructure. A minimal resource description provides information regarding the characteristics and resources of the host in the following terms:

Static Device characteristics: These are device physical characteristics which remain unchangeable over time. It includes elements such as the description of the architecture of the processors, their overall amount and characteristics, the complete memory capacity of the device, its installed operating System and supported application middleware, such as container execution support.

Dynamic Device characteristics: These are device characteristics related to the device capacity and its load, which will change over the operation time of the resource. They describe the resource in terms of available different processors usage, memory and internal storage, active network protocol, upload, and download network transmission rate, as well as, battery levels at a certain point of time.

In this way in the Ad-hoc Edge Cloud infrastructure an IoT Edge Device host is described as two collections of attributes, static and dynamic attributes, which constitute an available node in the infrastructure. The static characteristics of the devices are determined at publication phase while dynamic device characteristics are fed by means of the collected information on Node Monitor architecture component.

$$h = \{static_{characteristics} = (s_1 = vs_1, s_2 = vs_2, \ldots .s_i = vs_i), dynamic_{characteristics} = (d_1 = vd_1, d_2 = vd_2, \ldots .d_i = vd_i)\}$$

In addition to these, depending on the specific usage scenario other attributes linked with location could be of specific interest such as indoor or GPS coordinates.

The description of a Node permits the Edge Device ContextSection 3.5.1 of the Ad-hoc Edge Cloud framework to uniformly describe all participant IoT Edge resources detailing their offered capacities and features by means of the

Node Manager component. This information is used at operational time by the Admission Control in order to determine the best placement of a service to be executed in Ad-hoc Edge Cloud from all available IoT Edge resources. A diversity of resource description languages has been put forward over time in Grid and Cloud computing environments. Examples of these are the following: Kubernetes Node capacity description (“Nodes, Kubernetes by Example”, 2020), Cloud Infrastructure Management Interface (CIMI) Machine Definition (DMTF, 2016) or GLUE Schema in Grid Computing (Andreozzi, 2009). The suggested format for IoT Edge device description is adapted to the Ad-hoc Edge Cloud needs by offering information on processor heterogeneity can node connections. It is described in Listing 4.1 and represents a minimal description easily adaptable to future needs.

Listing 4.1: Proposed JSON format for IoT Edge device description

```
{"IoTEdgeDevice": {
  "id": "number",
  "IP": "string",
  "discoveryURL": "string",
    "labels": {
      "name": "string"
    }
    "static": {
      "architecture": "string",
      "os": "string",
      "totalMemory": "number",
      "totalDisk": "number",
      "totalCPU": "number",
      "totalGPU": "number",
      "connections": ["string", ]
    }
    "dynamic": {
      "memory": "number",
      "disk": "number",
      "cpu": "number",
      "gpu": "number",
      "uploadSpeed": "number",
      "downloadSpeed": "number",
      "batteryLevel": "number"
    }
}}
```

4.2.2 Registration

This phase requires keeping track of all available resources to a certain Ad-hoc Edge infrastructure. Given the suggested architecture for Ad-hoc Edge Cloud this process involves the registration of the published node into the Logical connectivity layer Nodes’ Distributed Registry. The selected mechanism for this module is a Distributed storage implemented using etcd(“Etcd, A distributed, reliable key-value store for the most critical data of a distributed system”, 2019) which relies in the Raft consensus algorithm (Ongaro, 2014; Ongaro & Ousterhout, 2014). This allows the distributed management and automated replication among all nodes in the infrastructure to turn to a well-known and widely used distributed storage system. The process of Node registration and cluster formation in an Ad-hoc Edge Cluster is detailed in Section 4.3.

At the stage in which a Node is registered, it begins to be monitored, gathering information about its dynamic characteristics and also connection and disconnections to the Ad-hoc Edge Cloud infrastructure in different periods. This information is the keystone on which to develop the resources availability prediction model which will be presented in Chapter 5 Section 5.3 Resource Availability prediction model.

4.2.3 Select

Selection occurs at the time a user makes a service execution request to the Ad-hoc Edge infrastructure. It consists of the Admission control process of identifying from the available host resources those which are in charge of executing the requested service. This phase involves the process of the resource appearing in the list of candidates to execute a service (or a part of it), given provided service requirements. Ad-hoc Edge Cloud provides an innovative mechanism in order to assess the quality of a resource in the infrastructure. This is formulated as a resource availability prediction model. These processes are presented in detail in upcoming Section 5.3.

4.2.4 Use

Once the resource is selected as a potential executor of a service (or a part of it) the corresponding service components have to be instantiated on the target device. Once components are deployed, this phase also considers their operational lifecycle management, including start, stop and resume of corresponding components. Owing to the nature of devices considered, diverse factors, as previously exposed, affect its availability to be part of the Edge infrastructure. Node churn and resource failures must be considered in order to address resource volatility. Therefore, the use phase has to implement continuous monitoring of the operation of the execution and put in place mechanisms for effective workload migration in case of node failure. At the same time, continuous monitoring will also allow the continuous adaptation of the size of the set-up to the defined application needs and user resource contribution constraints by triggering elasticity events. All operational adaptations of the service execution in a given resource will require repetition of phase Selection in favour of replacing or acquiring new resources.

4.2.5 Release

The Release of service resources will be associated with the finalisation of a service a resource executes once this has been terminated. This phase will consider the clean-up of all occupied resources due to the service deployment with a view to ensure its used capacity is released and remains available to other services.

4.2.6 Un-register

Resources registered in the Ad-hoc Edge infrastructure will be continuously monitored. An unreachable resource for a certain period will be considered no longer available to participate in the Ad-hoc Edge infrastructure and therefore un-registered after a certain period.

4.3 Ad-hoc Edge Cluster instantiation and management

Two inherent characteristics of Ad-hoc Edge Cloud are: the requirement for lightweight implementation, considering the nature of IoT Edge devices it aims to operate in; and the requisite to support high degrees of dynamicity, which stems from the high degrees of node churn in this context. While the first characteristic is specifically related to the framework implementation, reliability to node volatility directly influences the design of the Ad-hoc Edge Cloud.

As opposed to traditional centralised cloud resource management systems, Ad-hoc Edge Cloud aims at building a decentralised resource management system able to cope with instabilities in resource availability previously analysed. In order to do so, it observes previous research in distributed and P2P systems and its applicability to the build distributed storages which are vital for supporting decentralisation in Ad-hoc Edge Cloud distributed Resource and Service management.

A widely employed mechanism in distributed systems such Ad-hoc Edge Cloud for handling unreliability in network nodes are distributed storages and associated consensus algorithms. Ad-hoc Edge Cloud does not intend to build these from scratch but instead it targets at developing these features by employing widely spread technologies. Examples of these are Etcd("Etcd, A distributed, reliable key-value store for the most critical data of a distributed system", 2019) and Apache Cassandra ("Apache Cassandra", 2019). In the upcoming subsections the intended mechanisms in Resource management of Ad-hoc Edge Cloud for cluster initialisation and management are presented. These rely on built-in capacities in the selected distributed storage system, etcd("Etcd, A distributed, reliable key-value store for the most critical data of a distributed system", 2019) and its associated Raft consensus algorithm(Ongaro, 2014; Ongaro & Ousterhout, 2014).

In section Evaluation, we will pinpoint the differences among distributed storage systems, etcd("Etcd, A distributed, reliable key-value store for the most critical data of a distributed system", 2019) and Apache Cassandra ("Apache Cassandra", 2019), and provide the evaluation for Resource management and Nodes distributed registry we have performed as part of this Thesis.

4.3.1 Ad-hoc Edge Cluster instantiation

The capability of being able to participate in Ad-hoc Edge Clusters is granted by downloading and installing on the device the Ad-hoc Edge Cloud framework Section 3.5 software which is packaged as a Docker container, and therefore available, at the time of writing, natively in Linux and derivatives in x86-64, ARM and other diverse CPU architectures, as well as on Windows (x86-64). This process is described as Step 0 in Figure 4.2. Source code generated for initial validation of this approach is publicly available in GitHub Ad-hoc Cloud Software repository (Juan Ferrer, 2018). This repository currently contains code from validation of Etcd("Etcd, A distributed, reliable key-value store for the most critical data of a distributed system", 2019) and Apache Cassandra ("Apache Cassandra", 2019) which will be introduced in Section 4.4.

By obtaining the container image from existing repositories and executing the image in the device, the administrator will provide all Ad-hoc Edge Cloud framework

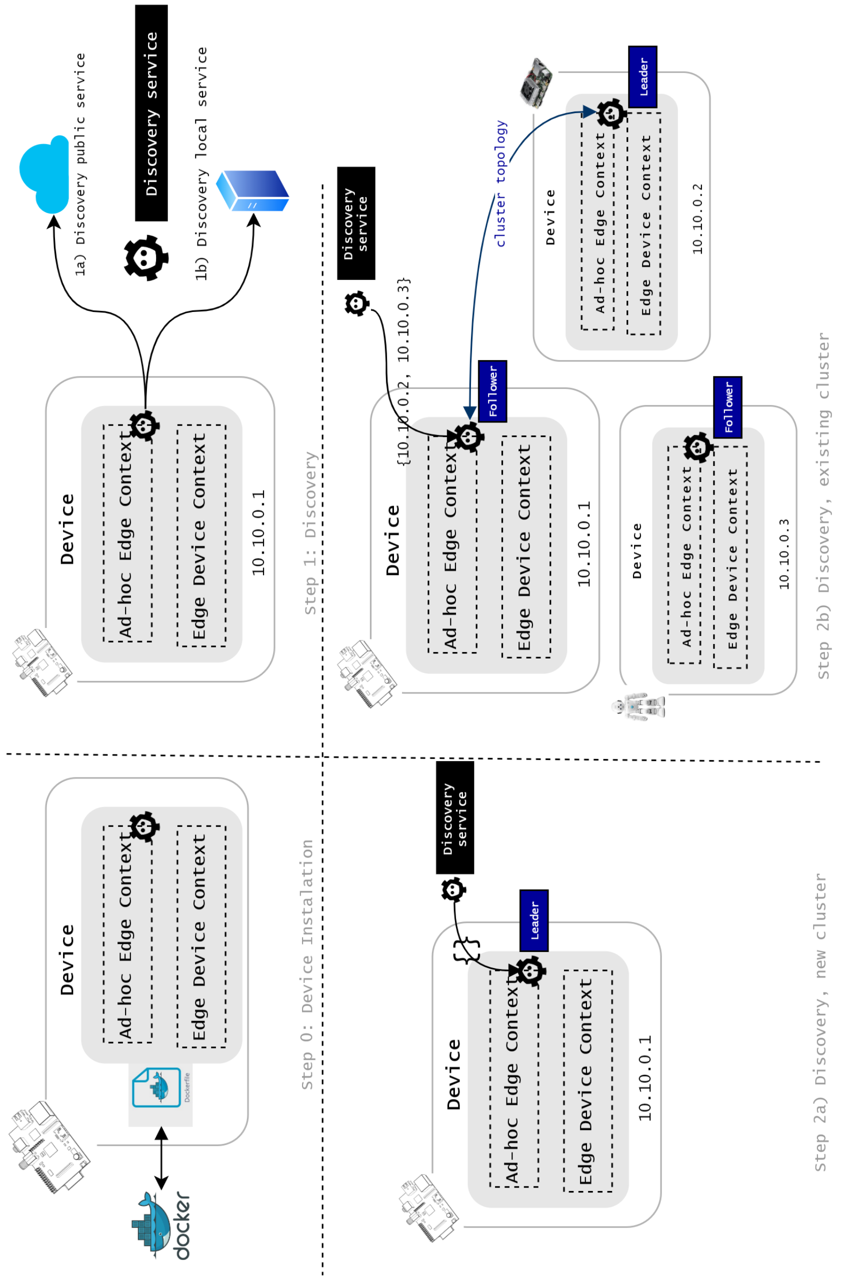


Figure 4.2: Ad-hoc Edge Cluster instantiation steps

software services to the node. In addition, this process will also generate the empty data structures for Nodes and Services Distributed registry and Monitoring distributed storage by instantiating an etcd Node as part of Ad-hoc Edge Cloud node instantiation.

During the execution of this Docker image, the administrator will be able to provide the URL for a discovery service as an environment variable. Depending on the desired configuration, this discovery service will be a public discovery service available in a Cloud or an existing local node external to the Ad-hoc Edge infrastructure. This variable will be fed into the discovery configuration of the etcd node.

To finalise the bootstrapping of the new Ad-hoc Edge Cloud node, the node will make a request to the configured discovery service to register itself into the discovery service supported natively by etcd ("Etcd Discovery service protocol", 2019). The process is profiled in (Figure 4.2 Step 1).

In the case that the discovery service returns an empty set of IP addresses (Figure 4.2 Step 2a), the node will know it is the first node in the Ad-hoc Edge Cloud and associated distributed storage cluster, therefore it will become the Leader of the etcd cluster and register itself into the Nodes distributed registry including its static and dynamic information.

On the contrary, in the case the etcd discovery service returns information about other available nodes in the cluster (Figure 4.2 Step 2b), the etcd node of the device will connect to the fist of returned IP addresses getting all cluster topology and synchronising data on the rest of existing nodes in addition to itself. By definition of the Raft consensus algorithm (Ongaro, 2014) the new added node will only be available to the rest of etcd cluster with full rights, including possibility of becoming cluster leader, once all synchronisation processes are finalised and the node is stabilised and in a consistent state as part of the distributed storage cluster.

4.3.2 Ad-hoc Edge Cluster management

This section provides an analysis of the Ad-hoc Edge Resource Management operation. It is constituted by the analysis of three distinguished cases: the normal cluster operation, the case of addition of a new node and the situation of a node failure. These three situations are detailed in the next sections.

4.3.2.1 Cluster operation

During Ad-hoc Edge Cluster operation, the node which receives a request for a service execution registers it into the Services distributed registry. The way in which the Etcd distributed storage and Raft consensus algorithm handles it is illustrated in Figure 4.3. If the node which has received the request is a cluster leader, it will handle the request directly. In the case it is a follower, it redirects the request to the appointed distributed storage leader. The leader will initiate a request to all followers to add the new registry. Once it gets confirmation from the majority of followers on the write the leader will respond to the request. In the case some of the follower nodes fail to respond, the Leader will continue to try until the write is confirmed in the failing follower or until the follower is removed from the cluster. It is important to note that in this section we are merely approaching the functioning of a Service Request in terms of cluster operation, all the rest of related operations

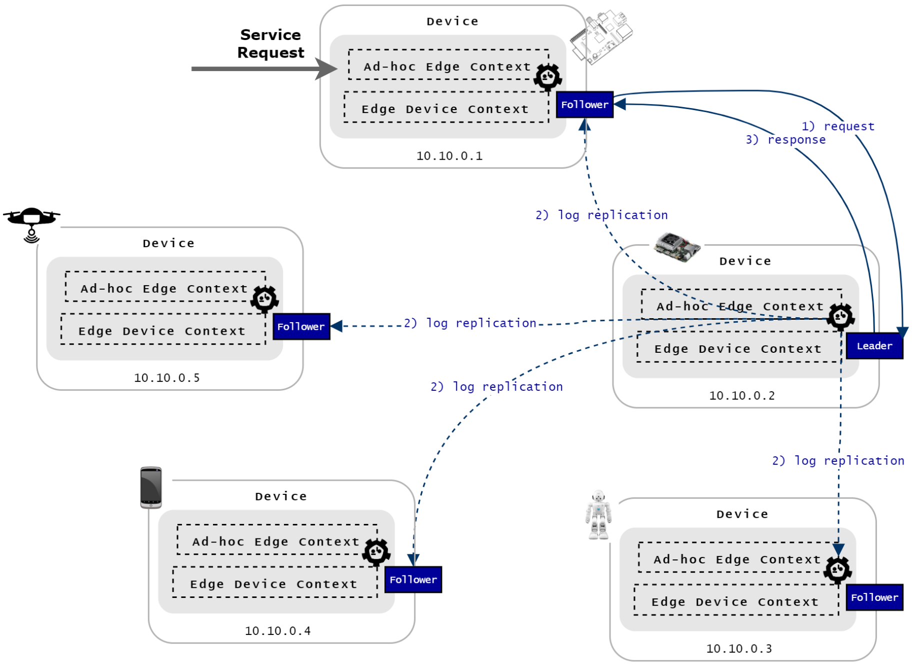


Figure 4.3: Ad-hoc Edge Cluster Operation

in Ad-hoc Edge Cloud have already been presented in Chapter 3 and upcoming Chapter 5.

4.3.2.2 Node Addition

The process of addition of a node has already been presented in Section 4.3.1. It is worth to mention that once a new node has completed the discovery process the Node will add itself to the Nodes Distributed Registry. In this sense the new node will be known to the rest of the cluster at the level of Ad-hoc Edge Cloud Architecture. At the level of the distributed storage layer, the new Node is acknowledged by the existing distributed storage cluster leader by means of the discovery process previously introduced. It is interesting to note that by definition of the Raft consensus algorithm the new Node will never initiate its operation as a leader, if there is an existing cluster, instead all leader election procedures will be activated strictly in the case the existing leader fails.

Typically, a new Node added to the infrastructure will not include pre-existing data. Depending on the amount of data available in distributed storage, data synchronisation process can require some time, and to some extend compromise the performance of the overall distributed storage cluster mechanism. The procedure distributed storage and consensus algorithm (“Etcd Runtime reconfiguration”, 2019) proposes for managing this situation is to add the new member as Learner during the data synchronisation process. Learner nodes synchronise data but they do not participate in several management processes, such as leader election and they do

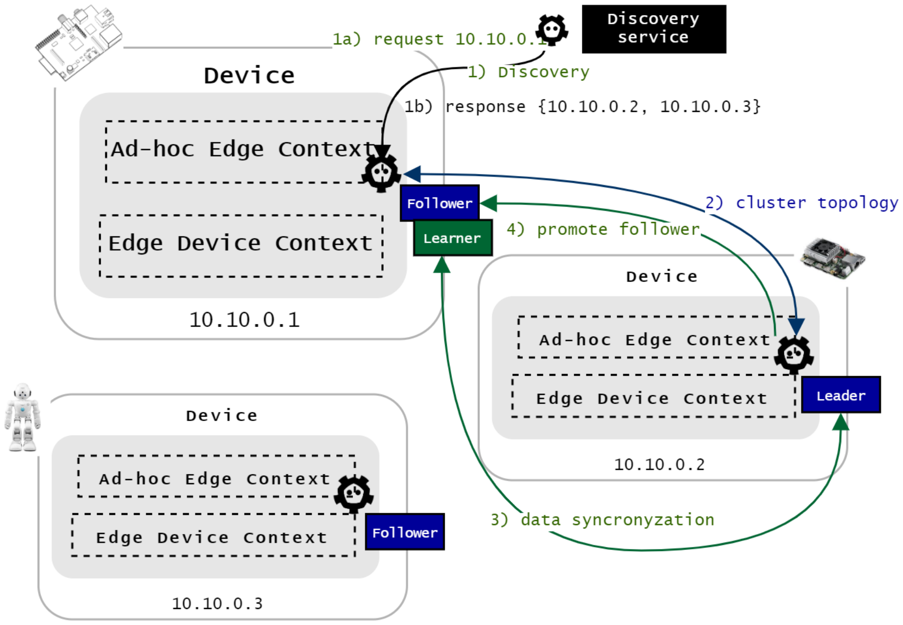


Figure 4.4: Ad-hoc Edge Cluster Node Addition process

not answer to client requests ("Etcd Learner design", 2019). Once the Learner node determines that is in a healthy state, it accepts Leader's request to be promoted as Follower in the cluster. This mechanism is depicted in Figure 4.4.

4.3.2.3 Node Failure

The case of the Node failure is divided into two situations: the situation in which the failing node is a follower at level of the distributed storage or the failure of a node which is acting as a leader.

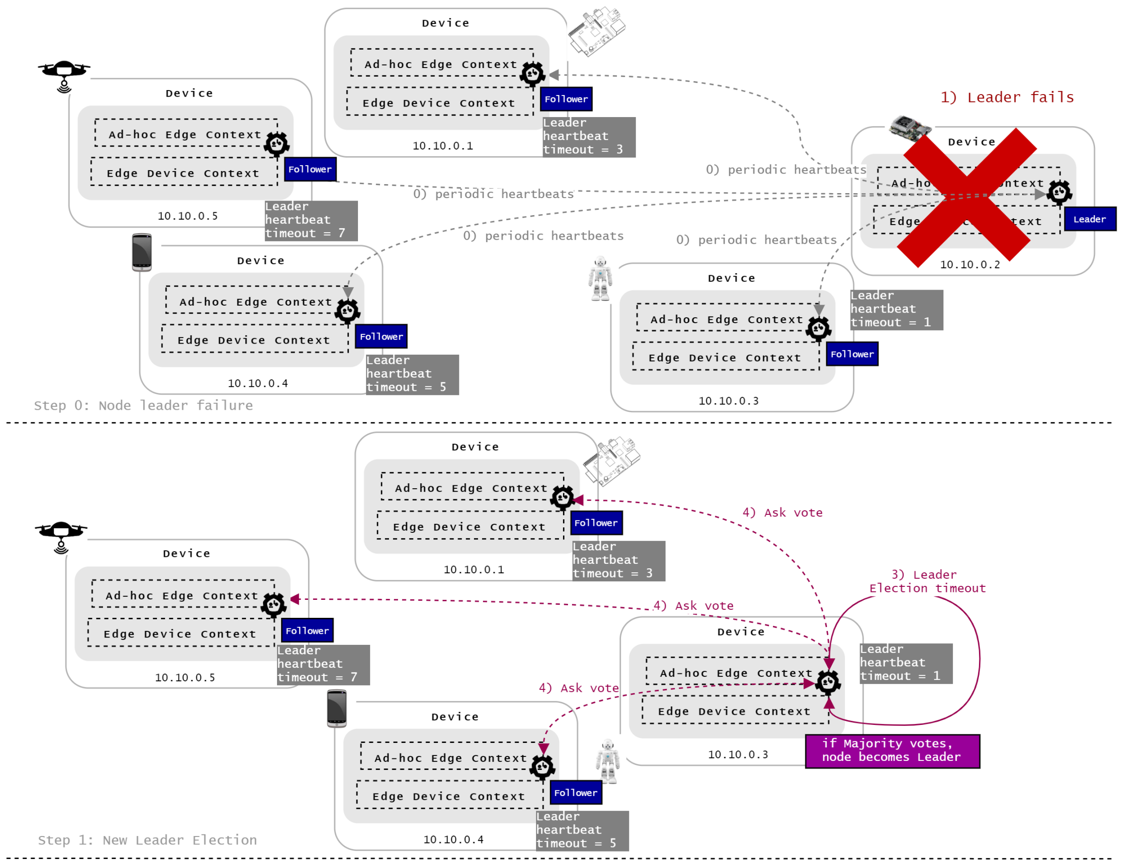


Figure 4.5: Ad-hoc Edge Cluster Node Failure as Leader process

Node Failure as Follower The failure to of a Node as a follower is the simplest case to manage from a cluster management perspective. At distributed storage level, the leader has a continuous heartbeat process to its followers which sustain its authority. Fault tolerance methods in etcd and Raft are able to lend support until the failure of the majority of the cluster. At Ad-hoc Edge Resource management level, a Node not being able to be contacted during a certain period will trigger the removal of the Node in the Nodes distributed storage. In order not to compromise quorum in the distributed storage, it will also trigger a cluster reconfiguration operation to remove the failing node from the list of distributed storage cluster nodes.

