Towards an Open, Secure, Decentralized and Coordinated Fog-to-Cloud Management Ecosystem

D2.1 Tracking Scientific, Technology and Business Trends (Version 1)

Project Number	**730929**
Start Date	**01/01/2017**
Duration	**36 months**
Topic	**ICT-06-2016 - Cloud Computing**

Work Package	**WP2, Technology survey, business models and architectural definition**
Due Date:	*M3*
Submission Date:	*30/03/2017*
Version:	*0.9*
Status	*Final*
Author(s):	*Alec Leckey, John Kennedy (INTEL),* *Jens Jensen, Shirley Crompton (STFC),* *Anna Queralt, Daniele Lezzi, Jorge Ejarque (BSC),* *Sandeep Singh (TUBS),* *Matija Cankar, Jolanda Modic (XLAB)* *Lara Lopez, Roi Sucasas Font (ATOS)* *Antonio Salis, Glauco Mancini (TISCALI)* *Eva Marín, Vitor Barbosa, Ester Simo, Alex Gomez Cardenas, Jordi Garcia, Sergi Sánchez López (UPC),* *Andrea Bartoli, Francisco Hernandez (WOS),* *Marc-Elian Bégin, Charles Loomis (SIXSQ)*
Reviewer(s)	*Ana Juan Ferrer (ATOS)* *Xavi Masip (UPC)* *Admela Jukan (TUBS)*

Project co-funded by the European Commission within the H2020 Programme		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission)	
RE	Restricted to a group specified by the consortium (including the Commission)	
CO	Confidential, only for members of the consortium (including the Commission)	

Version History

Version	Date	Comments, Changes, Status	Authors, contributors, reviewers
0.1	27/01/2017	Initial ToC added and doc structure	Alec Leckey (INTEL)
0.2	04/03/2017	All contributions added	All partners
0.3	05/03/2017	Contributions formatted and first version integrated	Section leaders
0.4	06/03/2017	First initial review	Ana Juan (ATOS), Xavi Masip (UPC), Admela Jukan (TUBS)
0.5	21/03/2017	All comments addressed	All partners
0.6	23/03/2017	Final review	Ana Juan (ATOS), Xavi Masip (UPC), Admela Jukan (TUBS)
0.7	29/03/2017	All comments addressed	All partners
0.8	30/03/2017	Final version released	Alec Leckey (INTEL)
0.9	31/03/2017	Quality check	Lara López (ATOS)

Table of Contents

List of figures

List of tables

_Toc472584844

Executive Summary

The objective of this deliverable is to track scientific, technology and business trends in the area of Fog and Cloud computing that are relevant to the mF2C project. This deliverable gives a generic overview of all these trends of which awareness is necessary for this project, and for the proposed area of research at large. Each chapter ends with a “key takeaways” section summarizing the main points of focus, helping us to understand where to prioritize during the project. This is the initial version of the deliverable (v1) which is aligned to iteration 1 (IT-1) of the project. A second version is due in M22 which is aligned to iteration 2 (IT-2). A final version is then due in M34 which will include global reporting on technology, business models and scientific trends.

1. Introduction

1.1 Introduction

Fog computing brings cloud computing capabilities closer to the end-device and users, enabling location-dependent resource allocation, low latency services, and significantly extending the IoT services portfolio as well as market and business opportunities in the cloud sector. The number of connected devices is expected to grow at exponential rates so cloud and fog models are expected to emerge. This will allow for shared, collaborative, extensible mobile, volatile and dynamic compute, storage and network infrastructure. We use the term Fog-to-Cloud (F2C) to refer to the new stack of resources created through this merging of cloud and fog computing. This creates the need for a new, open and coordinated management ecosystem. The mF2C project will deliver an open, secure, decentralized, multi-stakeholder management framework, which will include novel programming models, privacy and security, data storage techniques, service creation, brokerage solutions, SLA policies, and resource orchestration methods. This framework will set the foundations for an innovative distributed system architecture, providing a proof-of-concept system and platform, to be tested and validated in real-world use cases.

One of the activities of WP2 is to study state-of-the-art fog, cloud, network and IT infrastructure technologies. These range from sensors and smart end-devices to high speed connections and advanced cloud services. The goal is to identify technologies that are relevant for the deployment of the mF2C architectural building blocks in the two proposed iterations, IT-1 and IT-2. Together with the architecture design which will be defined later in this work package, the outcome of this state-of-the-art analysis is a set of technological requirements – both at a conceptual and a technological level – that service providers will be required to fully deploy and benefit from the proposed architecture.