Node Failure as Leader The failure of a Node which is a distributed storage leader will follow a similar approach as the previously described from an Ad-hoc Edge Resource management perspective. In other words, it will involve significant changes in terms of the distributed storage and consensus algorithm. In the case of the distributed storage leader failure, followers will cease to receive heartbeats of the leader for a certain period (see Figure 4.5 Step 0). By Raft definition, all nodes as configured in a certain time out to activate a leader election process which is a random value. The first follower node to get the timeout will raise a new leader election process. If it gets the majority of the rest of available nodes votes, it will become the new cluster leader (see Figure 4.5 Step 1). Afterwards, the new leader will begin to send heartbeats to the rest of nodes, preserving the rest of available nodes to initiate the process or terminate it in the case it has been already initiated.

4.4 Evaluation

Validation of the Ad-hoc Edge Computing Infrastructure work has been centred on the analysis of the behaviour of the Resource Management component in situations of massive and dynamic incorporation and removal of Edge nodes. The driving ambition has been to understand how dynamicity on resource availability must be handled in the Ad-hoc Edge Infrastructure. In order to do so, as previously mentioned, we have relied on Etcd(“Etcd, A distributed, reliable key-value store for the most critical data of a distributed system”, 2019) distributed storage. Specifically, we have analysed its behaviour and resource consumption in highly dynamic situations when installed over constrained Edge devices represented by Raspberry Pis. In order to extrapolate our results to a scale not achievable in our existing physical Edge cluster, we have developed a simulation environment in Amazon Web Services (“Amazon Web Services”, 2019) using EC2(“Amazon Web Services EC2”, 2019) and A1(“Amazon Web Services EC2 A1 Instances”, 2019) services.

This has also served us to compare the outcomes with another distributed storage system, Apache Cassandra (“Apache Cassandra”, 2019; Laksham Avinash & Prashant Malik, 2010) in order to assess the differences in performance and feasibility for implementing distributed storage components in Ad-hoc Edge Cloud.

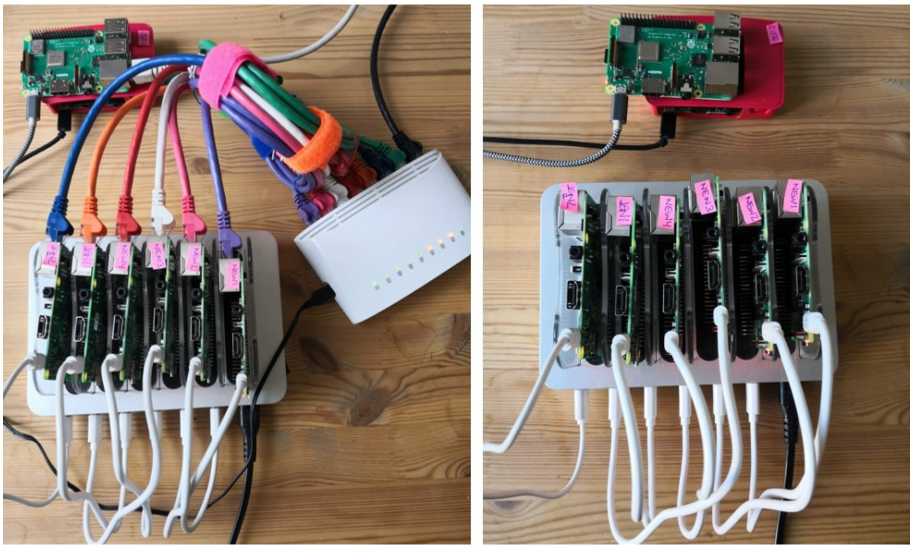


Figure 4.6: Raspberry Pi set-up Ethernet and WLAN connectivity.

4.4.1 Lab Evaluation

Our available physical testbed is depicted in Figure 4.6 and it is composed of hardware of the following characteristics:

- 3 x Raspberry Pi 3 Model B (RaspberryPi.org, n.d.-a) (1.2GHz 64-bit quad-core ARMv8 CPU, 802.11n Wireless LAN, Bluetooth 4.1 and 10/100Mbit/s Ethernet Port) equipped with 8GB 32 GB and 4GB micro-sd memory cards (95 MBs Read and 20 MBs Write speed (Samsung, n.d.-a)).
- 5 x Raspberry Pi 3 Model B + (RaspberryPi.org, n.d.-b) (1.4 GHz Quad Core ARM Cortex-A53, ARMv8-A (64/32-bit), On Board WiFi 802.11ac Dual Band 2.4GHz & 5GHz, 10/100/1000 Mbit/s Ethernet Port) all equipped with 128GB micro-sd memory cards(100MBs Read and 90MBs Write speed (Samsung, n.d.-b)).

The initial step has consisted of flashing all Raspberry Pis micro-sd memory cards with Raspbian Stretch Lite Kernel version 4.1.4 (Raspberrypi.org, 2019). Afterwards, in each Raspberry, Docker was installed. In the interest of respecting the constrained nature of Edge devices when executing Nodes Distributed Registry and Resource Manager functionalities Docker containers in this the Edge environment have been limited to use a maximum of 32MB. This parametrisation is an extremely constrained execution environment for Ad-hoc Edge infrastructure management processes. Our aim when setting up this outstanding resource limited environment has been to validate our approach in an environment which is lightweight in resources occupied by our runtime, leaving capacity for the execution of external workloads. However, we are fully cognisant of the fact that such constrained runtime parametrisation is likely to affect the resultant response times of our results.

The focal points of the evaluation of this environment have been the ability of Nodes Distributed Registry layer to support scale in a timely manner, as well as, analysis of the behaviour and response time in dynamic resource volatility scenarios, considering diverse rates of nodes churn for the Resource Manager. For both of the considered aspects we have drawn a comparison between the results obtained by utilising the Ethernet connectivity versus the usage of a Wireless LAN (WLAN) for the same operations. This has enabled us to establish a baseline in order to assess the impact of using WLAN connectivity. It is important to note, that in both cases, experimentation has relied on best effort from network perspective, as any of the developments of this PhD thesis or the Ad-hoc Edge framework addresses networking aspects associated to this research area.

4.4.1.1 Scalability Experimentation

Experimentation concerning scalability aspects has targeted the evaluation of the behaviour of Nodes Distributed Registry and Resource Manager to support the relevant addition of Edge nodes and the response times obtained. As previously presented in Section 4.3.2, the process of adding a new node consists of instantiating the Docker container in the Raspberry Pi out of the customised Docker image which contains the distributed storage and Node Resource management code. The new node is registering itself both in the Ad-hoc Edge Computing Infrastructure and Nodes Distributed Registry. By doing so, added resources on the infrastructure become an intrinsic part of it, by means of storing part of the information dedicated to its management. As previously mentioned, Source code responsible for implementing this behaviour for Cassandra and Etcd distributed registries is available on GitHub (Juan Ferrer, 2018).

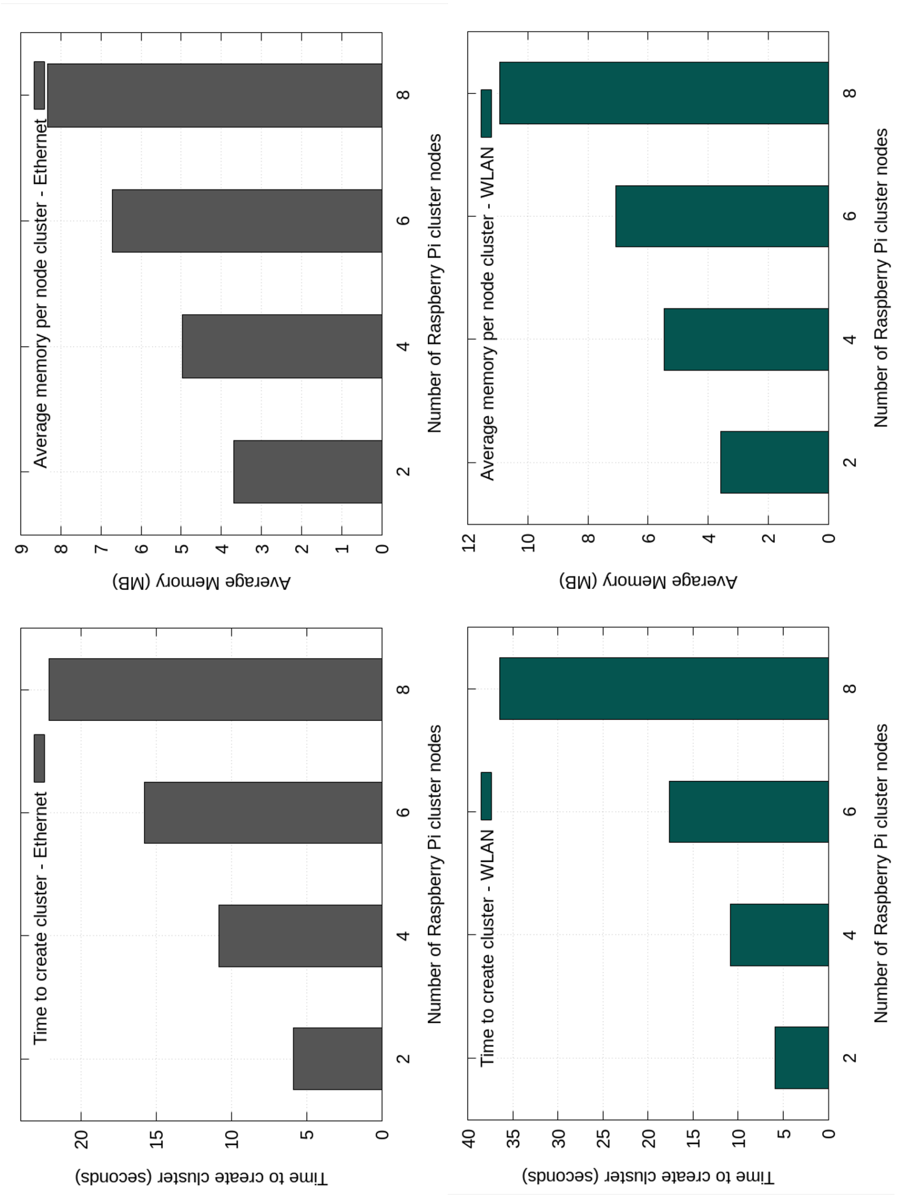


Figure 4.7: Scalability Experimentation over Raspberry Pi testbed.

Scalability experimentation in the physical Raspberry Pi environment has involved all available Raspberry Pis supported by the subsequent creation of clusters of 2, 4, 6 and 8 nodes. In order to do so, the measured aspects have been the necessary time to have an operative Ad-hoc Edge cluster distributed over the selected Raspberry nodes and the resources (memory) these processes consume from the physical nodes. In the execution of these tests we have observed a high degree of variability in response times, due to variable network conditions, therefore this experimentation presents average results obtained by 10 executions of the experiment.

Figure 4.7 exhibits tests performed using both Ethernet and WLAN connectivity. In these tests we can detect that on average the use of WLAN connectivity increases the time to create a cluster. The percentage of increment grows in relation with the size of the cluster. For a two-node cluster the difference of creation time is of 1% reaching 64% for an 8 Nodes cluster. The explanation encountered for this fact is the additional requisite for data synchronisation processes among distributed nodes given that both Ethernet and WLAN experimentation have used the same devices in the same order (with diverse hardware configurations, see Section 4.4.1). This observation imposes the requirement to keep clusters to the minimal possible granularity at level of constituent number of nodes to consider in a single cluster, or at least to take into account the performance overhead produced by larger clusters. It is significant to note that differences detected in memory usage of the nodes in the cluster present on average less than 10% variability.

Figure 4.8 presents another important aspect for cluster creation, the initial installation of the docker images on the target devices. Previous experiments were performed using a pre-installation of the docker image in the Raspberry Pis as a preliminary configuration step. This has proven to be necessary, as resultant cluster creation times considering image download and installation in the Raspberry Pi reach orders of magnitude of minutes per each aggregated node.

4.4.1.2 Availability / Churn rates Experimentation

The experimentation on node availability explores the behaviour of the Ad-hoc Edge Cloud under diverse churn rates. Namely, once the cluster is set up and a certain number of nodes become suddenly unavailable to the system. The overall intention of this experimentation is to comprehend how a highly dynamic environment with regard to nodes abandoning the system and therefore, in failure state from distributed storage perspective, affects the overall performance of Ad-hoc Edge Computing Infrastructure.

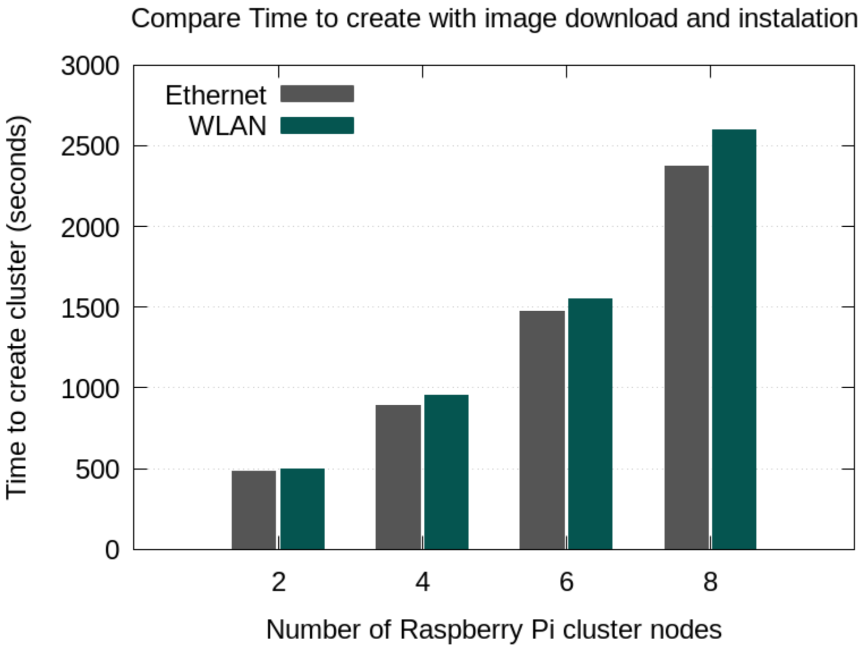


Figure 4.8: Cluster creation times considering Docker image installation.

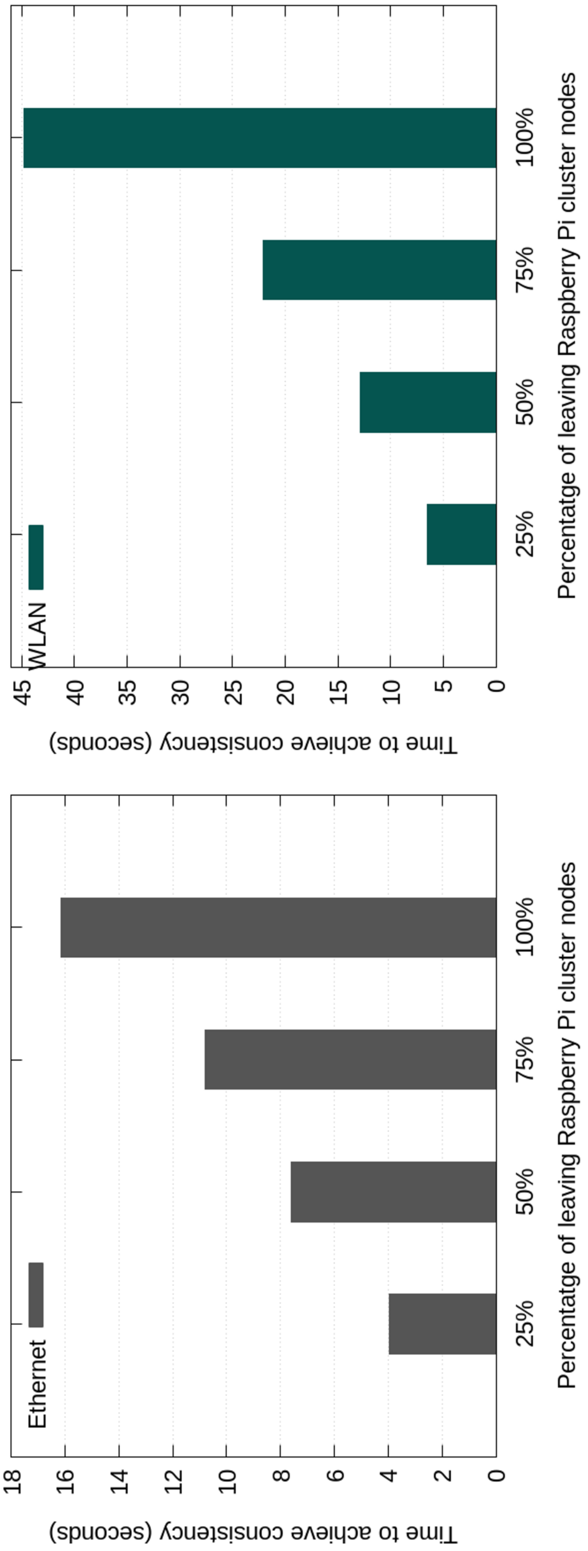


Figure 4.9: Time to recover from % node churn in 8 RPI Cluster.

In this set of experiments, we measure the time required for the Ad-hoc Edge Computing Infrastructure Nodes Distributed Registry to get to a consistent and healthy state once a given number of previously available nodes disappear. We define as consistent state the fact that all remaining nodes are in a healthy state concerning the cluster and the available data in the Nodes distributed registry is replicated among remaining nodes. It is paramount to underline that at this stage we are not considering migration processes of workloads in execution in this Edge node, instead we solely focus on the Node management processes. The analysis of recovery times from node churn has included the behaviour for 2, 4, 6 and 8 nodes suddenly abandoning the Ad-hoc Edge Infrastructure. These represent percentages of 25%, 50%, 75% and 100% (the disappearance of the cluster).

Figure 4.9 shows the obtained results for Ethernet and WLAN connectivity, we view this recovery time as the time necessary for the cluster to be in a consistent state once certain percentage of the nodes of the cluster leave the system. Following a logical correspondence with the rest of results gathered in this evaluation recovery times increment as the number of nodes disappearing from the system increase. Both from Ethernet and WLAN (Figure 4.9) tests it is apparent that the larger the number of abandoning members of the cluster, the longer times to recover. This is determined in the extreme cases which possess more that 75% of node churn, obtaining substantially long recovery times for the two last members remaining in the system

4.4.2 Large Scale Evaluation in AWS EC2

4.4.2.1 Scalability Experimentation

Experimentation in the AWS simulated environment focused on studying the behaviour of two distributed storage systems Apache Cassandra and Etcd in dynamic environments and at scale. The purpose of this validation has been to examine our initial technology choice based on Etcd and Raft consensus protocol. It is important to acknowledge that although both Apache Cassandra and Etcd are distributed storage systems, there are significant differences between them. Some examples of these are: Apache Cassandra is a fully-flagged database developed in Java to cover multipurpose application, whereas Etcd is a key-value distributed storage implemented in Go with the purpose of supporting multi-server configuration replication.

The first experiment is based on the latter by reproducing the process of creating from scratch a complete Ad-hoc Edge Computing Infrastructure cluster composed by 10, 20, 30, 40, 50 and 100 nodes, and response times obtained in seconds for this operation. Hence, it is immediately noticeable that creation times of Etcd clusters enhance results obtained by Apache Cassandra both in terms of time and necessary resources. In Etcd we obtain orders of magnitude of less than 1,4 minutes (87 seconds) to set-up a 100 nodes cluster, while necessary time in the same environment for a 10 nodes cluster for Apache Cassandra is 8,5 minutes. When checking resource consumption of the created nodes, the results are aligned: the average memory consumption of an Apache Cassandra node in a 10 nodes cluster is 328 Mb while results obtained for Etcd are 14Mb. Growth on cluster size follows a similar trend obtaining average memory usage of 100 nodes Etcd clusters inferior to 10 nodes cluster for Apache Cassandra. It has to be mentioned that tests for

Apache Cassandra were ceased after the creation of 40 nodes, as the cluster state remained unstable. Figure 4.10 illustrates this information.

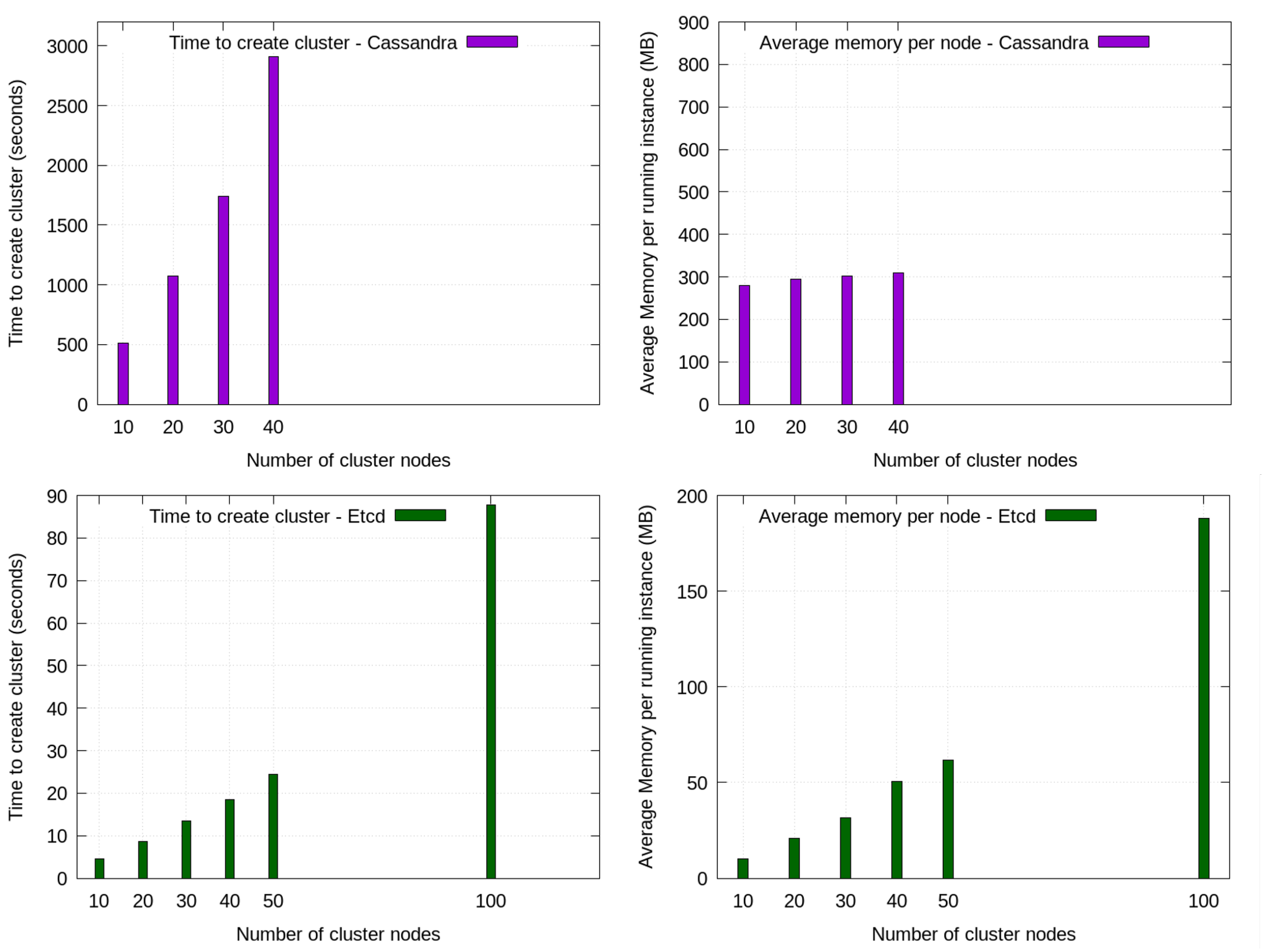


Figure 4.10: Large scale node creation experimentation.

It is essential to establish the fact that while the memory consumption for Apache Cassandra is above 280 Mb for a 10-node cluster, it remains steady on similar orders of magnitude as the cluster grows. Conversely, Etcd average memory usage per node increases with the size of the cluster. Average memory employed per node ranged from 10Mb for a cluster of 10 nodes to 188Mb in a 100 nodes cluster (see Figure 4.10). This is, by all means, explained considering the node synchronisation processes among data storage cluster nodes, however it raises an important aspect to take into account in future developments for our Ad-hoc Edge Cloud framework scalability. It is significant to remark that in Etcd the limit on memory consumption of our resource management tools stated in 32Mb to be usable in constrained Edge devices such as Raspberry Pis is in this environment achieved at a cluster size of 30 nodes.

Availability/ Churn rates Experimentation

This experiment employs a 10-nodes cluster and checks the time required to achieve the consistency state under diverse node churn rates: 20% churn rate representing 2 nodes becoming abruptly not available; 40% churn rate with 4 nodes abandoning; 60% with 6 nodes; and 80% corresponding to 8 nodes. The data obtained is illustrated in Figure 4.11. It shows that the time to recover, due to the data synchronisation processes, is linearly related with the number of nodes withdrawing from the system. As exemplary value, for Cassandra it takes 1,4 min for a 10 nodes cluster to recover from 20% nodes churn rate. Suffice it to say that this process is significantly more performant in Etcd, for which we have experimented 20%, 40%, 60% and 80% churn rates over 10 and 100 nodes cluster obtaining recovery times of 1 second for 20% churn rate in with cluster size of 10 nodes and 15 seconds for 100 nodes cluster. It is remarkable that Etcd is showing notably better recovery times for 80% churn rates than Cassandra for a 10 nodes cluster.

4.4.2.2 Large Scale Evaluation via AWS A1

In November 2018 Amazon Web Services announced the availability of its new EC2 A1 instances which offer for the first time ARM processor via the AWS EC2 Computing platform(“Amazon Web Services EC2 A1 Instances”, 2019; Barr, 2018b). AWS EC2 A1 instances were presented in diverse flavors ranging 1 to 16 vCPUs and 2 to 32Gb memory. With the aim of simulating the constrained nature of devices desired to used by this study, we have employed the more constrained image offered, the a1.medium, equipped with 2Gb memory, 1vCPU and 8Gb Disk for our validation. The aim of these tests has been to understand the behaviour of Etcd in Cloud resources which are more similar in hardware architecture and available resources to the ARM physical nodes to be used. Due to existing user quotas related to the recent launch of this service maximum number of simultaneous instances allowed to be created in AWS Ireland region used was 5. It is important to note, that this limitation was applied by AWS while having A1 service in Beta, and it is not related to any specific constraint in ARM architecture. In this constrained environment (accounting with a maximum of 4 nodes), we have performed initial tests over Etcd to validate the results obtained in our Lab environment with Raspberry Pis. Figure 4.12 show that creation times in A1 for same size cluster are in very similar orders of magnitude only accounting with differences of milliseconds for seconds and similar values in terms of used memory. The observable time difference in churn rates experimentation are due to the limited size of the cluster which minimise synchronisation processes among cluster members.

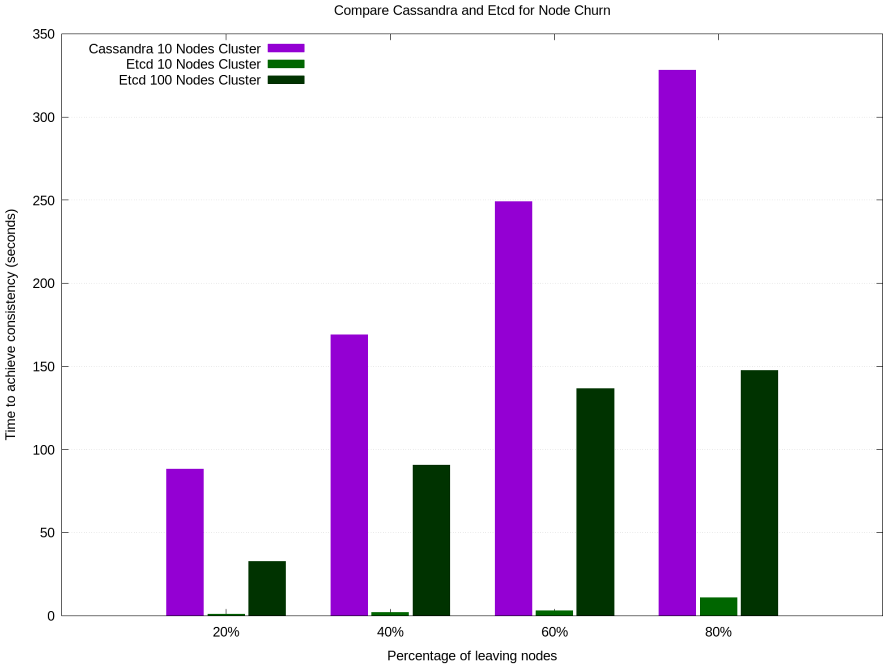


Figure 4.11: Time to recover from % node churn in 10 and 100 nodes cluster

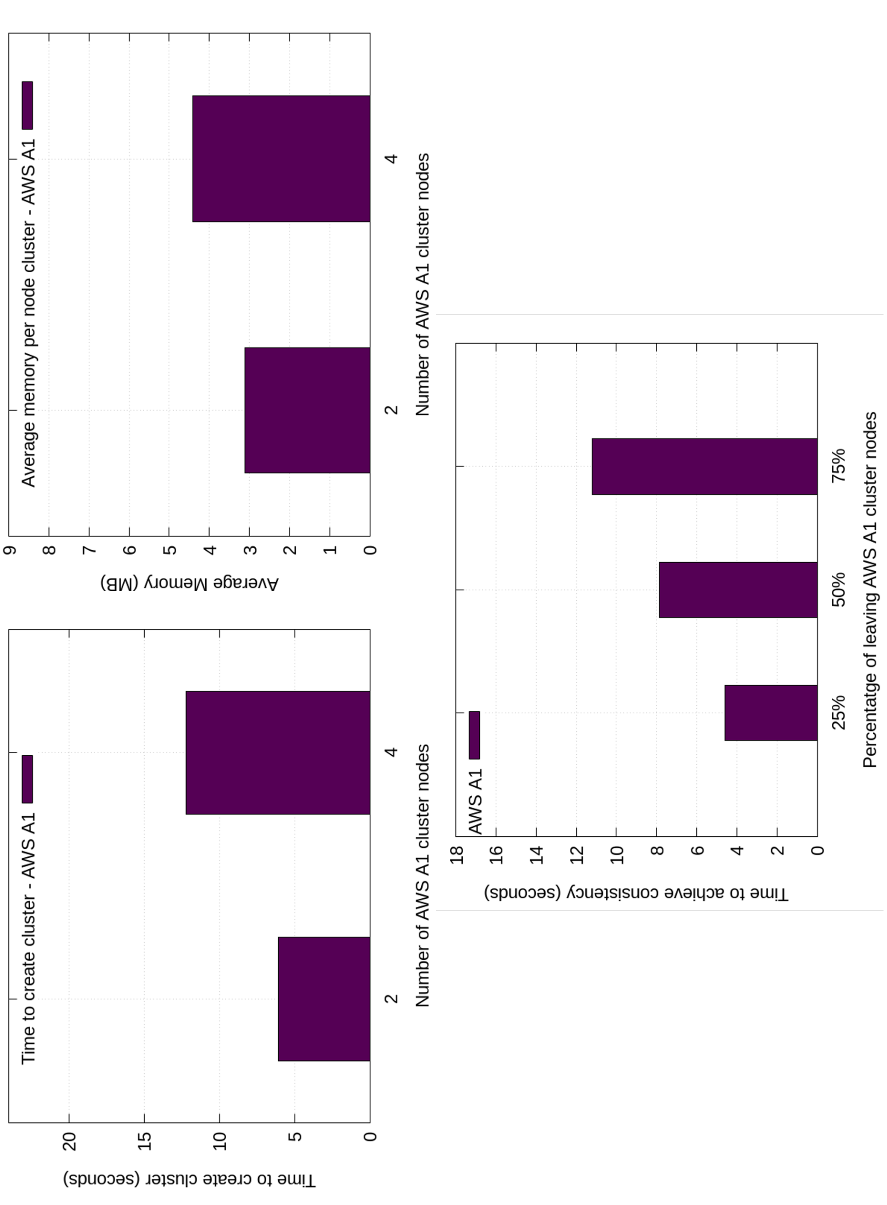


Figure 4.12: Experimentation over ARM-powered cluster in AWS

4.5 Conclusions

This section has presented the defined protocols for resource availability in the Ad-hoc Edge Cloud. It has also analysed the defined mechanism for Ad-hoc Edge Cluster instantiation and management relying on native capabilities offered by distributed storage systems.

The performed experimentation allows us to validate our initial technology choice of leveraging in Etcd for a key component of the Ad-hoc Edge Architecture for Resource management, the Distributed Storage.

Etcd has demonstrated outstanding performance in comparison to Apache Cassandra for this purpose. Etcd's baseline consensus algorithm enables Ad-hoc Edge Architecture to become fully decentralised and distributed, permitting the overall ambition to profit from the growing compute capacity at the Edge of the network in complex IoT devices, by dynamically forming out clusters of these devices in a decentralised and distributed manner.

The experimentation has also demonstrated that fault tolerance features of Etcd in terms of resource volatility are capable of coping with certain degrees of resource volatility when these are kept in volumes below to 75% of Nodes suddenly abandoning the system. This is a key lesson drawn from the applicability of this framework to use cases with situations of elevated levels of node churn can be expected. At current state of affairs, in scenarios which can account for volatility levels around 75% implications of the learned behaviour may imply the need of keeping a determined number of fixed nodes in order to achieve certain levels of quality in the services provided by the Ad-hoc Edge infrastructure.

In addition, this experimentation has reinforced the necessity of an initial registration phase in which the framework components are installed and registered into the target devices due to the performance overhead that installation of Docker component adds to this process, which get orders of magnitude of minutes, depending on network configuration.

Moreover, experimentation evidenced the relevance of scale both in terms of having the ability to support churn rates along with the management performance overheads it brings. These are directly related to the observed fact the Etcd clusters memory consumption is directly associated with the number of Nodes that participate in the cluster. In order to address this in future deployments of this technology, this is an essential consideration given that the larger the cluster is expected to be in terms of participating resources, the bigger these have to be dimensioned in order to address the observed scale overhead.

The findings of this chapter together with the architecture definition provided in Section 3.5 have been published in P2 (Juan Ferrer, Marqués, et al., 2019).

CHAPTER 5

Admission Control

5.1 Overview

Admission Control processes represent the decision which determines whether to accept a service to be executed in the infrastructure and if that is the case, to identify the set of the available IoT Edge resources which are most appropriate in order to place the different service components. The proper considerations regarding the level of Admission control mechanisms in Ad-hoc Edge Cloud are defined taking into account its past behaviour in terms of connections and disconnections, representing the source on which we base the resource availability prediction model for infrastructure which is the main research area of this chapter.

(Konstanteli et al., 2012) describes the admission control problem for Cloud computing as "the mechanism for deciding whether or not it is worth to admit a service into a Cloud, and in case of acceptance, obtain the optimum allocation for each of the components that comprise the service". Admission control and resource scheduling in loosely-coupled distributed systems such as Grid and Cloud computing systems have represented an extensively researched area (Bagchi, 2014; D. Chang et al., 2013).

Especially in Ad-hoc Edge computing Infrastructure, difficulties in the scope of Admission control for the Edge infrastructure of non-dedicated resources are not anticipated stemming from the characteristics of services to execute but due to volatility of resources which affects their availability, as described by Park (Park et al., 2011). In this PhD it is acknowledged that unpredictability of Edge resources (such as mobile devices) increments due to the following problems: unstable wireless connection, limitation of power capacity, low communication bandwidth and frequent location changes. These issues significantly increase levels of resource churn – a continuous process of resource enrolment and un-enrolment – that Ad-hoc Edge computing Infrastructure must handle.

To be more precise, two factors are considered to significantly influence service placement decisions in the context of Ad-hoc Edge Cloud:

Stability in the resource availability: Placement decisions in Ad-hoc Edge Cloud has to take into account the analysis of historical information on resource availability so as to determine if a service can be accepted, and which is be the most adequate assignment among available resources for this service by considering its requirements and the available resource' characteristics. Both Admission control and Service management are designed to support churn and subsequent resource volatility with the aim of ensuring as much as possible reliability of the overall infrastructure. With this aim, in this Thesis we employ the Node quality concept defined in (Panadero et al., 2018). Node Quality concept defines the predicted probability of a node be available in a certain time slot based on its historical behaviour of connections and disconnections to the Ad-hoc Edge Infrastructure.

Available battery levels in the contributed resources: Intrinsically related to Node's stability, Energy scarcity in IoT Edge devices is an issue largely studied in the context of IoT (Mousavi et al., 2017; Reinfurt et al., 2016) and Mobile Cloud computing (Juan Ferrer, Marquès, et al., 2019). Especially, as introduced in Chapter 2, energy optimisation of devices has been often used as motivation for off-loading computational loads to external clouds resulting in the production of energy savings in the context of Mobile Cloud Computing. Although this is not the main focus approach in this dissertation, it is clear that the available level of energy in the resource is a factor that cannot be neglected in the services placement decision to engage in the admission control, therefore this is a specific parameter Admission Control mechanism contemplates.

It is beneficial with a view to fully grasping the admission control issue to analyse the Service life-cycle in the Ad-hoc Edge infrastructure to introduce the phases of this lifecycle in which allocation decisions take place. Besides, we will introduce the Service Model we consider for Ad-hoc Edge Cloud. Afterwards we will present Ad-hoc Edge Cloud admission control and its associated mechanism to assess Node Quality based on the defined resource availability prediction model.

5.1.1 Admission Control in Service Lifecycle of Ad-hoc Edge infrastructure

Figure 5.1 describes the proposed Service Life-cycle in the Ad-hoc Edge infrastructure. This life-cycle provides an adaptation of the Life-cycle of a job inside a IaaS cloud defined by Ghosh in (Ghosh, Rahul and Trivedi, Kishor S. and Naik, Vijay K. and Kim, Dong Seong, 2010) extending it to the concept of Edge service as appliance considering multiple inter-related components.

This lifecycle describes the flow of actions from the time a specific IoT Edge device requests for service deployment until this service is up-and-running in the Ad-hoc Edge infrastructure. Actions in the operational lifecycle of services being executed respond to user requests.

In this context, it is observable that Admission control placement decisions crop up in two specific phases of this flow: at initial placement decision for the complete service and as a consequence of an adaptation decision, to allocate one or more service parts.

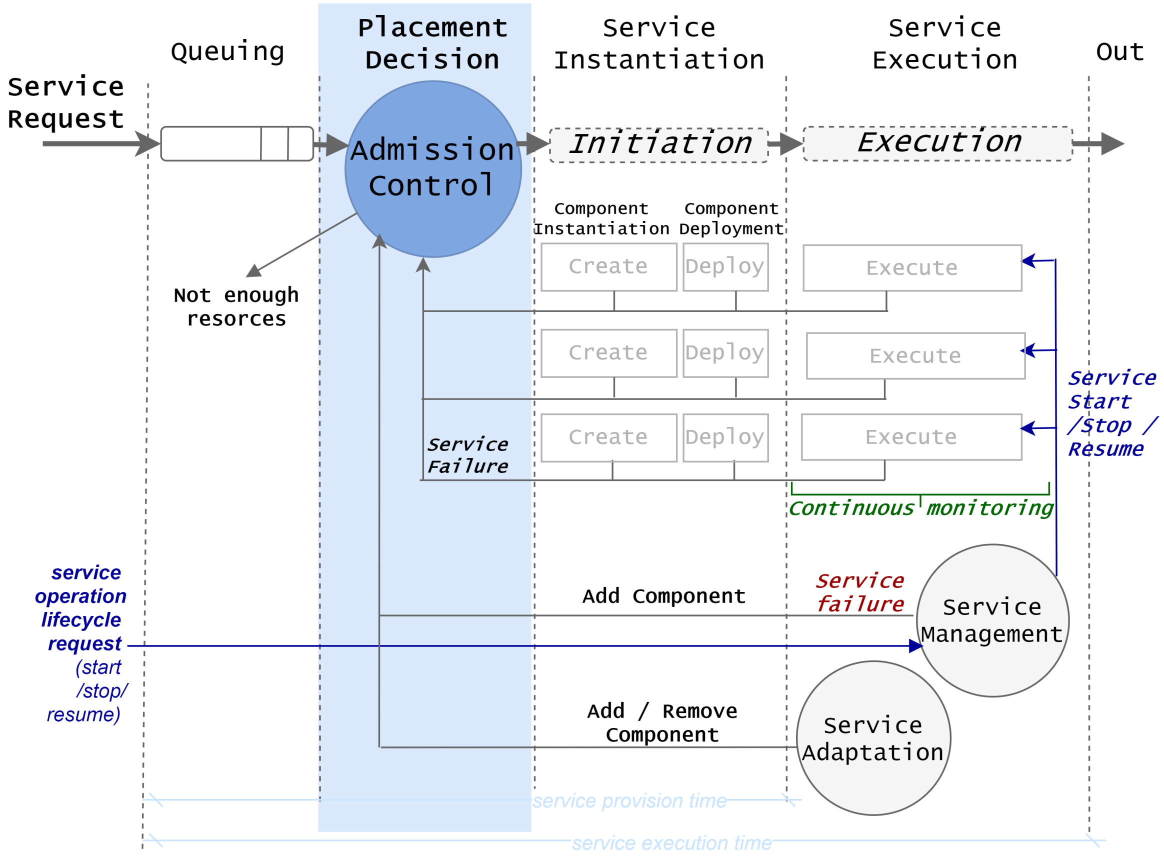


Figure 5.1: Ad-hoc Edge Service Life-cycle, Admission Control Role

5.2 Ad-hoc Edge Service Model

So as to ensure sufficient interoperability of the Ad-hoc Edge Computing infrastructure with existing Edge and Cloud offerings the resource model is illustrated in a generic manner which could be implemented using available service descriptors in standards and commercial offerings. Diverse examples exist of such service descriptors: in the Kubernetes architecture, these are represented by means of deployments and pods (“Kubernetes Pod Overview”, 2019) while Cloud vendors, such as AWS implement their own descriptors in AWS CloudFormation (“Amazon Web Services CloudFormation”, 2019) template and specific technologies like Terraform(Terraform.org, 2020) make use of service templates for this task. Independently of the language selected to express the workload execution characteristics, existing languages offer a similar structure which is represented in Figure 5.2 Ad-hoc Edge Computing Service Model.

More precisely, a service is composed of a series of components. The definition of a service is provided based on these set of components, the number of instances of each component, as well as, optionally a set of scalability rules. In its simplest form, a service has a single component. Each component belongs to a template, which determines necessary physical resources to allocate it. For each service component, its hardware requirements are expressed in its template. Common resources considered are the amount of disk, amount of memory and CPUs. Multiple instances can exist for a service component. The definition of the service indicates how many component instances will be instantiated when the service is deployed (minimum value of instances), as well as the maximum number of instances which can be created overall (as a result of the application of scalability rules). Each instance is deployed in the resource which has enough capacity to host its hardware requirements defined by its template. We name component creation to the instantiation of a significant component on an available resource. A deployment determines the necessary steps to have a service component up and running in a concrete infrastructure resource.

Typically, a service will be deployed as a set of containers described as service components. The use of virtualisation (i.e. OS or hypervisor based virtualisation) techniques provides isolation mechanisms among the user data and execution in hosting resources. It also allows transparent allocation of each component of the distributed service inside the Ad-hoc Edge Infrastructure. Moreover, it facilitates the process of horizontal elasticity, by adding or removing extra capacity for each component during runtime to maintain a certain level of performance for the overall service when variations occur in the workload.

5.3 Admission Control mechanism formulation

The Admission Control mechanism sorts out the problem of responding if a service s can be executed in an Ad-hoc Edge computing infrastructure E, and if so identifies the set of resources that can host it, R.

- If in the certain point of time in which the service execution request is done there are no hosts whose available capacity is sufficient to fulfil the execution requirements of the service, $R = \emptyset$.

- Otherwise, R is an ordered list of hosts h.