This deliverable gives a generic overview of all the scientific, technology and business trends in the area of fog computing that are relevant to the mF2C project and which awareness is required for this project. This is the initial version of the deliverable (v1) aligned to iteration 1 (IT-1) of the project. A second version is due in M22 aligned to iteration 2 (IT-2). A final version is then due in M34 which will include global reporting on technology, business models and scientific trends.

Chapter 2 reviews the scientific trends of Fog and Cloud computing, starting with a review of contributions relating to service management, resource management, and end-devices strategies for naming and addressing. While Cloud Computing is established compared to Fog Computing, issues related to strategies for optimal service placement and execution remain. We discuss protocols proposed in the field of IoT Management to support communications between end devices. This also includes techniques to identify objects being a prerequisite for the development, deployment, operation and exploitation of IoT applications and services. We also analyse trends coming from the field of High Performance Computing (HPC), in particular the problems found in the HPC, cloud, and big data areas, both at the infrastructure level and at the software level. Advances in technology have enabled researchers to collect, store and manipulate increasingly large and more complex datasets so methods to manage and process these are reviewed. Several security challenges specific to IoT systems are reviewed pertaining to trust, constrained resources and scalability. We review the different approaches, trust models, and security architectures that have been proposed to resolve these challenges. This also includes software integrity verification through remote attestation, a technique for remotely verifying a devices software integrity. Finally in this section, we review the convergence of Artificial Intelligence (AI) techniques including machine learning (ML), genetic algorithms (GA), fuzzy neural network, Markov decision process-based hidden models, and how they can be used to ensure the project's Fog-to-Cloud system will be intelligent as well as autonomous.

Chapter 3 looks at Technology trends in the area of Cloud Computing starting with the different tools and platforms currently available that enable the management of features such as storage, compute, machine instances, and containers. This is then expanded to include Fog Computing solutions, although most seem to be primarily focused at IoT. This is extended further with HPC as technology trends in this area are also reviewed under the headings of storage, architectural solutions, and software platforms. As orchestration is the entity that manages the interconnections and interaction among all cloud-based entities, we provide a definition here and review Cloud Management Platforms (CMP) that provide IaaS solutions. This also includes cloud container type approaches to decouple from underlying infrastructure and host filesystems.
Finally in this chapter, we discuss the role of standards in Fog Computing. While Cloud is a maturing compute model for which dedicated standards have already started to emerge. Fog computing is a very recent concept and is just beginning to be considered by standardisation initiatives. We review initial efforts in this area.

Chapter 4 reviews cloud, fog and IoT trends with respect to business needs. IoT will change the basis of competition and drive new business models for user and supplier companies. We review business surveys that show IoT will both enable and force new business models. In the future, it is expected that machines, products, systems, and people will communicate locally in real-time so that they can manage their needs in an efficient method. We discuss technologies that will provide benefits in the areas of productivity, quality, flexibility, information accuracy and safety. Methods to address security are reviewed from a business perspective including costs of Capital Expenditure (CapEx) and Operating Expenditure (OpEx). This also includes industry standards and best practices leading to adoption from standards organisations. Finally, we look at the evolution of technology and how it can have an impact on business as traditional business models applied to cloud computing are not sufficient to cover the dynamicity of all use cases.

Finally, we conclude with the key takeaways this document provides for the project. These include the areas of focus and where existing solutions can help accelerate development of the key components of the architecture.

1.2 Glossary of Acronyms

Acronym	Definition
6LoWPAN	Low Power Wireless Personal Area Networks
AI	Artificial Intelligence
CA	Certificate Authority
CapEx	Capital Expenditure
CID	Communication Identifier
CRTM	Core Root of Trust Measurement
DC	Data Center
DNS	Domain Name System
DRT	Dynamic Root of Trust
F2C	Fog-to-Cloud
GA	Genetic Algorithms

HPC	High Performance Computing
IaaS	Infrastructure as a Service
IMA	Integrity Measurement Architecture
IoT	Internet of Things
IIoT	Industrial Internet of Things
LHC	Large Hadron Collider
ML	Machine Learning
NVM	Non-Volatile Memory
ONS	Object Name Service
OpEx	Operation Expenditure
OTA	Over the Air
PaaS	Platform as a Service
PCT	Platform Configuration Register
SIoT	Social Internet of Things
TEE	Trusted Execution Environment
TLS	Transport Layer Security
TPM	Trusted Platform Module
URI	Uniform Resource Identifier
VM	Virtual Machine