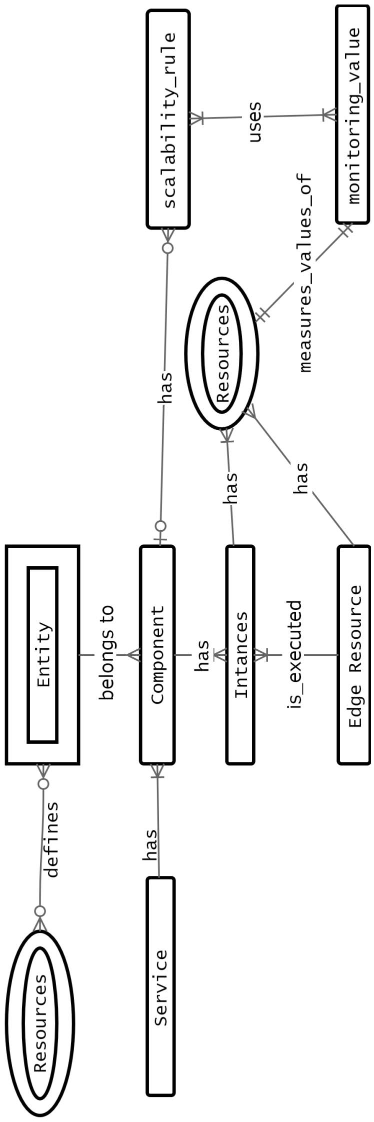


Figure 5.2: Ad-hoc Edge Computing Service Model

The Admission Control mechanisms considers that an Ad-hoc Edge computing infrastructure E is constituted by a set of computing nodes, hosts $H = \{1, ..., N_h\}$.

As introduced in Section 4.2, each participating resource in the Ad-hoc Edge computing infrastructure, a host h is characterised by two kinds of attributes: static and dynamic attributes. Among others, these attributes include: Computing Capacity by quantifying the number of available processors, their architecture and characteristics C_h; Available Memory M_h; Available Disk D_h; Available Battery level B_h and Quality of a node, Q_h. Q_h is based on the assessed node quality of a node based on its historical behaviour with regards to connections and disconnections to the Ad-hoc Edge infrastructure.

The Admission control considers that a service instance s to be executed in the Ad-hoc Edge Infrastructure belongs to a Service template which poses a set of requirements to the host to execute the service (see more details in Section 5.2). In its minimal form, these will include: Minimum computing capacity C_t; Minimum available memory M_t; Minimum available disk D_t; Processor Architecture Requirement $Arch_t$ and Operating System Requirement OS_t.

In order to admit a service to be executed in the Ad-hoc Edge Cloud, the Admission Control process executes two main steps: Filter and Ranking. Filter initial step is quantitative. It filters from all available resources those offering the sufficient device capacity in terms of dynamic characteristics (namely CPU, memory and storage) for the different parts that compose a service. The second step, Ranking, focuses on Qualitative characteristics. It prioritizes requirements according to critical aspects in resource management in Ad-hoc Edge Cloud infrastructure: node stability and node's current battery level. Node's stability is estimated employing the Node Quality value taking into account the time the device has been part of the infrastructure.

Table 5.1 presents the parameters considered in each IoT Device which takes part in the Ad-hoc Edge infrastructure and its role in Admission control mechanism while upcoming sections itemise the above mentioned two-step process.

Filter

The filter process determines from all available hosts $H = \{1, ..., N_h\}$, the set of host H' which available resources and characteristics meet the requirements of the service. These are set in terms of computing capacity, memory and disk are sufficient to host the service, as well as, the host conforms to the execution environment characteristics determined by the processor's architecture and operating system, according to the requirements exposed by the service template T. Therefore H' is the subset of H hosts which:

- Computing Capacity is greater than service s minimum computing capacity, $C_h \geq C_t$
- Available Memory is greater than service s minimum available memory, $M_h \geq M_t$
- Available Disk is greater than service s minimum available disk, $D_h \geq D_t$
- Processor Architecture corresponds to s requirement, $Arch_h = Arch_t$
- Operating System matches to s requirement, $OS_h = OS_t$

Parameter	Category	Type	Threshold	Rationale
% of connected time	Dynamic	Ranker, Quality	Maximize	Priories connection stability
# of disconnections	Dynamic	Ranker, Quality	Minimize	Penalize host with a higher number of disconnections
Computing Capacity, Memory and Disk	Dynamic	Filter, Capacity	Available values	If the host does not hold enough capacity to host the service, it is discarded
Processor Architecture	Static	Filter, Capacity	Defined host value	If the host does not posses the required value, it is discarded
Upload / Download speed	Dynamic	Filter, Capacity	Maximize	If the host does not include a certain value of upload/download network capacity, it is discarded
Battery level	Dynamic	Filter, Capacity	Maximize	If the host does not have a minimum level of battery, it is discarded
Battery level	Dynamic	Ranker, Quality	Maximize	Devices with higher battery level available are considered better.

Table 5.1: IoT Edge Device parameters classification

- And the available battery level is above a certain battery level threshold B_E established for the Ad-hoc Edge computing infrastructure E, $B_h \geq B_E$

If there are not hosts fulfilling the above-mentioned capacities and characteristics the $H' = \emptyset$. Therefore $R = \emptyset$ and the admission control process for service s resolves that the service cannot be hosted in the Ad-hoc Edge Cloud infrastructure at this point.

Ranker

If $H' \neq \emptyset$ the Ranker node returns the list of hosts $R = \{1, ..., N_R\}$ in H' ordered according to the assessed quality of the node, Q_h. As will be presented in detail in the upcoming section, Q_h value is calculated making use of historical values on resource availability to the infrastructure: percentage of connected time and number of disconnections since the initial registration of the host. In addition to this, in order to determine the overall Node Quality, the percentage of available battery of the host can be used with the purpose of balancing from all available hosts above a certain battery level, those with closest values to 100 percent battery levels and adequate historical behaviour, so as to give the current status of the device a prominent role in device selection.

5.3.1 Resource Availability prediction

As it has been previously mentioned, the Resource Availability prediction in this Thesis extends the concept of Node Quality presented in (Panadero et al., 2018). This builds on top of two parameters:

Percentage of connected time: which represents the percentage of time the node has been available to the infrastructure from its initial registration to the Ad-hoc Edge Cluster. This value is expressed as a value between 0 and 1, where higher percentages of connected time result in better node stability and reliability to the overall infrastructure.

Number of disconnections: Reflects the number of disconnections this specific node has had, normalised to the higher number of disconnections from all nodes currently available in the Ad-hoc Edge infrastructure. This parameter provides the sense of balance of the quality of the specific node behaviour for the rest of the available nodes in the infrastructure.

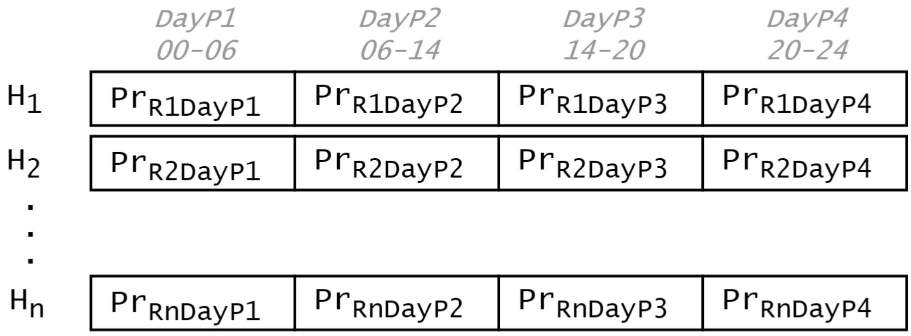
In more accurate terms, the Quality of a Node represents the probability of an IoT Edge Device to remain connected to the Ad-hoc Edge Infrastructure for a certain period (Panadero et al., 2018). A significant difference among this paper and (Panadero et al., 2018) as well as existing literature on Node availability prediction reported in Section 5.5, is the area of application which has been commonly present in Volunteer Computing environment. These utilise mechanisms to calculate availability prediction to forecast the behaviour of available peers in a matter of weeks, based on the behaviour on the different week days over the last month. The latter is motivated by to the roots of volunteer computing on using spare capacity of desktop computers in enterprise and home environments. This resulted in resource usage patterns in Volunteer Computing which outlines the distinction among working days for determining the time these resources have more probability of remaining idle.

Instead in Ad-hoc Edge Cloud environment we aim to interpret the behaviour of complex IoT devices located at the Edge of the network for determining periods in which these edge nodes exhibit more probability of having resources available for the Ad-hoc Edge infrastructure. In the interest of understanding this behaviour, there are significant difficulties at the current state of affairs: lack of real deployments of these specific scenarios for which we could gather data permitted us to discern the behaviour on the use patterns of the devices, and without leaving aside, the fact that this behaviour can significantly differ from specific usage scenarios we aim to address.

As far as IoT devices are concerned, usage patterns available literature in this field is particularly scarce (Mousavi et al., 2017; Poyraz & Memik, 2016; Reinfurt et al., 2016). We have not been able to identify any references to specific time usage patterns. The entirety of the analysed literature mainly focuses on the use of energy of devices and is strongly oriented to a specific type of IoT device. By way of example, the closest device to the kind of IoT Edge device we aim to employ is in (Poyraz & Memik, 2016). This work focuses on analysing the behaviour of SmartWatches' users while observing usage patterns of 32 users in 70 days of continuous use. This analysis concludes with the general statement that SmartWatches devices are in an idle state 89 percent of the time.

Smartphone usage patterns have been studied in depth, however, the number of analyses on mobile phone usage patterns which include references to idle time and time use patterns is also limited (Bhih et al., 2016; Shye et al., 2010; Van Canneyt et al., 2019). Findings of the available studies have reached the conclusion that a phone is largely in an idle state 89 percent of the time(Shye et al., 2010) (note that percentage is the same range as the assessed for smartwatches in (Poyraz & Memik, 2016)). (Shye et al., 2010) has analysed the activity of users on smartphones logs of 25 users over 6 months. When focusing on weekday behaviour, overall differences between week and weekend days do not appear as significantly different. (Bhih et al., 2016) examines patterns for mobile data traffic. It studies 3.3 terabytes of data gathered by the University of Cambridge using Device Analyzer. This information contains over 100 billion records of 17,000 android devices worldwide and extends from December 2010 to January 2014. These do not reveal any significant difference in Mobile data traffic between the weekend and working days. Otherwise, (Van Canneyt et al., 2019) collects information from 230K mobile apps and 600M daily unique users in 221 countries. (Van Canneyt et al., 2019) focuses on application sessions, concluding that during the weekend, application sessions seem to start and finish later, with a margin of one hour.

A striking aspect to be considered is the fact that these studies also examine mobile phone day hour patterns from different perspectives: (Van Canneyt et al., 2019) shows that mobile users are more active on the mobile device throughout the day, having the maximum activity during evenings locating the activity peak at 9pm. (Bhih et al., 2016) shows a similar pattern, stating that although the data traffic is quite stable during the day hours, it significantly increments and peaks around 22:00 hours. Finally (Xu et al., 2011) studies the temporal distribution of the traffic of apps focusing on more specific temporal patterns. It also reveals diurnal patterns concerning traffic volume and network access time. In contrast to previous papers, it locates the minimum use around 1AM and 2AM. This study shows that traffic volume starts increasing around 4AM and hits its peak around



	DayP1 00-06	DayP2 06-14	DayP3 14-20	DayP4 20-24
H_1	$Pr_{R1DayP1}$	$Pr_{R1DayP2}$	$Pr_{R1DayP3}$	$Pr_{R1DayP4}$
H_2	$Pr_{R2DayP1}$	$Pr_{R2DayP2}$	$Pr_{R2DayP3}$	$Pr_{R2DayP4}$
⋮				
H_n	$Pr_{RnDayP1}$	$Pr_{RnDayP2}$	$Pr_{RnDayP3}$	$Pr_{RnDayP4}$

Figure 5.3: Representation of Data Structure for Disconnection Probabilities

noon. After 3PM it decreases, to arrive at its minimal value after 8PM.

Intending to overcome the above-mentioned difficulties related to the availability of data to establish resource IoT Edge resource usage data, we have decided to centre our prediction model on the usage patterns of mobile phones, whose patterns on usage data are at this stage much more stable. Mobile phones are the most widespread devices available at the Edge the network. Figures on Edge devices penetration also present analogies in terms of scale with the expected distribution of IoT Edge devices.

Resting on the patterns that we could extract on mobile phone usage patterns from the data available in the previously identified works(Bhih et al., 2016; Van Canneyt et al., 2019; Xu et al., 2011), we have determined that the appropriate period to study in the context of Ad-hoc Edge Cloud. In our case, the time period to analyse is the behaviour of the IoT Edge device during the previous week. In addition, it is necessary that we consider four different time-slots along the day to inspect connection and disconnection patterns.

The timeslots we have taken into consideration comprise the following intervals:

– Day period 1 (DayP1): From 0:00:00 to 5:59:59

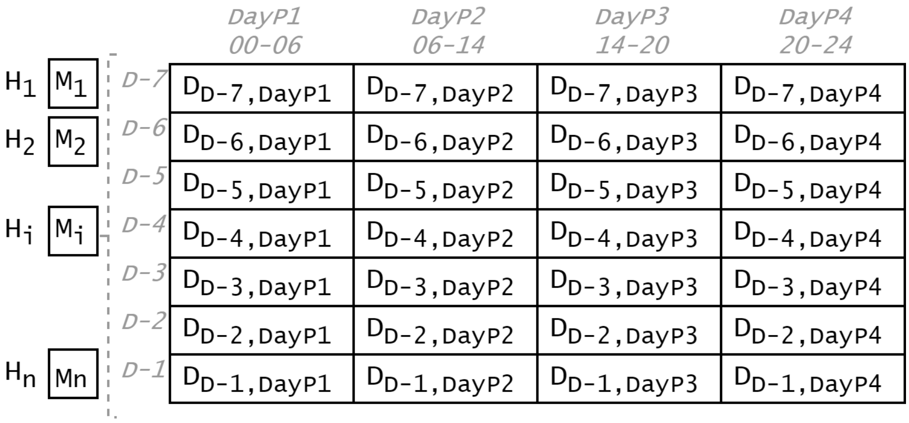
– Day period 2 (DayP2): From 6:00:00 to 13:59:59

– Day period 3 (DayP3): From 14:00:00 to 19:59:59

– Day period 4 (DayP4) From 20:00:00 to 23:59:59

Hence, in order to predict the probability of an IoT Edge Device to continue to be part of the Ad-hoc Edge Infrastructure we will employ the data on its behaviour during the previous week, gathering the data according to the four intervals in which we have divided each of the days, and obtaining a probability of a node to remain connected for a given period.

Therefore, for each host h we keep a data structure of disconnection probabilities as an array of 4 positions which stores host's probability of disconnection of each day period (host $h \in \{1, 2, ...N\}$, $Pr_{hDayPeriodp}$ day period $p \in \{1, ..4\}$) calculated by the prediction model using the information of the last 7 days. See representation in Figure 5.3.



	DayP1 00-06	DayP2 06-14	DayP3 14-20	DayP4 20-24
D-7	$D_{D-7,DayP1}$	$D_{D-7,DayP2}$	$D_{D-7,DayP3}$	$D_{D-7,DayP4}$
D-6	$D_{D-6,DayP1}$	$D_{D-6,DayP2}$	$D_{D-6,DayP3}$	$D_{D-6,DayP4}$
D-5	$D_{D-5,DayP1}$	$D_{D-5,DayP2}$	$D_{D-5,DayP3}$	$D_{D-5,DayP4}$
D-4	$D_{D-4,DayP1}$	$D_{D-4,DayP2}$	$D_{D-4,DayP3}$	$D_{D-4,DayP4}$
D-3	$D_{D-3,DayP1}$	$D_{D-3,DayP2}$	$D_{D-3,DayP3}$	$D_{D-3,DayP4}$
D-2	$D_{D-2,DayP1}$	$D_{D-2,DayP2}$	$D_{D-2,DayP3}$	$D_{D-2,DayP4}$
D-1	$D_{D-1,DayP1}$	$D_{D-1,DayP2}$	$D_{D-1,DayP3}$	$D_{D-1,DayP4}$

Figure 5.4: Representation of Data Structure for Storing the disconnection patterns per day period of the last 7 days

In addition to this, we keep data structure to collect disconnection patterns during the previous week of execution for every node. This maintains the disconnection patterns of all nodes in the last 7 days. The data structure used is represented in Figure 5.4. It is a matrix of 7x4, which will store for a period of one week, the number of times a node has changed the state (disconnections and connections) per each of the four day periods.

In detail per each host there is a matrix M_h which represents the seven position matrix corresponding to the node h. D is the Day number $d \in \{1, ..7\})$ and day period is $p \in \{1, ..4\})$. $D_{d,p}$ is the number of disconnections experienced by host h in day d period p. This matrix is kept updated by a daily process that each day analyses the performance of all nodes, updating the number of disconnections that each host experienced in the previous day aggregated by each of the four periods of the day. This process resorts to the monitoring information kept by each node.

Therefore, we define the probability of disconnection of a host h in a given day period as analogy to (Panadero et al., 2018) as:

$Pr_{h,p} = normalise(\frac{\sum_{d=D-7}^{D} D_{d,p}}{7})$

By using this probability we can calculate employing of Algorithm 1 the connection probability of a certain host in a given period as the inverse probability of a node from its initial registration, represented by $t_{registration}$ and the next t_{units} as:

$Pr_{h,t_{units}} = 1 - (DiscProb(h, t_{units}) * DiscProb(h, -t_{registration}))$

Disconnection probability results into the estimated Node Quality. This value allows us to have a measurement of the node's stability as part of the Ad-hoc Edge infrastructure for a given time period based on its past behaviour.

5.4 Evaluation

Evaluation has focused on the assessment of the resource availability prediction model previously presented in section Section 5.3.1 as part of the Admission control

Algorithm 1 Disconnection Probability

```
procedure DiscProb($h, x$)
    $nPeriods \leftarrow numTimePeriods(x)$    ▷ Number of time periods within x
    $sum \leftarrow 0$
    if $x > 0$ then
        for $i \leftarrow 1$ to $i = nPeriods$ do
            $p \leftarrow period(now() + i)$
            $sum \leftarrow sum + Pr_{h,p}$
    else
        for $i \leftarrow 1$ to $i = nPeriods$ do
            $p \leftarrow period(now() - i)$
            $sum \leftarrow sum + Pr_{h,p}$
    return $sum/nPeriods$
```

mechanism. This validation assesses the ability to predict node behaviour as the probability of a node to stay connected to the Ad-hoc Edge infrastructure in a certain period of the day as essential building block for the defined Ad-hoc Edge Admission control mechanism as the procedure that determines Node Quality. The overall aim is to evaluate separately this key mechanism that allows us to ensure service placement in more reliable and stable nodes in the dynamic churn-prone environments expected in the Ad-hoc Edge infrastructure. With this purpose we have developed a set of experiments utilising a distributed large-scale resource allocation simulator (Panadero et al., 2018).

This simulator allows us to represent large-scale and high dynamic and heterogeneous environments with diverse rates of node churn and diverse nodes configurations by means of the definition of different kinds of nodes that determine certain quality level (High, Medium and Low) for each node. Nodes of quality High have high probability of re-connection and low probability of disconnection. Nodes defined as Low quality show the opposite behaviour. The distributed large-scale resource allocation simulator is implemented in Java (SE 7).

For all the experiments we have represented three different experimentation scenarios which correspond to three different distributions of kinds of nodes for the experiment, namely the three experimentation scenarios used for experimentation are: Most_Nodes_High, Most_Nodes_Medium and Most_Nodes_Low. In each of these three scenarios we assign a certain percentage of nodes of each quality over the total experiment number of nodes. These percentages determine the number of nodes of each node quality (*High*, *Medium*, *Low*) that will be present during the execution of the experiment. The configured node type determines the node quality, therefore defines its behaviour during experimentation with regards to number of disconnections and probability of re-connection.

The three defined experimentation scenarios are represented in Table 5.2. To be more precise, in the scenario Most_Nodes_High for each experiment a 60% of nodes are configured of type High, while a 20% of generated nodes correspond to Low and Medium quality nodes. Scenario Most_Nodes_Medium represents an execution environment in which 60% of nodes are of Medium quality and 20% to are of High and Low quality. Similarly, Most_Nodes_Low account of a 60% of Low quality nodes and 20% of High and Medium nodes quality.

Attempting to fully assess the impact of scale both in terms of nodes and

Scenario	**Percentage of nodes with *High* quality**	**Percentage of nodes with *Medium* quality**	**Percentage of nodes with *Low* quality**
Most_Nodes_High	60%	20%	20%
Most_Nodes_Medium	20%	60%	20%
Most_Nodes_Low	20%	20%	60%

Table 5.2: Scenarios: Percentage of Nodes of *High*, *Medium* and *Low* quality per scenario

services we have performed the experimentation the execution of up to 200, 400, 600, 800 and 1000 services over 100, 200, 300, 400 and 500 nodes of the three different experimentation scenarios described before. The number of nodes present in each experiment of each node type is determined by the percentages over the total number of nodes defined in each execution scenario (Most_Nodes_High, Most_Nodes_Medium and Most_Nodes_Low). All executed services have been constructed with the same configuration, with two replicated components. Overall, the execution of these different service and node combinations have resulted in the execution of 75 experiments. The findings of this experimentation have been summarised in the following subsections.

5.4.1 Node Quality

Figure 5.5, Figure 5.6 and Figure 5.7 exhibits per each of the three experimentation scenarios (Most_Nodes_High, Most_Nodes_Medium and Most_Nodes_Low) the total number of nodes created by the experiment of nodes of each node quality (Number of nodes per Quality), the number of disconnections of each node quality (Disconnections per Quality) and the time each type of node has been connected during for the execution of 600 concurrent services (Connection Time per Quality).

More specifically each figure represents per each scenario (Most_Nodes_High, Most_Nodes_Medium and Most_Nodes_Low) in the first row Number of Nodes, the actual number of nodes of each quality type generated by the simulator in each of the three defined nodes qualities (High, Medium and Low) when creating environments of 100 to 500 nodes. For instance, in the experiment which corresponds to scenario Most_Nodes_High with 500 nodes, 300 nodes are created of quality High, this means these nodes have low probability of disconnection and high probability of re-connection. At the same time, additional 20% of total experiment nodes (100 nodes) are created with qualities Low and Medium, exposing their corresponding connection and re-connection patterns. This behaviour is observable in the number of disconnections of nodes per quality (Disconnections per Quality, second row) and evidenced the overall connection time (Connection Time, third row) this node quality presents in the experiment.

Additionally, Figure 5.5, Figure 5.6 and Figure 5.7 display in the bottom row the values of battery level, a dynamic dynamic node characteristic defined in our IoT Edge Device parameters presented in Table 5.1 and Section 4.2 classification used in these experiments. As "Percentage of Nodes with Battery" we represent number of nodes with average battery level below and above 50 percent created in the corresponding scenario.