Table 1. Acronyms

2. Scientific trends

2.1. Service management, resource management, end-devices

2.1.1. Service/Resource management in cloud and fog

This section aims at highlighting some of the most relevant issues and open challenges in the areas of cloud and fog computing, hence drawing the foundational context where the mF2C project must deploy the envisioned management architecture. The section starts by browsing the link between energy and quality of service in cloud computing to later shift the attention to fog computing.

Several contributions dealing with service management may already be found in both cloud and fog areas. Albeit Cloud Computing is with no doubt well established compared to Fog Computing, some issues remain unsolved, particularly addressing aspects related to strategies for optimal service placement and execution when considering both quality of service guarantees and energy savings. Indeed, even though quality of service provisioning is a key objective in the portfolio for any cloud provider, some critical aspects may also depend on that, such as for example energy consumption, since the amount of resources to be "consumed" strongly impacts on the global energy consumption. However, several platforms do not consider energy consumption when handling quality of service through their own resources management strategies –e.g., resource virtualization, service orchestration or VM migration. Despite the strong focus on green data centres (DC) design and implementation, so far driving many different contributions from the scientific community, service management in cloud computing still remains a challenge in real systems development. In particular, contributions [Beloglazov2012] and [Wolke2015] aims at the reduction of energy consumption in cloud data centres.

Managing energy consumption is obviously related to the amount of resources to be used. However, despite this assessment seems easy to handle in static scenarios like a DC –i.e., cloud providers may avail from the DC resources static behaviour–, VMs allocation, deallocation and migration policies inherent to a DC management strategy, may significantly add a non-negligible dynamicity component. To cope with this dynamicity, authors in [Xiao2013] propose a load prediction algorithm as part of a strategy to achieve efficient tradeoff between QoS and energy consumption in VMs distribution. Besides green computing, another trend is worth mentioning in cloud computing arena is regarding the autonomic and distributed adaption in cloud resources.

[Hummaida2016] and identifies the main research challenges in cloud system adaptation, such as the workloads characterization and accurate online profiling as well as development of adaptation strategies offering high scalability. The autonomic resource management analysis presented in [Singh2016] takes into consideration both general and QoS-aware self-management challenges. Regarding the distributed resource management in cloud, [Chaisiri2012] have proposed the so-called optimal cloud resource provisioning (OCRP) algorithm, which aims at the optimal management of resources in multiple cloud providers, through the employment of stochastic integer programming.

Moving towards Fog Computing, we realize that service management is currently more subject to research than deployment. Indeed, Cloud and Fog, even though conceptually similar, present crucial differences. Unlike cloud DCs, service orchestration in Fog Computing systems must deal with the particular constraints brought by the edge devices, such as high dynamicity, mobility, energy issues, reliability, security and heterogeneity, to name a few. In [Simoens2015], authors present the so-called FUSION framework including its main architectural aspects concerning services orchestration in fog aiming at the reduction of the Service Response Time (SRT) through the employment of parallelism and service-chaining. Authors employ the *session slot* concept in order to express resources capacity by means of the number of service sessions it can handle. Hence, the service composition is represented by a graph, where each node is an instance of the required service whilst the edges are associated to the employed metric for data path cost. In [Mukherjee2015], it is presented a service orchestration strategy to divide the execution of a service with high processing requirements into small tasks. The resulting tasks may be executed in parallel and the partial results are then combined into the final result. In that work, authors implement the distributed management and execution of services in a real scenario constituted by constrained edge devices. Service management in [Alam2016] is performed by means of multi-agent model. Therefore, execution and accessing time of mobile services can be diminished through the employment a distributed reinforcement learning algorithm which enables decentralized code execution. On the other hand, [Kwon2012], the employed offloading strategy is focused in achieving energy-efficiency. Authors, propose a strategy that enables mobile services distribution based on a threshold to offload the service execution to a remote server. However, in this approach, the service offloading is done without partitioning. Rather, the service execution is replicated to the remote server and, under energy-intensive functionality, the execution state is replicated between edge device and remote server.