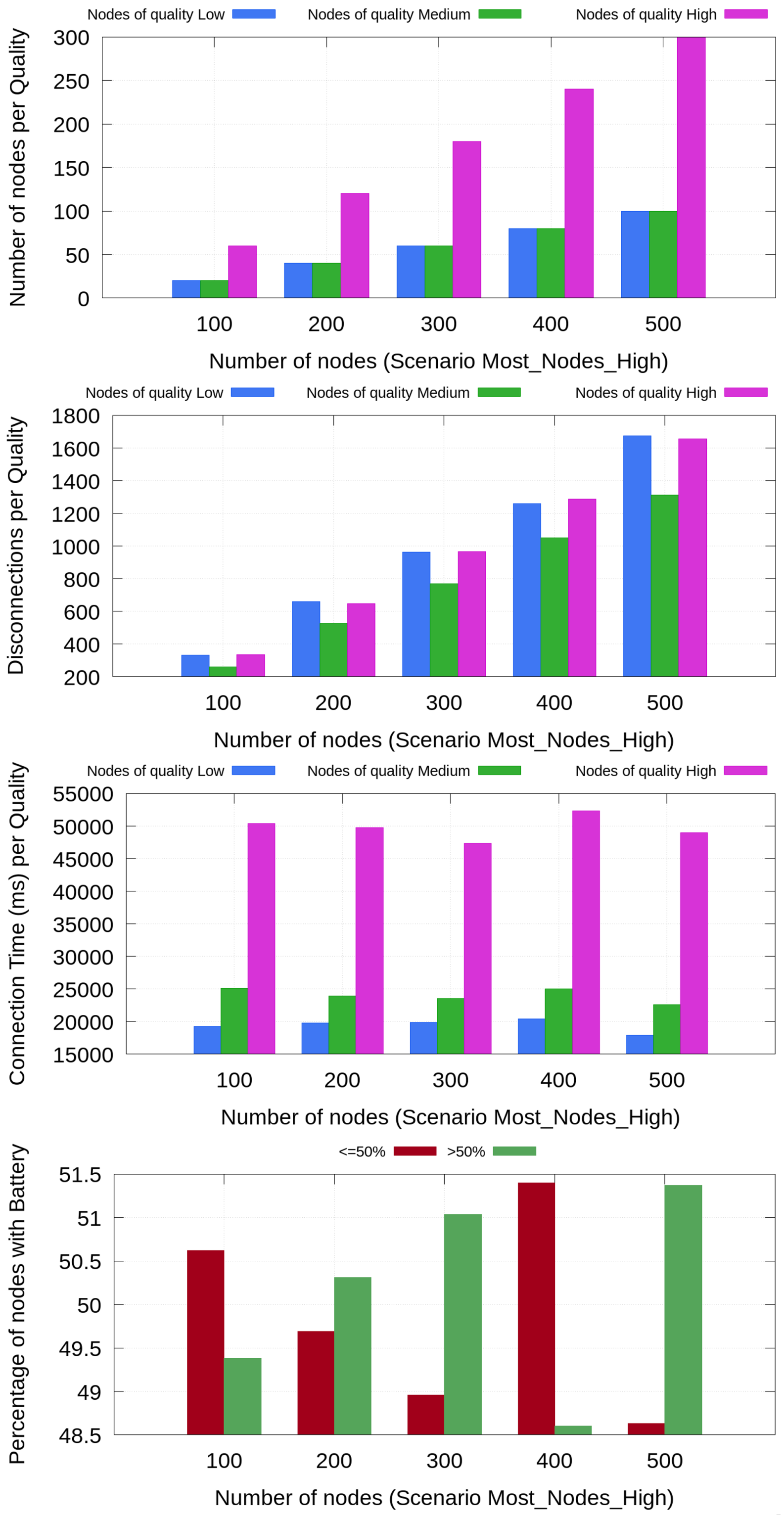


Figure 5.5: Most_Nodes_High Experimentation Scenario

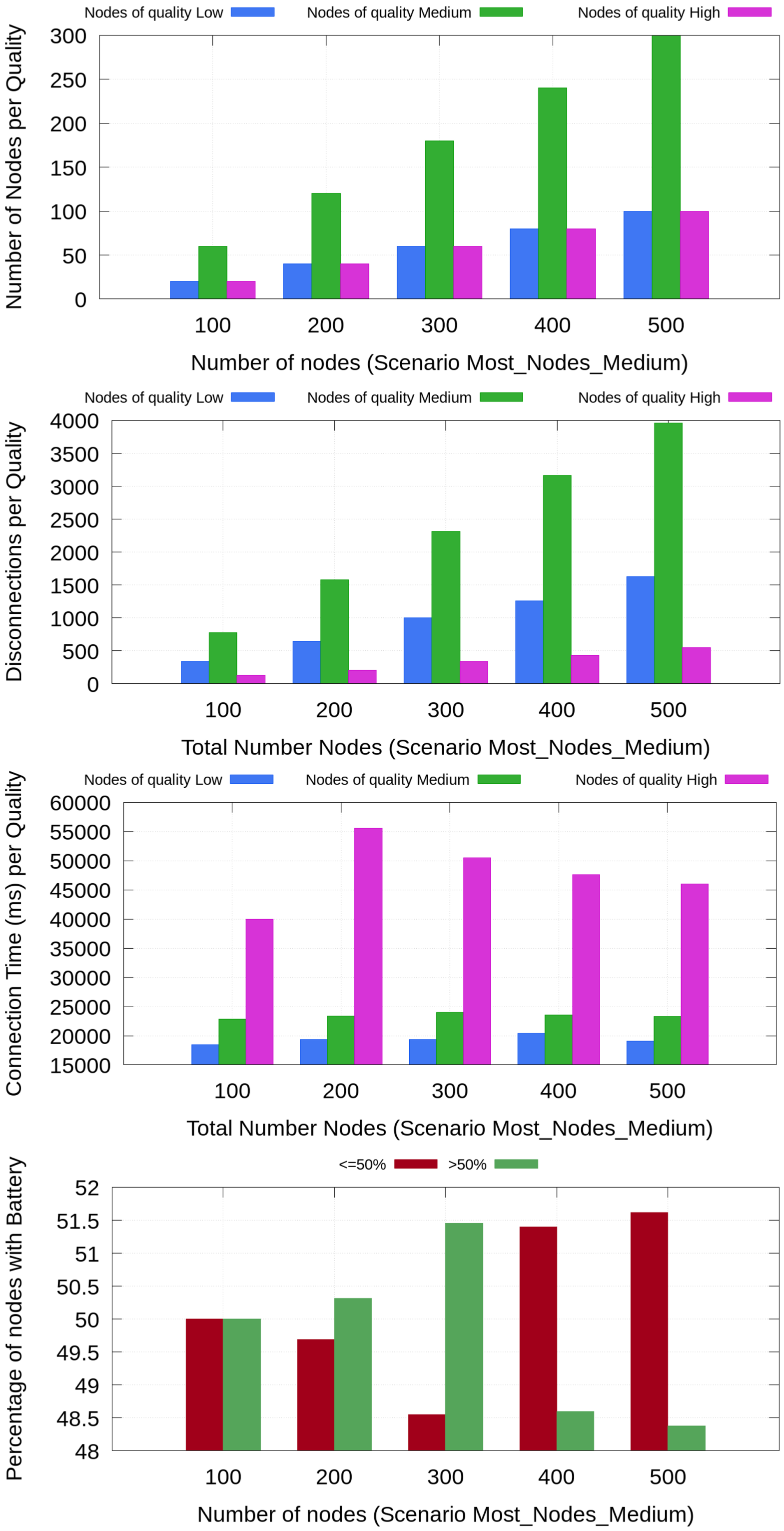


Figure 5.6: Most_Nodes_Medium Experimentation Scenario

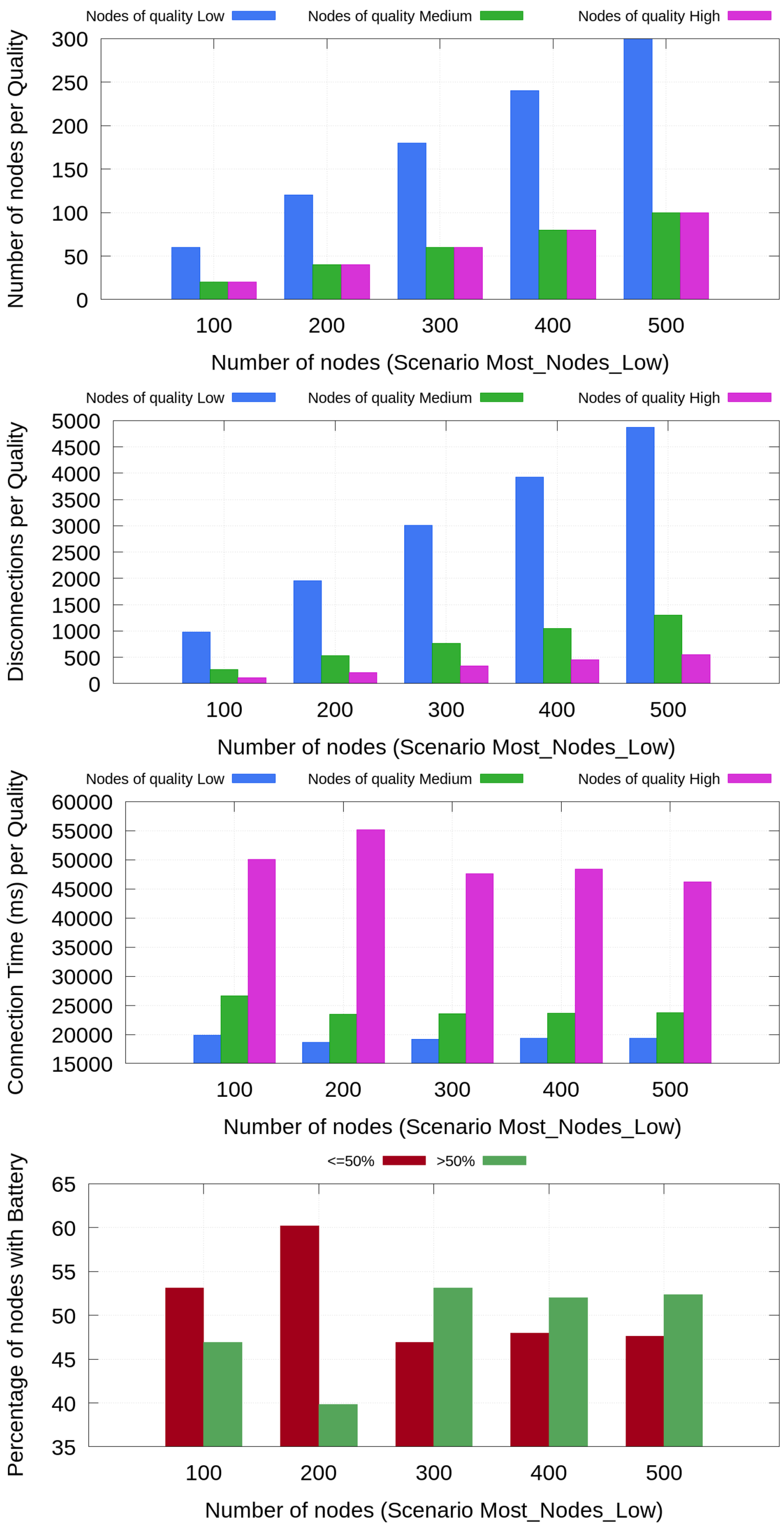


Figure 5.7: Most_Nodes_Low Experimentation Scenario

Overall, Figure 5.5, Figure 5.6 and Figure 5.7 are meant for representing the execution environment generated for each of the three defined Experimentation scenarios which was used for the Node Quality experiments, as the measure that allows us to assess the stability of a node. In this execution environment, we aim to represent the different nodes behaviours in each scenario, as well as, the heterogeneity on the characteristics of nodes taking part in the Ad-hoc Edge infrastructure.

Relying on the previously presented scenarios, tests performed with the execution of 600 services over 100 to 500 nodes obtain nodes qualities aligned to the defined nodes behaviours.

Node quality represent the value between 0 and 1 which is obtained by the resource availability prediction model as calculation of its disconnection probability considering node behaviour with regards to connections and disconnections, as introduced in section Section 5.3.1.

Results obtained in these tests are depicted in Figure 5.8. In the scenario Most_Nodes_High, nodes obtain Node Qualities values between 0.86 and 0.90 while these qualities descend to values that range among 0.80 and 0.88 in for the scenario Most_Nodes_Medium (with 60 percent presence of medium quality nodes), and among 0.77 and 0.82 for the high presence of low-quality nodes in scenario Most_Nodes_Low. It is important to note that the performed experimentation show for 600 services execution increments of Node Quality of percentages among 3 and 5 for scenarios Most_Nodes_High, Most_Nodes_Medium and Most_Nodes_Low, however keeping coherent values for probabilities of disconnection and its associated Node Quality independently of the number of nodes being employed in the experiment.

This assessment is further demonstrated in the analysis of the performance of obtained average Node Quality values by generalising services' execution figures from 600 services to the execution of 200 to 1000 services over 100 to 500 nodes (Figure 5.9) for the three defined scenarios. We can observe once more that both node scale and services scale do not influence the obtained values of Node Quality. Consequently, we conclude that Node Quality is solely determined by the specific node characteristics and its historical behaviour concerning connection and disconnection to the system.

5.4.2 Service Quality

The quality of a Service is assessed by the employed large scale simulation environment(Panadero et al., 2018) as the sum of the Node Quality values of the different nodes that execute a service together with normalised values for nodes characteristics. The configuration of Ad-hoc Edge experimentation has considered the execution of two replicas of each service representing two identical service components.

In addition to this, the assessed Service Quality takes into account the characteristics of the normalised values per each node for computing capacity, memory, and available disk, available upload and download connectivity speed and battery level.

The values gathered during this experimentation for Service Quality obtained on the execution of 200, 400, 600, 800 and 1000 services over 100 to 500 nodes are illustrated in Figure 5.10. We observe that maximum Service Quality value is collected on the execution of 1000 services over 500 nodes accounting for 4.5 and

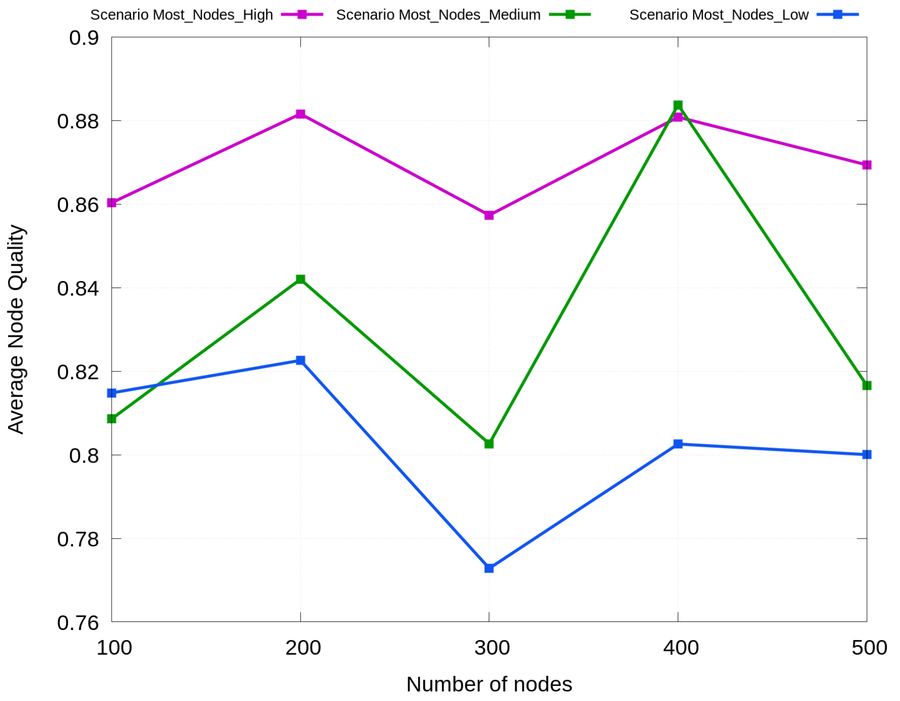


Figure 5.8: Nodes Quality per scenario

minimum values of 2.2 are found in the execution of services with execution scenario Most_Nodes_Low, with more presence of low quality nodes. This behaviour is again aligned with previous conclusions on Node Quality and allows us to infer that based on the results of our experimentation the presented Ad-hoc Edge Resource management and its associated mechanism for Node Quality prediction is adequate for managing the expected scale and heterogeneity expected in such dynamic and distributed environment.

5.5 Related Works

Resource Availability prediction has been an area previously studied in Contributory communities and Volunteer Computing. These computing systems have the objective of taking advantage of spare compute capacity of desktop computers at home or offices for individual resources or for groups of them. In order to do so, they take into account desktop computers, hosts, usage patterns in terms of connections and disconnections. Differently, devices tackled in the context of Ad-hoc Edge Cloud are IoT Edge devices. IoT Edge devices probability of node churn require that the mechanisms to predict IoT device node availability evolve in order to cope with higher dynamicity rates on resource availability, as well as, to define mechanism that permit us to discern usage patterns in IoT Edge devices environment.

To elaborate on this topic, in the context of volunteer computing (Panadero

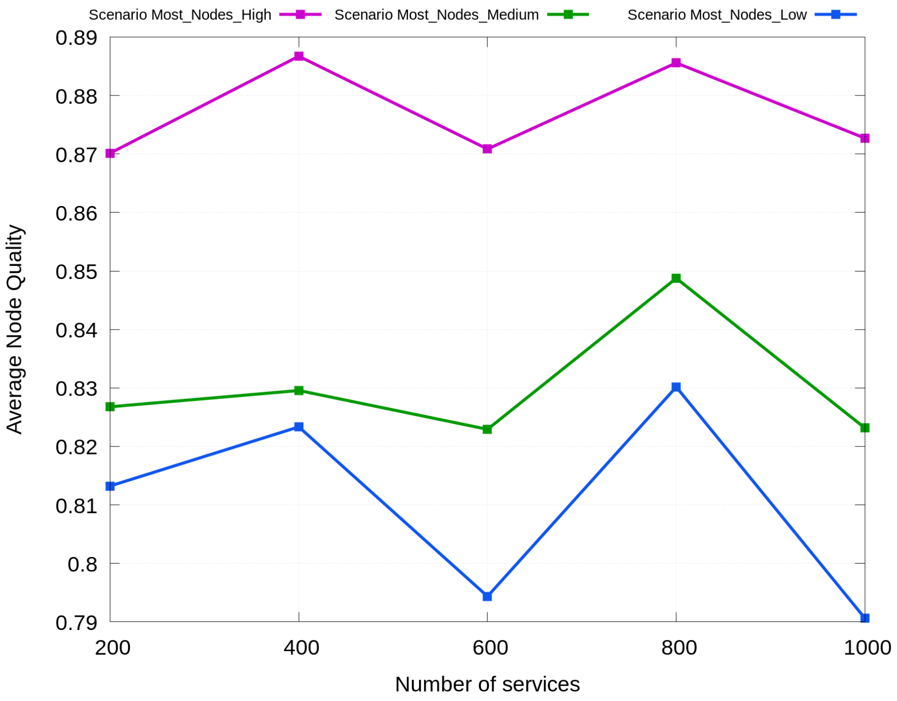


Figure 5.9: Average Nodes Quality in 100-500 Nodes executing 200-1000 services

et al., 2018) provides a multi-criteria optimisation strategy for selection of reliable nodes in large scale systems based on a Multi Criteria Biased Randomized method. (Panadero et al., 2018) prediction of node availability capitalises on the behaviour of nodes regarding the connections and disconnections over the past four weeks to calculate the probability of disconnection of a node in a given weekday. Along these lines nonetheless using a different granularity level for representing the availability of a host (Kondo et al., 2008; Lázaro et al., 2012) employs the availability of the hosts per hour for the previous week to represent resource availability. These two papers had available existing traces of 11000 hosts in SETI@HOME collected by means of the instrumentation of BOINC server during seven months period (Kondo et al., 2008). This aspect has allowed for the understanding of the devices' usage patterns, determining day of week and hour of week hosts connection patterns for the constituent hosts in these Contributed Desktop Grids environments.

Our work in Ad-hoc Edge Cloud environment advances these previous works in Desktop Grids and Volunteer computing in the time granularity employed to build the model, elaborated in matter of weeks in these environments and being evolved to day periods for Ad-hoc Edge Clouds, as we foresee a more dynamic behaviour of complex IoT devices. In addition, a significant differentiation among resource availability prediction in Volunteer Computing and IoT Edge environments is the availability of data that permits to understand the resource usage patterns.

The fact that IoT Edge devices are not yet so widely spread hinders at this stage the availability of data to be analysed to determine connection and disconnection

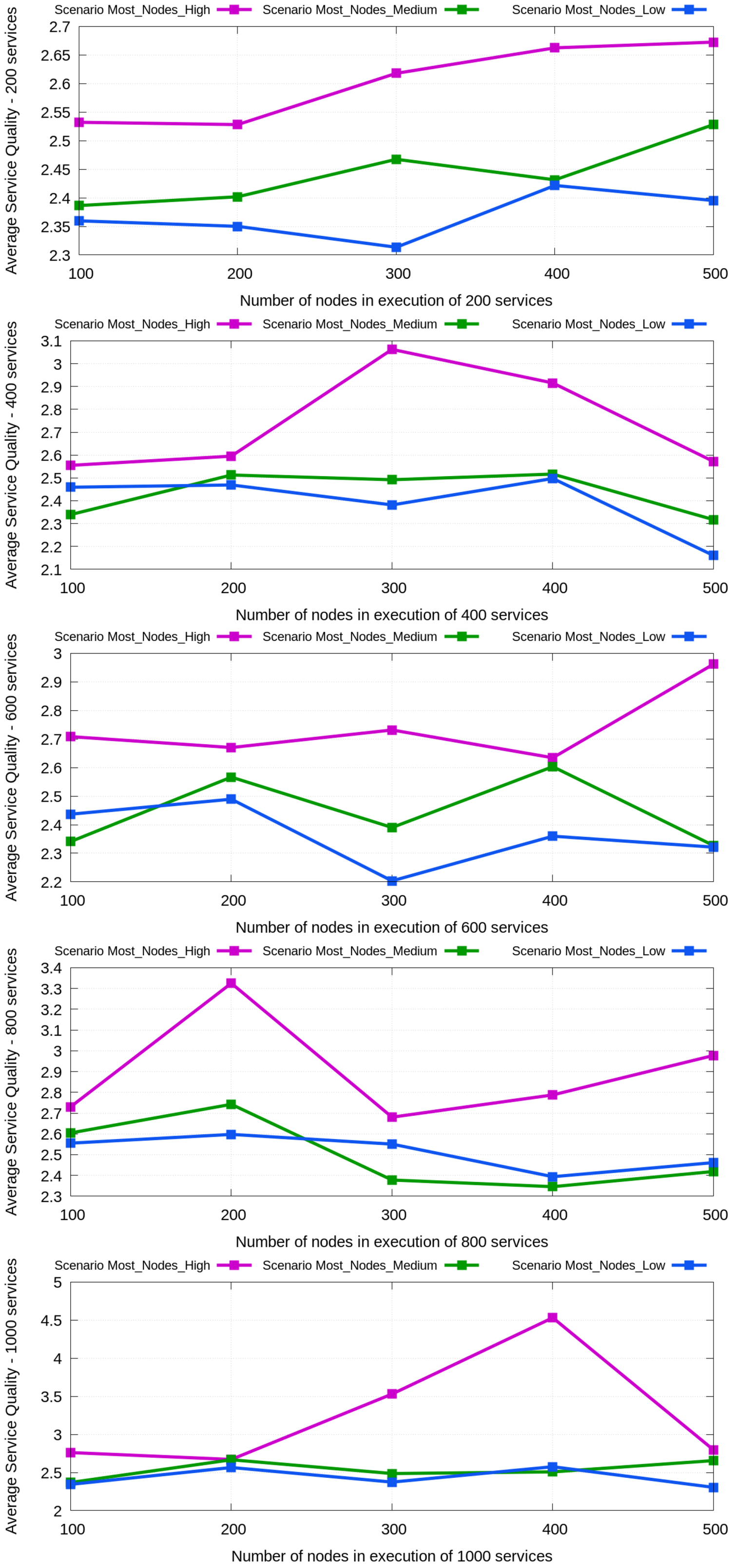


Figure 5.10: Services Quality per scenario

patterns for these kinds of devices. In the interest of overcoming these difficulties, we have extrapolated the behaviour of available specific edge devices, mobile phones, and draw on its common usage patterns in order to describe our resource availability model. This has determined us to define a mechanism which instead of focusing on connection and disconnection patterns over weekdays we have employed the resource historical behaviour over four different day periods in a week therefore being capable of providing probability of node disconnection for certain day hours instead of week days in volunteer computing. Moreover, we understand that in the long term, as IoT and Edge deployments become more widely available and data on devices usage patterns is accessible, this model could be further refined in order to consider not only even the specific behaviour per type of complex IoT Device and could be tailored for particular IoT usage scenarios.

In the area of Fog and Edge computing scheduling and placement, only recently some works have started to envelop the aspects of edge node availability and churn. We understand this is due to the fact that so far typically Edge computing environments are not yet considering IoT Edge devices as appropriate execution environments for Edge service execution. Therefore, their dynamic behaviour in which this Thesis focuses on is yet generally not considered. Some examples of papers which take into consideration IoT Edge device resource availability issues and heterogeneity aspects are presented below.

Broggi(Brogi et al., 2020) recognises that IoT and Fog nodes mobility and dynamicity have not yet been addressed to a greater extent and acknowledges the emerging need of addressing application placement adapted to the phenomenon of node instabilities at the Edge, as elaborated by Ad-hoc Edge Cloud admission control mechanism. Ad-hoc Edge Cloud Admission Control processes selects from the existing pool of resources in a determined point in time, those assessed to be the existing set of resources which better serve the service execution request. This mechanism establishes a two step process in which resources are first filtered according to their dynamic and static features considering the service execution needs. Results of the filtering process are then ranked according to assessed Node Quality which assesses the probability of an IoT Edge Device to continue to be part of the Ad-hoc Edge Infrastructure for a certain period time. Therefore, Admission control in Ad-hoc Edge Cloud addresses existing challenges in relation to resource instability, dynamic availability and probability of node churn identified by Broggi as application placement challenges.

(Bittencourt et al., 2017) introduces the concept of mobility in the Edge infrastructure formulating it as User mobility. It studies how user mobility affects the demand that has to be served from Edge infrastructures, to optimise user application provision bringing it closer to the user. In this context, (Bittencourt et al., 2017) still considers Edge infrastructures as stationary environments, therefore not yet taking into account the challenges that instability on Edge resource availability can bring to scheduling in Edge infrastructures.

Differently, (Daneshfar et al., 2019) is one of the few examples that at this stage contemplate uncertainty of Edge node availability in fog infrastructure management in a similar approach than Ad-hoc Edge Cloud. It constructs a centralised environment that the users contact in order to execute services in the fog infrastructure comprised of several fog nodes for which a predefined availability rate has been calculated. It formulates an Integer Program to indicate if a user has to send a service to one of the Edge nodes and the probability of failure of the service execution is used as QoS measurement. Two main aspects differentiate the

Ad-hoc Edge Cloud approach and this paper: first of all, the consideration of a central decision system; and secondly, our view on the need of formulating resource availability based on the Edge devices' current status, given all dynamic aspects which influence it.

(Mouradian et al., 2019) develops on the placement of Virtual Network Functions in Edge infrastructure nodes for scenarios in which fog nodes are mobile. It models the location of a fog node as Random Waypoint Model formulating different among others movement velocity and probabilities for nodes to be static and paused. While the problem approach in this paper is different and complementary to Ad-hoc Edge Cloud, it assesses that higher mobility probabilities drive to lower qualities of service similarly to our conclusions.

Lera(Lera et al., 2019) elaborates in a Service placement policy that takes into account the availability for complex services composed by multiple service components as addressed by Ad-hoc Edge Cloud. This study tackles the node availability system following a completely different approach to Ad-hoc Edge computing. It creates the concept of a community of nodes, as a set of mutually-interconnected devices. By deploying services to devices in communities it aims to avoid service failure generated by the failure of a specific node, however adding the need for an additional management layer at a community level. In this way, it avoids directly taking the issues raised by probability of node churn addressed in Ad-hoc Edge Clouds by relying on node redundancy to ensure service availability.

The consideration of Edge node heterogeneity in combination to user mobility is addressed in (Fizza et al., 2018) which presents a scheduling mechanism for privacy constrained real-time jobs in Edge micro data centres. Heterogeneity in this work is considered at the level of the different compute capacities available on each node, however no concrete formulation on how to define these characteristics is provided permitting the comparison to the approach based on device characteristics and processor architectures (CPU, GPU, TPU, FPGA) developed in this Ad-hoc Edge Cloud.

5.6 Conclusions

This section has elaborated on an Admission control mechanism which takes into consideration the specifics in terms of dynamic availability present in Ad-hoc Edge Clouds. As part of this work we have defined and validated a mechanism for prediction of resource availability based on past behaviour which is the core contribution of this Thesis.

This evaluation reveals that node behaviour concerning connection and disconnection is crucial for the overall performance of services provisioned by the infrastructure, as we intuitively understood.

Also, at this stage, the data we could use to develop our availability prediction model relates to the behaviour of mobile devices. This fact results from the absence of reports on usage patterns of a diversity of complex IoT devices, such as drones, robots and connected cars. We understand that the diversity of IoT devices and different usage contexts of these devices will facilitate the future definitions of more specific prediction models adjusted to the diversities of the common behaviour of additional IoT devices in concrete usage scenarios.

Furthermore, it is important to notice, that in this chapter we have focused on the analysis of the computational part of decentralised cloud management however we acknowledge that a multitude of concerns remains to be explored both at network, security, and data storage levels.

The results obtained in the development of this Thesis chapter have been published in P3 (Juan Ferrer et al., 2021).

CHAPTER 6

Conclusions

The main idea which has driven the development of this Thesis is the perception that computing is now commonly spread outside the Data Centres boundaries.

Initially dricen by the popularisation of smartphones, computing is now available everywhere in a myriad of connected devices, and accessible at the palm of our hands in our last generation phones, in our speakers, fridges, in our TV, to cite some examples. Considering the increasing compute demands we observe in multitude of situations in our daily life today, it is presumed as an unjustifiable wastage to only use compute capacity in all kinds of devices as gateway to access services offered from data centre clouds.

This initial reflection has been increasingly conceded in the course of the development of this Thesis. First by observing how Internet of Things development is expanding at a significant rapid pace, making the figures on the expected number of connected devices increment year after year. In addition, Internet of Things proliferation now complements growth of smartphones, and is facilitating to further distribute computing elements at the Edge of the network in large number of locations in environments counting on dedicated Edge compute units to serve nearby IoT technology deployments.

More recently, we can witness how increasingly IoT devices are surpassing initial capabilities focused on just sensing, to gain significant sophistication in the computational power they are today able to carry. This is making that the limits initially established in Edge computing among the IoT and Edge devices are today blurring, making it possible to pack in a single device sensing and compute capacities, thanks to the development of micro-processor technologies thanks to Moore's law.

These complex IoT Edge devices, such as robots and autonomous vehicles, can be viewed as mobile devices which provide complex aggregations of computing and storage resources together with diverse and heterogeneous sensors and actuators which all together implement a cognitive loop.

According to predictions these devices will need to become better, faster and cheaper. Producers will soon be under pressure to provide complex behaviours, cognitive capabilities and skills at competitive costs, while increasing on-board computation and storage of IoT Edge Compute devices will raise their costs, increase energy demand and reduce their autonomy.

The self-contained and self-sustaining nature of these novel IoT Edge resources combined with their size and energy harvesting constrains will require of novel computing and communication architectures beyond state of the art today.

A caveat has to be made in relation to Moore's law development. For the last few decades, overcoming similar challenges has always relied on the application of Moore's law.

This has allowed producing ever better and capable hardware. Hardware manufacturers are increasingly encountering more difficulties in producing ever miniaturised low-power computing units which are cheaper and faster. This does not probably mean that computing progress will suddenly stall, but rather it can affect the nature of that progress. The computing progress could be progressively changing to approaches which take better advantage of available resources while coping with the necessary balance among resources in high demand: computing and energy.

6.1 Thesis findings

In this Thesis we have addressed some of the challenges these expected changes will bring to the development of Edge computing technologies by developing the concept of Ad-hoc Edge Clouds and advancing an architecture model for its implementation. This architectural approach represents a significant breakthrough to initial Edge computing developments concentrated in providing low latency compute environments for which IoT devices are solely considered as data sources. Ad-hoc Edge Cloud is a distributed and decentralised Edge computing system dynamically formed out of IoT Edge computing resources, which aims to exploit increasingly available compute capacity at the Edge.

In addition, we have deeply analysed the particularities of IoT Edge devices which constitute this infrastructure. IoT Edge devices pose explicit challenges to resource and service management in this context especially due to heterogeneity, dynamicity, and volatility of resources, resulting in the probability of node churn. We have analysed these specific issues in two main contexts: At the level of Resource management, elaborating on the mechanisms for Ad-hoc Edge Cluster formation and management; in relation to Admission control and Service placement processes, presenting an Admission Control mechanism and an associated resource availability prediction model driven by the needs exposed by dynamic behaviour of participant IoT Edge devices.

The specific Thesis contributions are developed in Section 1.6. Their relation to research questions posed for this Thesis together with a summary of gathered conclusions is presented in detailed in the upcoming subsections.

6.1.1 Analysis of Research Questions

6.1.1.1 Research Question 1- RQ1

RQ1.1 Are IoT devices suitable devices to create ad-hoc Ad-hoc Edge Computing infrastructures based on decentralised computing approaches?

Yes, the assessment performed in this Thesis allows us to conclude that at current stage of technology developments, IoT devices classified as Large (which support general purpose OS) and therefore , capable of executing Docker containerisation technologies, are suitable devices to create Ad-hoc Edge Computing infrastructures. Distributed key storages are a fundamental part to build Ad-hoc Edge Computing distributed and decentralised infrastructures. Distributed key storages bring the necessary mechanisms for data distribution, synchronisation and fault tolerance which permit Ad-hoc Edge Infrastructure to handle node churn at information level and sustain distribution and decentralisation. In this context, experimentation has shown the outstanding performance of Etcd in comparison with other existing solutions such as Apache Cassandra. It is remarkable that Etcd's consensus algorithm is the build in mechanism which allows Ad-hoc Edge Architecture to achieve full decentralisation and distribution (see Chapter 3 for full details).

RQ1.2 Is completely decentralised management feasible in all contexts?

No. While the evaluation performed as part of this work (Section 4.4, Section 4.5) shows the feasibility of fully decentralised cluster management systems based on Raft consensus mechanism and Etcd, it has also evidenced the limits of full decentralisation. We have identified that scenarios in which resource volatility levels which can go over 75%, would have to include some dedicated fixed nodes in order not to compromise the quality on the services offered by the Ad-hoc Edge Computing infrastructure.

RQ1.3 What are the performance overheads that decentralisation and complete distribution bring to ad-hoc infrastructures?

Findings of evaluation also demonstrate performance overheads of full decentralisation and distribution. In Section 4.4 we have shown that in Etcd there is a direct relation between the memory employed in the hosting node and the overall size of the cluster. This is, a natural consequence of the necessary data synchronisation processes among the different distributed storage cluster nodes which sustains the distribution and decentralisation of Ad-hoc Edge Infrastructure. Nevertheless, it raises an important aspect to consider in future studies with regards to the potential size of clusters (established in a limit of 30 forming devices in the performed evaluation) and even bringing the need of developing Edge Hierarchical models as analysed in our state of the art (Section 2.5.2).

RQ1.4 What specific use cases could benefit from such ad-hoc edge infrastructures?

This dissertation has identified use cases in the areas of Social computing, Smart Home, Industrial plant, UAVs for inspection and Connected vehicles as examples of use cases (see Section 3.4 for complete description) in which Ad-hoc Edge Cloud infrastructure can show its benefits and added value.

RQ1 Related contributions:

Contribution #1 A systematic literature review of papers in the areas of Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing

Contribution #2 A Characterisation of the Ad-hoc Edge Cloud Concept, Architecture and Use cases

Contribution #3 A mechanism for Ad-hoc Edge Cluster instantiation and management

6.1.1.2 Research Question 2- RQ2

RQ2.1 How are IoT Edge resources characterised?

This Thesis has elaborated on specific resource management characteristics which substantially distinguish IoT Edge resources management from traditional resource management in Cloud computing(Section 3.3). Four different aspects characterise IoT Edge resources: (1) the scale with respect to the number of devices that take part in the infrastructure; (2) the diversity of devices that can participate in the infrastructure and their inherent heterogeneity; (3) the fact that predominantly these devices are mobile and therefore (4) they are generally affected by connectivity instabilities and resource constraints both in terms of battery and resource and storage capacity. The latter two originate the main specific aspect among typical Cloud resources and resources in Ad-hoc Edge Cloud, which is the probability of node churn.

RQ2.2 Do IoT Edge devices specific characteristics' condition or determine the applicability of the approach?

The previously identified IoT Edge devices resource characteristics, determine two inherent requirements for the Ad-hoc Edge Cloud approach:

The first requirement determines the need for reliability to node volatility and relates to the expectable high degrees of node churn in this context. This need is addressed at architectural level. Our solution relies on distributed storages for logical connectivity layer to develop a decentralised distributed management among all participant nodes in the Ad-hoc Edge Cloud infrastructure. This solution avoids having a single a point of failure for Ad-hoc Edge infrastructure and develops inbuilt scale mechanisms.

The second requirement is inherent to the constrained nature of the resources targeted in this work. It specifies the requirement of lightweight implementation in order to cope with the specific IoT Edge devices resource constrains. It refers to the need of not hinder the normal operation of IoT Edge resources which participate in a certain moment in the Ad-hoc Edge Cloud infrastructure . Thanks to technology availability our work has elaborated on top of Docker. However more and more lightweight virtualisation technologies emerge such us the cited Unikernels ("Unikernels, Rethinking Cloud Infrastructure", 2019), Kata Container("Kata Containers", 2019) and gVisor("gVisor, Container Runtime Sandbox", 2019).

RQ2.3 What are the implications of scale and heterogeneity?

In our work we have determined that expected scale and heterogeneity has implications in two main aspects.

Scale together with the inherent IoT device dynamicity prove the need for fault tolerant decentralised resource management. Decentralised resource management incorporates mechanisms in the Ad-hoc Edge infrastructure to transparently handle the dynamicity in resource availability.

Heterogeneity requirements cater for the definition of a specific resource description schema, considering both static and dynamic characteristics. This schema with help of the Node manager of our architecture allows handling transparently heterogeneous devices. This fact, together with the technology choice made for Docker as containerisation technology, has permitted us to have a uniform management of resources independently of their built-in differences.

As previously introduced, the performed experimentation has helped to understand the performance overheads in terms of memory consumption that scale brings. To put it differently, the fact of relying on a common and widespread technology such as Docker, has offered us a way to transparently handle heterogeneity. This decision to some extent limits the range of devices to which the approach is applicable (not considering tiny and small devices in HEADS classification(HEADS Project, 2016)). Nevertheless, it is an essential characteristic to develop a consistent technology platform with current Cloud computing state of the art and practice.

RQ2.4 What are the particularities of IoT devices which have to be considered on Admission control and Resource management?

Node churn is unquestionably the device characteristic fundamental to the development of Ad-hoc Edge Cloud concept and architecture. At the level of Admission control, the stability in resource availability and the inherently related node battery levels (as indicator of device in risk to abandon abruptly the infrastructure) are the key influential factors for Ad-hoc Edge Cloud placement decisions. For Resource management Node churn shapes the mechanisms that manage IoT Edge devices participation in Ad-hoc Edge Cloud infrastructures. Moreover, it underlines the need of relying fault tolerant distributed key storages and the key role of its distributed consensus algorithms.

RQ2 Related contributions:

Contribution #1 A systematic literature review of works in the areas of Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing

Contribution #2 A Characterisation of the Ad-hoc Edge Cloud Concept, Architecture and Use cases

Contribution #3 A mechanism for Ad-hoc Edge Cluster instantiation and management

6.1.1.3 Research Question 3- RQ3

RQ3.1 What are the specific Resource management issues derived from the use of IoT Edge devices?

As mentioned earlier, the specific characteristics of the IoT Edge devices which partake in Ad-hoc Edge Cloud infrastructures poses particular challenges to resource management due to the dynamicity on the availability of resources and the expected massive scale of participant devices. The need in Ad-hoc Edge Cloud to cope with high degrees of node churn brings the challenge of developing a decentralised resource management system able to handle scale and reliable to node volatility.

RQ3.2 What is the impact of dynamicity and node churn for Resource management in this context?

The impact of dynamicity and node churn has been measured in our experimentation in terms of recovery time. This represents the necessary time for the Nodes Distributed registry to achieve a healthy state when a number of previously available nodes disappear. Our findings demonstrate(Section 4.4) the direct link among the number of nodes abandoning the system and the necessary recovery times due to the data synchronisation processes. Results obtained show that recovery times are correlated linearly with the number of nodes that withdraw from the Ad-hoc Edge infrastructure.

RQ3.3 What is the degree of node churn that makes the infrastructure unmanageable?

Our experimentation(Section 4.4) has also evidenced that built-in fault tolerance features in Etcd are suitable to cope with volatility degrees that do not overpass the 75% of existing nodes abruptly leaving the Ad-hoc Edge infrastructure.

RQ3 Related contributions:

Contribution #1 A systematic literature review of works in the areas of Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing

Contribution #3 A mechanism for Ad-hoc Edge Cluster instantiation and management

6.1.1.4 Research Question 4- RQ4

RQ4.1 Which could be the mechanisms to decide on service acceptance or rejection based on current Ad-hoc Edge infrastructure status?

This Thesis has proposed an Admission control mechanism which chooses from all available resources to the Ad-hoc Edge infrastructure in a certain moment, those identified as most suitable to serve the service requirements. This process is a two steps process. It initially filters from all available resources those that have sufficient capacity in terms of dynamic characteristics (CPU, memory and storage) to execute the service or part of it. After, a qualitative step prioritises available nodes by the assessment of its Node Quality. Node Quality measures stability of the node as part of the infrastructure based on its historical behaviour with regards to connections and disconnections. Node Quality is determined by the Node availability prediction algorithm adapted in this work from previous work in the context of Volunteer Computing (Panadero et al., 2018). The performed adaptations refer to changes to reflect IoT Edge devices time frame dynamicity and specific parameters of Ad-hoc Edge Cloud (see all details in Section 5.3)..

RQ4.2 What are the node parameters that determine the Quality of a Node?

This study assess the quality of a node making use of historical values with regards to the resource availability to the infrastructure by observing two aspects: percentage of connected time and number of disconnections since the initial registration of the node (details available in Section 5.3.1). These parameters aims at measuring the stability of the resource as part of the infrastructure and constitute essence to develop the concept of Quality of a Node defined as probability of an IoT Edge Device to remain connected to the Ad-hoc Edge Infrastructure for a certain period.

RQ4.3 What are the parameters which indicate their capacity?

This thesis dissertation offers a resource characterisation which observes capacity of resources in terms of computing Capacity, Memory and disk, processor architecture , upload and download speed as well as battery level (Table 5.1).

RQ4.4 Is the stability of a node as part of the Ad-hoc Edge an effective measurement to predict resource/node behaviour?

Yes. Experimentation performed in this work has shown that Node Quality is solely determined by the specific node characteristics and its historical behaviour concerning connection and disconnection. This is the node stability as part of the infrastructure, and hence it is not affected by scale (Section 5.4, Section 5.6).

RQ4.5 Is past behaviour a good indication of probability of node churn?

Yes, our evaluation reveals that obtained Node Qualities in the different experimentation scenarios defined in this work are coherent with the defined node behaviour. It is also understood from experimentation the node behaviour concerning connection and disconnection is crucial to assess the overall performance of services provisioned by the infrastructure (see (Section 5.4, Section 5.6)). Also, it is important to note, that at current stage our resource availability prediction model takes in account particularities of IoT Edge resources in terms of dynamic resource behaviour. However, it has been developed solely taking into account usage patterns of mobile devices, in absence of data of existing IoT devices deployments which allows us to understand temporal usage patterns for other diverse IoT Edge devices such as drones, robots and connected cars.

RQ3 Related contributions:

Contribution #1 A systematic literature review of works in the areas of Mobile Cloud Computing, Mobile Ad-hoc Computing and Edge computing

Contribution #4 A mechanism for Admission control and service placement which includes a validated IoT Edge Resource

6.2 Future Work

In previous section we have elaborated in the conclusions and findings of this Thesis. Also, in the bullets below we have sketched areas for future work in relation to this PhD Thesis.

Adaptation to more lightweight virtualisation technologies This thesis has validated its approach making use of state-of-the art containerisation technologies based on Docker. This technological approach offers the advantage of having a common technology context with existing Cloud computing developments, but at the same time frames its applicability devices equipped with general purpose OS capable of supporting Docker. At the same time, several more lightweight virtualisation technologies have recently come out, such as

Unikernels and micro-containers. In this context, advanced lightweight compute containers can be the enabler technology that will allow the execution of self-contained, purpose-specific services that specifically target only certain functions needed at the edge in more resource constrained devices. The ability to use micro-containers will permit much thicker, monolithic services previously running resource richer devices will be transformed into collections of lighter, purpose-specific micro services which can be deployed and executed at edge as needed and on-demand.