Works focused in the F2C architecture such as [Masip2016] and [Souza2016] have further studied the combination of fog and cloud resources presenting the upcoming challenges in resource/service management in the proposed architecture. On the other hand, authors in [Skala2015] present an architecture composed by cloud, fog and dew layers where dew computing layer leverages smart edge devices to create an ad-hoc and self-adaptive new layer logically placed bellow fog computing layer in the distributed computing hierarchy.

We may summarize that despite many efforts have been done in the area of resources management, the focus of recent researches in cloud and fog computing converge only partially. Indeed, recent researches in cloud computing resource management have focused mostly in energy consumption aspects whilst few works in self-managing networks are yet available. Besides the fact that fog computing inherits challenges not addressed in cloud computing, fog research is much broader due to its relative novelty in comparison to cloud and the dynamicity, volatility and large heterogeneity observed in fog devices.

2.1.2. End Devices / IoT Management

In this subsection, we revisit recent contributions coming from the research community focused on end devices and IoT management. For the sake of global understanding we consider that IoT devices are sensors and/or wireless sensor networks (WSN) with a high degree of heterogeneity, different data formats and communication protocols, etc. The following trends are worth noting:

- Many different protocols have been proposed in the field of IoT management addressing specific problems. We may group some of the proposed contributions into the following:
- from the point of view of the resource discovery: mDNS (RFC6763), Hypercat [Hypercat], etc.
- from the point of view of the data: XMPP [Xmpp], REST [Zeng2011], etc.,
- from the point of view of the communication: ZigBee [Zigbee], CoAP [CoAP], EnOcean [EnOcean], etc.,
- from the point of view of the semantic approach: Semantic Sensor Net Ontology W3C [W3] IOTDB [iotdb],
- from the point of view of the IoT service OSGI [osgi].

Other protocols proposing interfaces and frameworks addressing the problem from a multilayer perspective, that is, for example, from the point of the view of the data and communications layer developed by the Open

Mobile Alliance such as, LightweightM2M, a client-server communication protocol, also including a secure data transfer (CoAP) at [openmobilealliance].

- Some of the recent proposals in the literature addressing the IoT management are also based on the listed protocols, utilizing and combining them as well as proposing modifications of them, and platforms managing a combination of these protocols.
- Some works consider IoT devices as mere data consumer/producer (sensors and actuators); then the management from the point of the data consists on reading data from sensors and sending commands to actuators. Few works consider the computer capacity of the IoT devices, but do not propose strategies to manage such a computer capacity.
- It is worth mentioning that many proposals on IoT management come from the area of Industrial Internet of Things.

Although there are many contributions in the area of IoT standards, there is not a widely accepted consensus on the solution to be adopted. As a consequence, each type of sensor and edge device utilizes its own format data, communication protocol, etc., thus driving a difficult interoperability. To solve this issue, some proposals propose the use of an IoT gateway to serve as the interface between IoT devices and cloud. In short an IoT gateway offers a service of data filtering, aggregation, communication with cloud, protocol translation and in some cases, also provides the first level of data computation and security.

Taking into account the considerations made in the previous paragraphs, recent works in IoT devices management can also be classified depending on their focus. Indeed, contributions in [Cai2014] and [Främling2014] are based on a semantic approach, aiming at hiding the inherent heterogeneity and particularity of IoT devices, through a classification and categorization of resources (and even services) into different categories, providing a necessary abstraction layer. The abstraction layer is the interface between IoT devices and cloud, or more in general service requesters. Work [Cai2014] focuses on a manufacturing IoT environment proposing a framework for the management of the product lifecycle, where: i) resources and services are managed based on ontologies, and; ii) services are also decomposed into atomic operations. In this scenario, IoT applications are interpreted in a semantic level matching atomic operations with abstract resources. Authors in [Främling2014] propose a new standard for IoT messaging also based on ontologies.

Contributions in [Petrolo2017], [Kim2015] and [Vögler2015] propose the use of IoT gateways acting as an interface between sensor networks and cloud. Besides, in [Petrolo2017] and [Kim2015] the IoT gateway management provides sensor data abstraction according to an ontology model. Although most contributions do not consider the computer capacity of edge devices, some of the existing proposals pose the IoT gateway computer capacity to be used to provide a first level of data computation, thus avoiding the need to send all sensor raw data to cloud. This is the case of [Petrolo2017], where in order to allocate services to IoT gateways and due to their usual limited CPU and storage capacity, a container-based virtualization as a lightweight alternative to hypervisor-based virtualization is proposed. In a different strategy [Vögler2015] also proposes to execute services (or part of them) in the IoT gateways, although considering a large-scale deployment with thousands of different IoT gateways, enabling the provisioning of service components through installable application packages.