Hierarchical cluster management In this study we have identified certain limits in the ability to scale of the proposed approach due to increase in memory consumption of Etcd as long as the cluster grows. This raises the need to further refine the approach to handle scalability in the number of constituent devices as defined as part of this work. A potential approach already analysed in the state of the art analysis, is the definition of hierarchical clusters at the Edge. A potential approach to tackle this issue could develop mechanisms for Edge cluster federation, even considering federation with resource richer environments in traditional clouds.

Location handling The previous 'future work area' can be linked directly to enhanced location area. Our work addresses the concept of location as the conceptual mechanism which makes a set of IoT Edge resources available to each other. A more elaborated consideration of location could take into account the determination of the physical location of nodes and automated discovery, as well as, mechanisms to determine execution zones and associate clusters of diverse granularities to it.

Adaptation of node prediction to diverse usage patterns Another area in which this Thesis could clearly identify potential for future work is in the defined mechanism for resource availability prediction. Due to data availability the developed model could only consider usage patterns of mobile devices. It is anticipated that current development of IoT deployments, will soon make available more richer data sets from which to infer usage patterns of several kinds of IoT Edge devices. Even more, the development of such algorithms adapted to the specificities of concrete use cases will help for further refinement.

Cognitive infrastructure management The ability to predict resource availability is an area of application of cognitive techniques (machine learning and AI) to Edge resource management that we have developed in this Thesis. Beyond this simple application, there is a complete area of development for Ad-hoc Edge related to the development of self-management techniques applied to Edge infrastructure management. This self-management approach has to take advantage of self-learning techniques for instance developing adaptive selection, conflict resolution, adaptations for QoS and techniques to consider further the volatility and uncertainty introduced due to real-world dynamics to enable efficient and reliable service provisioning.

Furthermore, in Section 2.4.1 and Section 2.5.1 we already listed a set of research challenges that are in our opinion a basis to guide next steps after the completion of this work. In Table 6.1 we summarise them and indicate the degree in which this Thesis has elaborated in the corresponding area.

Challenge	Research Area	Development in Ad-hoc Edge Cloud?
QoS and Fault tolerance	Mobile Ad-hoc Cloud Computing	Experimentally addressed
Scalability	Mobile Ad-hoc Cloud Computing	Experimentally addressed
Incentives for participation	Mobile Ad-hoc Cloud Computing	Not addressed
Resource Heterogeneity	Mobile Ad-hoc Cloud Computing	Experimentally addressed
Resource Discovery	Mobile Ad-hoc Cloud Computing	Experimentally addressed
Edge management: Resource Management	Edge Computing	Main focus
Edge Management: Fault tolerance and distributed service management	Edge Computing	Main focus
Edge management: Edge Workload management	Edge Computing	Not addressed
Edge management: Edge Workload Scheduling	Edge Computing	Main focus
Edge management:Data management	Edge Computing	Not addressed
Edge Interoperability: Across Edge execution orchestration	Edge Computing	Not addressed
Edge Interoperability: Cloud and Edge Interoperability	Edge Computing	Not addressed
Economy	Edge Computing	Not addressed
Eco-efficiency	Edge Computing	Not addressed
Security and privacy	Edge Computing	Not addressed
Connectivity and Resilience	Edge Computing	Not addressed

Table 6.1: Research Challenges analysis for Next steps

Bibliography

Abolfazli, S., Sanaei, Z., Ahmed, E., Gani, A., & Buyya, R. (2014). Cloud-based augmentation for mobile devices: Motivation, taxonomies, and open challenges. *IEEE Communications Surveys and Tutorials*, *16*(1), 337–368. https://doi.org/10.1109/SURV.2013.070813.00285

Alizadeh, M., Abolfazli, S., Zamani, M., Baaaharun, S., & Sakurai, K. (2016). Authentication in mobile cloud computing: A survey. *Journal of Network and Computer Applications*, *61*, 59–80. https://doi.org/10.1016/j.jnca.2015.10.005

Amazon Web Services. (2019). Retrieved October 20, 2019, from https://aws.amazon.com/

Amazon Web Services CloudFormation. (2019). Retrieved, from https://aws.amazon.com/cloudformation/

Amazon Web Services EC2. (2019). Retrieved October 20, 2019, from https://aws.amazon.com/ec2/

Amazon Web Services EC2 A1 Instances. (2019). Retrieved December 29, 2019, from https://aws.amazon.com/ec2/instance-types/a1/

Amazon Web Services Greengrass. (2017). Retrieved August 1, 2018, from https://aws.amazon.com/greengrass/

Amazon Web Services Greengrass FAQs. (2017). Retrieved August 1, 2018, from https://aws.amazon.com/greengrass/faqs/

Amodei, D., & Hernandez, D. (2018). *AI and Compute*. Retrieved December 1, 2019, from https://openai.com/blog/ai-and-compute/

Andreozzi, S. (2009). *GLUE Specification v. 2.0*. Retrieved June 27, 2020, from https://www.ogf.org/documents/GFD.147.pdf

Apache Cassandra. (2019). Retrieved December 28, 2019, from http://cassandra.apache.org/

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G., Patterson, D. A., Rabkin, A., Stoica, I., & Zaharia, M. (2009). *Above the clouds: A berkeley view of cloud computing* (tech. rep. UCB/EECS-2009-28). EECS Department, University of California, Berkeley. http://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

Azure IoT Edge. (2017). Retrieved August 1, 2018, from https://azure.microsoft.com/en-us/campaigns/iot-edge/

Azure IoT GitHub. (2017). Retrieved August 1, 2018, from https://github.com/Azure/iot-edge/

Azure IoT Reference Architecture (tech. rep.). (2016). Retrieved August 1, 2018, from http://download.microsoft.com/download/A/4/D/A4DAD253-BC21-41D3-B9D9-87D2AE6F0719/Microsoft_Azure_IoT_Reference_Architecture.pdf

Bagchi, S. (2014). Admission control and scheduling of remote processes in loosely-coupled distributed systems. *Computers and Electrical Engineering*, *40*(5), 1666–1682. https://doi.org/10.1016/j.compeleceng.2013.08.013

Balasubramanian, V., & Karmouch, A. (2017). An infrastructure as a Service for Mobile Ad-hoc Cloud. *2017 IEEE 7th Annual Computing and Communication Workshop and Conference, CCWC 2017*, 1–7. https://doi.org/10.1109/CCWC.2017.7868393

Bandara, H. M. N. D., & Jayasumana, A. P. (2013). Collaborative applications over peer-to-peer systems-challenges and solutions. *Peer-to-Peer Networking and Applications*, *6*(3), 257–276. https://doi.org/10.1007/s12083-012-0157-3

Barr, J. (2018a). *AWS IoT, Greengrass, and Machine Learning for Connected Vehicles at CES*. Retrieved August 1, 2018, from https://aws.amazon.com/blogs/aws/aws-iot-greengrass-and-machine-learning-for-connected-vehicles-at-ces/

Barr, J. (2018b). *New – EC2 Instances (A1) Powered by Arm-Based AWS Graviton Processors*. Retrieved December 29, 2018, from https://aws.amazon.com/blogs/aws/new-ec2-instances-a1-powered-by-arm-based-aws-graviton-processors/

Bhih, A. A., Johnson, P., & Randles, M. (2016). Diversity in smartphone usage. *Proceedings of the 17th International Conference on Computer Systems and Technologies 2016*, 81–88. https://doi.org/10.1145/2983468.2983496

Bilal, K., Khalid, O., Erbad, A., & Khan, S. U. (2018). Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers. *Computer Networks*, *130*, 94–120. https://doi.org/10.1016/j.comnet.2017.10.002

Bitam, S., Zeadally, S., & Mellouk, A. (2018). Fog computing job scheduling optimization based on bees swarm. *Enterprise Information Systems*, *12*(4), 373–397. https://doi.org/10.1080/17517575.2017.1304579

Bittencourt, L. F., Diaz-Montes, J., Buyya, R., Rana, O. F., & Parashar, M. (2017). Mobility-Aware Application Scheduling in Fog Computing. *IEEE Cloud Computing*, *4*(2), 26–35. https://doi.org/10.1109/MCC.2017.27

Bojkovic, Z. S., Bakmaz, B. M., & Bakmaz, M. R. (2017). Vision and enabling technologies of tactile internet realization. *2017 13th International Conference on Advanced Technologies, Systems and Services in Telecommunications (TELSIKS)*, 113–118. https://doi.org/10.1109/TELSKS.2017.8246242

Bonomi, F., Milito, R., Natarajan, P., & Zhu, J. (2014). Fog computing: A platform for internet of things and analytics. In N. Bessis & C. Dobre (Eds.), *Big data and internet of things: A roadmap for smart environments* (pp. 169–186). Springer International Publishing. https://doi.org/10.1007/978-3-319-05029-4_7

Bonomi, F., Milito, R., Zhu, J., & Addepalli, S. (2012). Fog computing and its role in the internet of things. *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing*, 13–16. https://doi.org/10.1145/2342509.2342513

Brogi, A., Forti, S., Guerrero, C., & Lera, I. (2020). How to place your apps in the fog: State of the art and open challenges. *Software: Practice and Experience*, *50*(5), 719–740. https://doi.org/10.1002/spe.2766

Chandra, A., Weissman, J., & Heintz, B. (2013). Decentralized Edge Clouds. *IEEE Internet Computing*, *17*(5), 70–73. https://doi.org/10.1109/MIC.2013.93

Chang, D., Xu, G., Hu, L., & Yang, K. (2013). A network-aware virtual machine placement algorithm in mobile cloud computing environment. *2013 IEEE Wireless Communications and Networking Conference Workshops, WCNCW 2013*, 117–121. https://doi.org/10.1109/WCNCW.2013.6533325

Chang, J., Balan, R. K., & Satyanarayanan, S. (2005). *Exploiting rich mobile environments* (tech. rep. December). http://reports-archive.adm.cs.cmu.edu/anon/anon/usr0/ftp/home/ftp/2005/CMU-CS-05-199.pdf

Chang, R. S., Gao, J., Gruhn, V., He, J., Roussos, G., & Tsai, W. T. (2013). Mobile cloud computing research - Issues, challenges, and needs. *Proceedings - 2013 IEEE 7th International Symposium on Service-Oriented System Engineering, SOSE 2013*, 442–453. https://doi.org/10.1109/SOSE.2013.96

Chen, D., Cong, J., Gurumani, S., Hwu, W.-m., Rupnow, K., & Zhang, Z. (2016). Platform choices and design demands for IoT platforms: cost, power, and performance tradeoffs. *IET Cyber-Physical Systems: Theory & Applications*, *1*(1), 70–77. https://doi.org/10.1049/iet-cps.2016.0020

Chen, J., & Ran, X. (2019). Deep Learning With Edge Computing: A Review. *Proceedings of the IEEE*, *107*(8), 1655–1674. https://doi.org/10.1109/jproc.2019.2921977

Chun, B.-G., Ihm, S., Maniatis, P., Naik, M., & Patti, A. (2011). CloneCloud: Elastic Execution Between Mobile Device and Cloud. *Proceedings of the Sixth Conference on Computer Systems*, 301–314. https://doi.org/10.1145/1966445.1966473

Chun, B.-G., & Maniatis, P. (2009). Augmented Smartphone Applications Through Clone Cloud Execution. *Proceedings of the 12th Conference on Hot Topics in Operating Systems*, 8. http://dl.acm.org/citation.cfm?id=1855568.1855576

Cisco Systems. (2016). *Fog Computing and the Internet of Things: Extend the Cloud to Where the Things Are* (tech. rep.). https://www.cisco.com/c/dam/en_us/solutions/trends/iot/docs/computing-overview.pdf

Conti, M., & Kumar, M. (2010). Opportunities in opportunistic computing. *Computer*, *43*(1), 42–50. https://doi.org/10.1109/MC.2010.19

Cuervo, E., Balasubramanian, A., Cho, D.-k., Wolman, A., Saroiu, S., Chandra, R., & Bahl, P. (2010). MAUI: Making Smartphones Last Longer with Code Offload. *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, 49–62. https://doi.org/10.1145/1814433.1814441

Daneshfar, N., Pappas, N., Polishchuk, V., & Angelakis, V. (2019). Service Allocation in a Mobile Fog Infrastructure under Availability and QoS Constraints. *2018 IEEE Global Communications Conference, GLOBECOM 2018 - Proceedings*, 1–6. https://doi.org/10.1109/GLOCOM.2018.8647488

Deng, R., Lu, R., Lai, C., Luan, T. H., & Liang, H. (2016). Optimal Workload Allocation in Fog-Cloud Computing Toward Balanced Delay and Power Consumption. *IEEE Internet of Things Journal*, *3*(6), 1171–1181. https://doi.org/10.1109/JIOT.2016.2565516

DITAS, Data-intensive applications improvement by moving data in mixed cloud/fog environments. (2017). Retrieved August 1, 2018, from http://www.ditas-project.eu/

DMTF. (2016). *DSP0263, Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol*. Retrieved June 27, 2020, from https://www.dmtf.org/sites/default/files/standards/documents/DSP0263_2.0.0.pdf

Docker comes to Raspberry Pi. (2016). Retrieved August 30, 2016, from https://www.raspberrypi.org/blog/docker-comes-to-raspberry-pi/

Docker: Enterprise Container Platform for High-Velocity Innovation. (2019). Retrieved August 15, 2019, from https://www.docker.com

El-Sayed, H., Sankar, S., Prasad, M., Puthal, D., Gupta, A., Mohanty, M., & Lin, C.-T. (2018). Edge of Things: The Big Picture on the Integration of Edge, IoT and the Cloud in a Distributed Computing Environment. *IEEE Access*, *6*, 1706–1717. https://doi.org/10.1109/ACCESS.2017.2780087

Enzai, N. I. M., & Tang, M. (2014). A Taxonomy of Computation Offloading in Mobile Cloud Computing. *2014 2nd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, 19–28. https://doi.org/10.1109/MobileCloud.2014.16

Etcd Discovery service protocol. (2019). Retrieved October 20, 2019, from https://etcd.io/docs/v3.3.12/dev-internal/discovery%7B%5C_%7Dprotocol/

Etcd Learner design. (2019). Retrieved December 29, 2019, from https://github.com/etcd-io/etcd/blob/master/Documentation/learning/design-learner.md

Etcd Runtime reconfiguration. (2019). Retrieved December 29, 2019, from https://github.com/etcd-io/etcd/blob/master/Documentation/op-guide/runtime-configuration.md#cluster-reconfiguration-operations

Etcd, A distributed, reliable key-value store for the most critical data of a distributed system. (2019). Retrieved October 20, 2019, from https://etcd.io

Fernando, N., Loke, S. W., & Rahayu, W. (2011). Dynamic Mobile Cloud Computing: Ad Hoc and Opportunistic Job Sharing. *Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on*, 281–286. https://doi.org/10.1109/UCC.2011.45

Fernando, N., Loke, S. W., & Rahayu, W. (2013). Mobile cloud computing: A survey. *Future Generation Computer Systems*, *29*(1), 84–106. https://doi.org/http://dx.doi.org/10.1016/j.future.2012.05.023

Fernando, N., Loke, S. W., & Rahayu, W. (2012). Mobile crowd computing with work stealing. *2012 15th International Conference on Network-Based Information Systems*, 660–665.

Fizza, K., Auluck, N., Rana, O., & Bittencourt, L. (2018). PASHE: Privacy Aware Scheduling in a Heterogeneous Fog Environment. *Proceedings - 2018 IEEE 6th International Conference on Future Internet of Things and Cloud, FiCloud 2018*, 333–340. https://doi.org/10.1109/FiCloud.2018.00055

Gao, J., Gruhn, V., He, J., Roussos, G., & Tsai, W.-T. (2013). Mobile Cloud Computing Research - Issues, Challenges and Needs. *2013 IEEE Seventh International Symposium on Service-Oriented System Engineering*, 442–453. https://doi.org/10.1109/SOSE.2013.96

Garcia Lopez, P., Montresor, A., Epema, D., Datta, A., Higashino, T., Iamnitchi, A., Barcellos, M., Felber, P., & Riviere, E. (2015). Edge-centric Computing.

ACM SIGCOMM Computer Communication Review, *45*(5), 37–42. https://doi.org/10.1145/2831347.2831354

Gartner. (2017). *The Edge Manifesto: Digital Business, Rich Media, Latency Sensitivity and the Use of Distributed Data Centers*. Retrieved August 1, 2018, from https://www.gartner.com/doc/3104121?ref=SiteSearch%7B%5C&%7Dsthkw=micro%20data%20centres%7B%5C&%7Dfnl=search%7B%5C&%7DsrcId=1-3478922254

Ghosh, Rahul and Trivedi, Kishor S. and Naik, Vijay K. and Kim, Dong Seong. (2010). End-to-End Performability Analysis for Infrastructure-as-a-Service Cloud: An Interacting Stochastic Models Approach. *2010 IEEE 16th Pacific Rim International Symposium on Dependable Computing*, 125–132. https://doi.org/10.1109/PRDC.2010.30

Gkatzikis, L., & Koutsopoulos, I. (2013). Migrate or not? exploiting dynamic task migration in mobile cloud computing systems. *Wireless Communications, IEEE*, *20*(3), 24–32. https://doi.org/10.1109/MWC.2013.6549280

Godfrey, P. B., Shenker, S., & Stoica, I. (2006). Minimizing churn in distributed systems. *SIGCOMM Comput. Commun. Rev.*, *36*(4), 147–158. https://doi.org/10.1145/1151659.1159931

Google Cloud Edge TPU - Run Inference at the Edge. (2019). Retrieved December 1, 2019, from https://cloud.google.com/edge-tpu/

Google Coral (beta). (2019). Retrieved August 15, 2019, from https://coral.withgoogle.com/

Google Trends, fog computing vs Edge Computing. (2018). Retrieved August 1, 2018, from https://trends.google.com/trends/explore?q=fog%20computing,edge%20computing

Guan, L., Ke, X., Song, M., & Song, J. (2011). A survey of research on mobile cloud computing. *Proceedings - 2011 10th IEEE/ACIS International Conference on Computer and Information Science, ICIS 2011*, 387–392. https://doi.org/10.1109/ICIS.2011.67

Guess, A. (2015). *Ray Kurzweil Predicts Our Phones Will Be As Smart As Us By 2020*. Retrieved December 1, 2019, from https://www.dataversity.net/ray-kurzweil-predicts-our-phones-will-be-as-smart-as-us-by-2020/

gVisor, Container Runtime Sandbox. (2019). Retrieved October 20, 2019, from https://github.com/google/gvisor

Ha, K., Chen, Z., Hu, W., Richter, W., Pillai, P., & Satyanarayanan, M. (2014). Towards wearable cognitive assistance. *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services*, 68–81. https://doi.org/10.1145/2594368.2594383

Hammam, A., & Senbel, S. (2014). A reputation trust management system for ad-hoc mobile clouds. *Intelligent Systems Reference Library*, *70*, 519–539. https://doi.org/10.1007/978-3-662-43616-5_20

Hasan, R., Hossain, M. M., & Khan, R. (2015). Aura: An IoT Based Cloud Infrastructure for Localized Mobile Computation Outsourcing. *2015 3rd IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, 183–188. https://doi.org/10.1109/MobileCloud.2015.37

HEADS Project. (2016). *D3.3. Final Framework of resource-constrained devices and networks* (tech. rep.). http://heads-project.eu/sites/default/files/HEADS%20D3.3%20V1.0.pdf

Hoang, D. T., Lee, C., Niyato, D., & Wang, P. (2013). A survey of mobile cloud computing: Architecture, applications, and approaches. *Wireless*

Communications and Mobile Computing, *13*(18), 1587–1611. https://doi.org/10.1002/wcm.1203

Hong, K., Lillethun, D., Ramachandran, U., Ottenwälder, B., & Koldehofe, B. (2013). Mobile fog: A programming model for large-scale applications on the internet of things. *Proceedings of the Second ACM SIGCOMM Workshop on Mobile Cloud Computing*, 15–20. https://doi.org/10.1145/2491266.2491270

Huerta Cánepa, G. F. (2012). *A context-aware application offloading scheme for a mobile peer-to-peer environment* (Doctoral dissertation). Ph. D. dissertation, KAIST, South Korea.

Huerta-Canepa, G., & Lee, D. (2010). A virtual cloud computing provider for mobile devices. *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services: Social Networks and Beyond*. https://doi.org/10.1145/1810931.1810937

IDC FutureScape: Worldwide IT Industry 2019 Predictions. (2018). Retrieved October 20, 2019, from https://www.idc.com/getdoc.jsp?containerId=US44403818

Intel Movidious. (2019). Retrieved August 15, 2019, from https://www.movidius.com/

Jeong, S., Simeone, O., & Kang, J. (2018). Mobile Edge Computing via a UAV-Mounted Cloudlet: Optimization of Bit Allocation and Path Planning. *IEEE Transactions on Vehicular Technology*, *67*(3), 2049–2063. https://doi.org/10.1109/TVT.2017.2706308

Johnston, S., & Cox, S. (2017). The Raspberry Pi: A Technology Disrupter, and the Enabler of Dreams. *Electronics*, *6*(3), 51. https://doi.org/10.3390/electronics6030051

Jonathan, A., Ryden, M., Oh, K., Chandra, A., & Weissman, J. (2017). Nebula: Distributed Edge Cloud for Data Intensive Computing. *IEEE Transactions on Parallel and Distributed Systems*, *9219*(100), 1–1. https://doi.org/10.1109/TPDS.2017.2717883

Juan Ferrer, A. (2018). *adhoc-edge-cloud*. Retrieved June 28, 2020, from https://github.com/anajuan/adhoc-edge-cloud

Juan Ferrer, A., Hernandez, F., Tordsson, J., Elmroth, E., Ali-Eldin, A., Zsigri, C., Sirvent, R., Guitart, J., Badia, R. M., Djemame, K., et al. (2012). OPTIMIS: A holistic approach to cloud service provisioning. *Future Generation Computer Systems*, *28*(1), 66–77.

Juan Ferrer, A., Marqués, J. M., & Jorba, J. (2019). Ad-hoc Edge Cloud : A framework for dynamic creation of Edge computing infrastructures. *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, 1–7. https://doi.org/10.1109/ICCCN.2019.8847142

Juan Ferrer, A., Marquès, J. M., & Jorba, J. (2019). Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing. *ACM Computing Surveys*, *51*(6). https://doi.org/10.1145/3243929

Juan Ferrer, A., Panadero, J., Marques, J.-M., & Jorba, J. (2021). Admission control for ad-hoc edge cloud. *Future Generation Computer Systems*, *114*, 548–562. https://doi.org/https://doi.org/10.1016/j.future.2020.08.024

Juan Ferrer, A., Woitsch, R., Kritikos, K., Kousiouris, G., Aisopos, F., Garcia, D., Plebani, P., & Masip, X. (2017). *Future Cloud Research Roadmap, FP9 Cluster inputs* (tech. rep.). Future Cloud Cluster, Clusters of

European Projects on Cloud. https://drive.google.com/file/d/0B4hHTKjZDMXGSGxoYnh4eXhURzA/view

Kata Containers. (2019). Retrieved October 20, 2019, from https://katacontainers.io/

Kemp, R., Palmer, N., Kielmann, T., & Bal, H. (2012). Cuckoo: A Computation Offloading Framework for Smartphones. In M. Gris & G. Yang (Eds.), *Mobile computing, applications, and services* (pp. 59–79). Springer Berlin Heidelberg.

Khan, A. N., Kiah, M. L. M., Khan, S. U., & Madani, S. A. (2013). Towards secure mobile cloud computing: A survey. *Future Generation Computer Systems*, *29*(5), 1278–1299. https://doi.org/http://dx.doi.org/10.1016/j.future.2012.08.003

Kirby, G., Dearle, A., Macdonald, A., & Fernandes, A. (2010). An Approach to Ad hoc Cloud Computing. *Arxiv preprint arXiv*, 2–5. https://doi.org/arXiv:1002.4738v1

Kondo, D., Andrzejak, A., & Anderson, D. P. (2008). On correlated availability in internet-distributed systems. *Proceedings of the 2008 9th IEEE/ACM International Conference on Grid Computing*, 276–283. https://doi.org/10.1109/GRID.2008.4662809

Konstanteli, K., Cucinotta, T., Psychas, K., & Varvarigou, T. (2012). Admission control for elastic cloud services. *Proceedings of the 2012 IEEE Fifth International Conference on Cloud Computing*, 41–48. https://doi.org/10.1109/CLOUD.2012.63

Kosta, S., Aucinas, A., Hui, P., Mortier, R., & Zhang, X. (2011). Unleashing the Power of Mobile Cloud Computing using ThinkAir. *CoRR*, *abs/1105.3*.

Kosta, S., Aucinas, A., & Mortier, R. (2012). ThinkAir: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading. *2012 Proceedings IEEE INFOCOM*, 945–953. https://doi.org/10.1109/INFCOM.2012.6195845

Kovachev, D., Cao, Y., & Klamma, R. (2011). Mobile Cloud Computing: A Comparison of Application Models. *CoRR*, *abs/1107.4*. http://arxiv.org/abs/1107.4940

Kubernetes Pod Overview. (2019). Retrieved, from https://kubernetes.io/docs/concepts/workloads/pods/pod-overview/

Kubik, S. (2017). *A plain language post about fog computing (that anyone can understand)*. Retrieved May 13, 2018, from https://www.openfogconsortium.org/a-plain-language-post-about-fog-computing-that-anyone-can-understand/

Kumar, K., Liu, J., Lu, Y.-H., & Bhargava, B. (2012). A Survey of Computation Offloading for Mobile Systems. *Mobile Networks and Applications*, *18*(1), 129–140. https://doi.org/10.1007/s11036-012-0368-0

Kushida, K. E., Murray, J., & Zysman, J. (2015). Cloud Computing: From Scarcity to Abundance. *Journal of Industry, Competition and Trade*, *15*(1), 5–19. https://doi.org/10.1007/s10842-014-0188-y

La, H. J., & Kim, S. D. (2014). A Taxonomy of Offloading in Mobile Cloud Computing. *2014 IEEE 7th International Conference on Service-Oriented Computing and Applications*, 147–153. https://doi.org/10.1109/SOCA.2014.22

Laksham Avinash, & Prashant Malik. (2010). Cassandra: A decentralized structured storage system. *ACM SIGOPS Operating Systems Review*, *44*(2), 35–40. https://doi.org/10.1145/1773912.1773922

Lázaro, D., Kondo, D., & Marquès, J. M. (2012). Long-term availability prediction for groups of volunteer resources. *Journal of Parallel and Distributed Computing*, *72*(2), 281–296. https://doi.org/10.1016/j.jpdc.2011.10.007

Lee, E. A., Rabaey, J., Hartmann, B., Kubiatowicz, J., Pister, K., Sangiovanni-Vincentelli, A., Seshia, S. A., Wawrzynek, J., Wessel, D., Rosing, T. S., Blaauw, D., Dutta, P., Fu, K., Guestrin, C., Taskar, B., Jafari, R., Jones, D., Kumar, V., Mangharam, R., . . . Rowe, A. (2014). The Swarm at the Edge of the Cloud. *IEEE Design & Test*, *31*(3), 8–20. https://doi.org/10.1109/MDAT.2014.2314600

Lera, I., Guerrero, C., & Juiz, C. (2019). Availability-aware service placement policy in fog computing based on graph partitions. *IEEE Internet of Things Journal*, *6*(2), 3641–3651. https://doi.org/10.1109/JIOT.2018.2889511

Linux Containers. (2020). Retrieved May 22, 2020, from https://linuxcontainers.org/

Loke, S. W., Napier, K., Alali, A., Fernando, N., & Rahayu, W. (2015). Mobile Computations with Surrounding Devices. *ACM Transactions on Embedded Computing Systems*, *14*(2), 1–25. https://doi.org/10.1145/2656214

Madhavapeddy, A., Mortier, R., Rotsos, C., Scott, D., Singh, B., Gazagnaire, T., Smith, S., Hand, S., & Crowcroft, J. (2013). Unikernels: Library operating systems for the cloud. *Proceedings of the Eighteenth International Conference on Architectural Support for Programming Languages and Operating Systems*, 461–472. https://doi.org/10.1145/2451116.2451167

Marinelli, E. E. (2009). *Hyrax : Cloud Computing on Mobile Devices using MapReduce* (Doctoral dissertation). http://www.contrib.andrew.cmu.edu/%7B~%7Demarinel/masters%7B%5C_%7Dthesis/emarinel%7B%5C_%7Dms%7B%5C_%7Dthesis.pdf

Markoff, J. (2016). *The New York Times, Moore's Law Running Out of Room, Tech Looks for a Successor*. Retrieved February 23, 2020, from https://www.nytimes.com/2016/05/05/technology/moores-law-running-out-of-room-tech-looks-for-a-successor.html

Mell, P. M., & Grance, T. (2011). *SP 800-145. The NIST Definition of Cloud Computing* (tech. rep.). Gaithersburg, MD, USA, National Institute of Standards & Technology.

Miluzzo, E., Cáceres, R., & Chen, Y.-F. (2012). Vision: Mclouds - computing on clouds of mobile devices. *Proceedings of the Third ACM Workshop on Mobile Cloud Computing and Services*, 9–14. https://doi.org/10.1145/2307849.2307854

Mishra, A., & Masson, G. (2013). MoCCA: A Mobile Cellular Cloud Architecture. *Journal of Cyber Security and Mobility*, *2*(2), 105–125. https://doi.org/10.13052/jcsm2245-1439.221

Mobile-edge computing. (2016). Retrieved August 1, 2018, from http://www.etsi.org/images/files/ETSITechnologyLeaflets/MobileEdgeComputing.pdf

Mollah, M. B., Azad, M. A. K., & Vasilakos, A. (2017). Security and privacy challenges in mobile cloud computing: Survey and way ahead. *Journal of Network and Computer Applications*, *84*, 38–54. https://doi.org/https://doi.org/10.1016/j.jnca.2017.02.001

Morshed, A., Jayaraman, P. P., Sellis, T., Georgakopoulos, D., Villari, M., & Ranjan, R. (2018). Deep Osmosis: Holistic Distributed Deep Learning in Osmotic

Computing. *IEEE Cloud Computing*, *4*(6), 22–32. https://doi.org/10.1109/MCC.2018.1081070

Mouradian, C., Kianpisheh, S., Abu-Lebdeh, M., Ebrahimnezhad, F., Jahromi, N. T., & Glitho, R. H. (2019). Application Component Placement in NFV-Based Hybrid Cloud/Fog Systems with Mobile Fog Nodes. *IEEE Journal on Selected Areas in Communications*, *37*(5), 1130–1143. https://doi.org/10.1109/JSAC.2019.2906790

Mousavi, N., Aksanli, B., Akyurek, A. S., & Rosing, T. Š. (2017). Accuracy-resource tradeoff for edge devices in Internet of Things. *2017 IEEE International Conference on Pervasive Computing and Communications Workshops, PerCom Workshops 2017*, 581–586. https://doi.org/10.1109/PERCOMW.2017.7917627

Mukherjee, A., Paul, H. S., Dey, S., & Banerjee, A. (2014). ANGELS for distributed analytics in IoT. *2014 IEEE World Forum on Internet of Things (WF-IoT)*, 565–570. https://doi.org/10.1109/WF-IoT.2014.6803230

Nodes, Kubernetes by Example. (2020). Retrieved June 27, 2020, from https://kubernetesbyexample.com/nodes/

Nordrum, A. (2016). *The Internet Of Fewer Things - IEEE Spectrum*. Retrieved, from https://spectrum.ieee.org/telecom/internet/the-internet-of-fewer-things

Nov, O., Anderson, D., & Arazy, O. (2010). Volunteer computing: A model of the factors determining contribution to community-based scientific research. *Proceedings of the 19th International Conference on World Wide Web*, 741–750. https://doi.org/10.1145/1772690.1772766

NVIDIA Jetson. (2019). Retrieved August 15, 2019, from https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/

Ongaro, D. (2014). *Consensus: Bridging theory and practice* (Doctoral dissertation). Stanford University. http://purl.stanford.edu/qr033xr6097

Ongaro, D., & Ousterhout, J. (2014). In search of an understandable consensus algorithm. *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, 305–320. http://dl.acm.org/citation.cfm?id=2643634.2643666

OpenFog Architecture Overview. (2016). Retrieved August 1, 2018, from https://www.openfogconsortium.org/wp-content/uploads/OpenFog-Architecture-Overview-WP-2-2016.pdf

OpenFog Consortium. (2016). Retrieved August 1, 2018, from https://www.openfogconsortium.org

Panadero, J., de Armas, J., Serra, X., & Marquès, J. M. (2018). Multi criteria biased randomized method for resource allocation in distributed systems: Application in a volunteer computing system. *Future Generation Computer Systems*, *82*, 29–40. https://doi.org/10.1016/j.future.2017.11.039

Park, J., Yu, H., Chung, K., & Lee, E. (2011). Markov Chain Based Monitoring Service for Fault Tolerance in Mobile Cloud Computing. *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, 520–525. https://doi.org/10.1109/WAINA.2011.10

Petcu, D. (2013). Multi-cloud: Expectations and current approaches. *Proceedings of the 2013 International Workshop on Multi-Cloud Applications and Federated Clouds*, 1–6. https://doi.org/10.1145/2462326.2462328

Petrov, C. (2019). *40 Internet Of Things Statistics From 2019 To Justify The Rise Of IoT*. Retrieved December 15, 2019, from https://techjury.net/stats-about/internet-of-things-statistics/%7B%5C#%7Dgref

Pham, X. Q., & Huh, E. N. (2016). Towards task scheduling in a cloud-fog computing system. *18th Asia-Pacific Network Operations and Management Symposium, APNOMS 2016: Management of Softwarized Infrastructure - Proceedings*. https://doi.org/10.1109/APNOMS.2016.7737240

Poyraz, E., & Memik, G. (2016). Analyzing power consumption and characterizing user activities on smartwatches: Summary. *Proceedings of the 2016 IEEE International Symposium on Workload Characterization, IISWC 2016*, 219–220. https://doi.org/10.1109/IISWC.2016.7581282

Qi, H., & Gani, A. (2012). Research on mobile cloud computing: Review, trend and perspectives. *2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP)*, 195–202. https://doi.org/10.1109/DICTAP.2012.6215350

Raspberry Pi. (2019). Retrieved October 20, 2019, from https://www.raspberrypi.org/

RaspberryPi.org. (n.d.-a). *Raspberry Pi 3 Model B*. Retrieved June 28, 2020, from https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

RaspberryPi.org. (n.d.-b). *Raspberry Pi 3 Model B+*. Retrieved June 28, 2020, from https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/

Raspberrypi.org. (2019). *Raspbian*. Retrieved October 20, 2019, from https://downloads.raspberrypi.org/raspbian_lite_latest

Rec, C. (2018). *Building Connected Vehicle Solutions on the AWS Cloud*. Retrieved August 1, 2018, from https://aws.amazon.com/blogs/iot/building-connected-vehicle-solutions-on-the-aws-cloud/

Reinfurt, L., Breitenbücher, U., Falkenthal, M., Leymann, F., & Riegg, A. (2016). Internet of things patterns. *Proceedings of the 21st European Conference on Pattern Languages of Programs*. https://doi.org/10.1145/3011784.3011789

ROS, Powering the world's Robots. (2019). Retrieved, from https://www.ros.org/

Ryden, M., Oh, K., Chandra, A., & Weissman, J. (2014). Nebula: Distributed edge cloud for data-intensive computing. *2014 International Conference on Collaboration Technologies and Systems (CTS)*, 491–492. https://doi.org/10.1109/CTS.2014.6867613

Samsung. (n.d.-a). *EVO microSD Memory Card 32GB*. Retrieved June 28, 2020, from https://www.samsung.com/in/memory-storage/evo-microsd-card-with-sd-adapter-95/MB-MP32GAIN

Samsung. (n.d.-b). *EVO Plus microSD Memory Card 128GB*. Retrieved June 28, 2020, from https://www.samsung.com/in/memory-storage/evo-plus-microsd-card-with-sd-adapter-100/MB-MC128GAIN/

Sanaei, Z., Abolfazli, S., Gani, A., & Buyya, R. (2014). Heterogeneity in mobile cloud computing: Taxonomy and open challenges. *IEEE Communications Surveys and Tutorials*, *16*(1), 369–392. https://doi.org/10.1109/SURV.2013.050113.00090

Sanaei, Z., Abolfazli, S., Gani, A., & Hafeez, R. (2012). Tripod of Requirements in Horizontal Heterogeneous Mobile Cloud Computing.

Sanaei, Z., Abolfazli, S., Gani, A., & Shiraz, M. (2012). SAMI: Service-based arbitrated multi-tier infrastructure for Mobile Cloud Computing. *2012 1st IEEE International Conference on Communications in China Workshops (ICCC)*, 14–19. https://doi.org/10.1109/ICCCW.2012.6316466

Satyanarayanan, M. (1996). Fundamental challenges in mobile computing. *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, 1–7. https://doi.org/10.1145/248052.248053

Satyanarayanan, M. (1993). Mobile computing. *Computer*, *26*(9), 81–82. https://doi.org/10.1109/2.231283

Satyanarayanan, M. (2017). The emergence of edge computing. *Computer*, *50*(1), 30–39. https://doi.org/10.1109/MC.2017.9

Satyanarayanan, M., Bahl, P., Caceres, R., & Davies, N. (2009). The Case for VM-Based Cloudlets in Mobile Computing. *Pervasive Computing, IEEE*, *8*(4), 14–23. https://doi.org/10.1109/MPRV.2009.82

Satyanarayanan, M., Chen, Z., Ha, K., Hu, W., Richter, W., & Pillai, P. (2014). Cloudlets: at the Leading Edge of Mobile-Cloud Convergence. *Proceedings of the 6th International Conference on Mobile Computing, Applications and Services*, 1–9. https://doi.org/10.4108/icst.mobicase.2014.257757

Satyanarayanan, M., Schuster, R., Ebling, M., Fettweis, G., Flinck, H., Joshi, K., & Sabnani, K. (2015). An open ecosystem for mobile-cloud convergence. *IEEE Communications Magazine*, *53*(3), 63–70. https://doi.org/10.1109/MCOM.2015.7060484

Satyanarayanan, M., Simoens, P., Xiao, Y., Pillai, P., Chen, Z., Ha, K., Hu, W., & Amos, B. (2015). Edge Analytics in the Internet of Things. *IEEE Pervasive Computing*, *14*(2), 24–31. https://doi.org/10.1109/MPRV.2015.32

SETI@home. (2020). Retrieved May 22, 2020, from https://setiathome.berkeley.edu/

Shi, W., Cao, J., Zhang, Q., Li, Y., & Xu, L. (2016). Edge Computing: Vision and Challenges. *IEEE Internet of Things Journal*, *3*(5), 637–646. https://doi.org/10.1109/JIOT.2016.2579198

Shila, D. M., Shen, W., Cheng, Y., Tian, X., & Shen, X. S. (2017). AMCloud: Toward a Secure Autonomic Mobile Ad Hoc Cloud Computing System. *IEEE Wireless Communications*, *24*(2), 74–81. https://doi.org/10.1109/MWC.2016.1500119RP

Shiraz, M., Gani, A., Khokhar, R. H., & Buyya, R. (2013). A review on distributed application processing frameworks in smart mobile devices for mobile cloud computing. *IEEE Communications Surveys Tutorials*, *15*(3), 1294–1313. https://doi.org/10.1109/SURV.2012.111412.00045

Shye, A., Scholbrock, B., Memik, G., & Dinda, P. A. (2010). Characterizing and modeling user activity on smartphones: Summary. *ACM SIGMETRICS Performance Evaluation Review*, *38*(1), 375–376. https://doi.org/10.1145/1811099.1811094

Skarlat, O., Schulte, S., & Borkowski, M. (2016). Resource Provisioning for IoT Services in the Fog Resource Provisioning for IoT Services in the Fog. *2016 IEEE 9th International Conference on Service-Oriented Computing and Applications Resource*, (November). https://doi.org/10.1109/SOCA.2016.10

Stojmenovic, I. (2012). Keynote 1: Mobile Cloud and Green Computing. *Procedia Computer Science*, *10*, 18–19. https://doi.org/10.1016/j.procs.2012.06.004

Stutzbach, D., & Rejaie, R. (2006). Understanding Churn in Peer-to-peer Networks. *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, 189–202. https://doi.org/10.1145/1177080.1177105

Taivalsaari, A., & Mikkonen, T. (2018). A Taxonomy of IoT Client Architectures. *IEEE Software*, *35*(3), 83–88. https://doi.org/10.1109/MS.2018.2141019

Tang, S. (2019). *A List of Chip/IP for Deep Learning*. Retrieved August 10, 2019, from https://medium.com/@shan.tang.g/a-list-of-chip-ip-for-deep-learning-48d05f1759ae

Terraform.org. (2020). *Terraform*. Retrieved June 29, 2020, from https://www.terraform.io/

The Economist. (2016). *After Moore's law, The future of computing*. Retrieved August 1, 2018, from https://www.economist.com/News/Leaders/21694528-Era-Predictable-Improvement-Computer-Hardware-Ending-What-Comes-Next-Future

Triggs, R. (2020). *Does Moore's Law still apply to smartphones in 2020?* Retrieved March 7, 2020, from https://www.androidauthority.com/moores-law-smartphones-1088760/

Unikernels, Rethinking Cloud Infrastructure. (2019). Retrieved October 20, 2019, from http://unikernel.org/

Valentino, R., Jung, W.-s., & Ko, Y.-b. (2018). Opportunistic computational offloading system for clusters of drones. *2018 20th International Conference on Advanced Communication Technology (ICACT)*, 303–306. https://doi.org/10.23919/ICACT.2018.8323734

Van Canneyt, S., Bron, M., Lalmas, M., & Haines, A. (2019). Describing patterns and disruptions in large scale mobile app usage data. *26th International World Wide Web Conference 2017, WWW 2017 Companion*, 1579–1584. https://doi.org/10.1145/3041021.3051113

Vaquero, L. M., & Rodero-Merino, L. (2014). Finding your Way in the Fog. *ACM SIGCOMM Computer Communication Review*, *44*(5), 27–32. https://doi.org/10.1145/2677046.2677052

Varghese, B., & Buyya, R. (2018). Next generation cloud computing: New trends and research directions. *Future Generation Computer Systems*, *79*, 849–861. https://doi.org/https://doi.org/10.1016/j.future.2017.09.020

Villari, M., Fazio, M., Dustdar, S., Rana, O., & Ranjan, R. (2016). Osmotic Computing: A New Paradigm for Edge/Cloud Integration. *IEEE Cloud Computing*, *3*(6), 76–83. https://doi.org/10.1109/MCC.2016.124

Vizard, M. (2017). *Cisco to run containers at the network edge*. Retrieved, from https://containerjournal.com/features/cisco-run-containers-network-edge/

Xu, Q., Erman, J., Gerber, A., Mao, Z., Pang, J., & Venkataraman, S. (2011). Identifying diverse usage behaviors of smartphone apps. *Proceedings of the 2011 ACM SIGCOMM Conference on Internet Measurement Conference*, 329–344. https://doi.org/10.1145/2068816.2068847

Yannuzzi, M., Milito, R., Serral-Gracia, R., Montero, D., & Nemirovsky, M. (2014). Key ingredients in an IoT recipe: Fog Computing, Cloud computing, and more Fog Computing. *2014 IEEE 19th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks, CAMAD 2014*, 325–329. https://doi.org/10.1109/CAMAD.2014.7033259

Yaqoob, I., Ahmed, E., Abdullah, G., Salimah, M., Muhammad, I., & Sghaier, G. (2016). Mobile ad hoc cloud: A survey. *Wireless Communications and Mobile Computing*, *16*(16), 2572–2589. https://doi.org/10.1002/wcm.2709

Yaqoob, I., Ahmed, E., Gani, A., Mokhtar, S., & Imran, M. (2017). Heterogeneity-Aware Task Allocation in Mobile Ad Hoc Cloud. *IEEE Access*, *5*, 1779–1795. https://doi.org/10.1109/ACCESS.2017.2669080

Zaghdoudi, B., Ayed, H. K. B., & Gnichi, I. (2017). A protocol for setting up ad hoc mobile clouds over spontaneous MANETs: A proof of concept. *2016 Cloudification of the Internet of Things, CIoT 2016*, 1–6. https://doi.org/10.1109/CIOT.2016.7872919

Zaghdoudi, B., Ayed, H. K. B., & Riabi, I. (2015). Ad hoc cloud as a service: A protocol for setting up an ad hoc cloud over MANETs. *Procedia Computer Science*, *56*(1), 573–579. https://doi.org/10.1016/j.procs.2015.07.256

Zhang, Y., Niyato, D., & Wang, P. (2015). Offloading in mobile cloudlet systems with intermittent connectivity. *IEEE Transactions on Mobile Computing*, *14*(12), 2516–2529. https://doi.org/10.1109/TMC.2015.2405539

Zhong, L., Wang, B., & Wei, H. (2012). Cloud computing applied in the mobile Internet. *2012 7th International Conference on Computer Science & Education (ICCSE)*, 218–221. https://doi.org/10.1109/ICCSE.2012.6295061

Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge Intelligence: Paving the Last Mile of Artificial Intelligence With Edge Computing. *Proceedings of the IEEE*, *107*(8). https://doi.org/10.1109/JPROC.2019.2918951