The works reviewed so far may be classified as IoT service/data/resource management. However, from a different perspective, a current trend in IoT network management, is the use of Software-Defined Networks (SDN) based concepts. Due to the high heterogeneity of networks connecting edge devices –including sensors networks–, along with the inherent mobility of edge devices, [Wu2015] proposes the use of a software-defined IoT system, based on an abstraction overlay, for ubiquitous flow control and mobility management in multinetworks.

Finally, another trend in IoT management relates to Social Internet of Things (SIoT) and user-centric network concepts, which introduce the idea of social relationships among objects. Authors in [Chen2016] propose an IoT trust protocol for IoT systems with application in service composition. In this service composition, the service requester selects the device providing the service with the highest trust value. This trust value is based on the relationship between objects, which is also related to the humans' owners of the objects, and to previous experiences with the same device.

As a final conclusion, we may conclude that IoT management proposals rarely address the IoT management from a whole entire perspective, including data, resources, service, network, etc. One of the works trying to address IoT management from all the possible perspectives is [Al-Fuqaha2015]. In this paper, authors review recent literature addressing IoT from the point of view of proposed architectures, mechanisms of addressing IoT devices, communication technologies, categorization of IoT services, proposed protocols for application, service discovery, routing/communication in IoT, security, QoS, etc. On the other hand, existing works addressing data and service management are usually based on semantics and leverage ontologies to classify sensors, data and services. Last but not least, IoT management is frequently addressed by using IoT gateways, providing an interface to overcome the interoperability issues coming from the unavoidable heterogeneity.

2.1.3. End devices / Naming and Addressing

A key challenge that needs to be addressed in an Internet of Things scenario is the identification (naming and addressing) of the end devices. Therefore, the availability of a technique for unambiguously identify those objects (physical and virtual entities) is a key prerequisite for the development, deployment, operation and exploitation of IoT applications and services [European2014].

In this section some of the most common identification (naming and addressing) technologies that are being used nowadays are presented. In addition, some novel proposals have been included aiming at showing a comprehensive panorama of the state of the art.

2.1.3.1. IP, Domain Name System and Uniform Resource Identifier

The most common naming scheme in the current Internet architecture is the Uniform Resource Identifier (URI), which is used to identify a name of a web resource. Once the desired resource is located, the IP structure and related technologies provide reliable connection to it. The Domain Name System (DNS) enables people to use URI, which is easier to remember compared to an IP address, to reach out to a certain web resource without knowing the IP address of it [Yijian2013].

However, some contributions are posing some doubts about a common, global and unique identifier. Contribution in [Sandoche2017] argues that a universal naming scheme is nearby impossible, mainly because many industries have been using their own proprietary naming conventions for long time, what makes extremely difficult a potential migration to a global naming convention –that will undoubtedly impact on their infrastructure. Instead, a feasible alternative will be to let the different sectors in the IoT use their existing naming conventions, but to evolve the naming service (DNS) to resolve the IoT identifiers (using the existing or new naming conventions) to its related digital information in the internet.

2.1.3.2. IPv6 and IPv6 over Low Power Wireless Personal Area Networks

IPv6 was proposed as a solution for the originally limited IPv4 address space (2^{32}). By providing 2^{128} possible addresses, it is feasible to assign a unique IPv6 address to every device in the world, what will support the needs demanded by the constant and unstoppable Internet-connected-objects population growth, which, according to certain predictions will contain 50 to 100 billion connected things by 2020.

IPv6 over Low power Wireless Personal Area Networks (6LoWPAN) is a solution aiming at replacing proprietary communication protocols like ZigBee, by enabling constrained devices (mainly sensors) to connect directly to the global Internet. 6LoWPAN works with the 802.15.4 standard and extends IPv6 to the devices using the communication technologies described in the cited standard [Yijian2013].

2.1.3.3. Global IP Protocol and Access Address/Identifier

One of the 6LoWPAN's inherent disadvantages is its relatively large overhead (26-41 bytes) that may highly limit the payload size within an IEEE 802.15.4 frame (standard used in very constrained devices). Taking the 25 bytes with the extended MAC address into account, overhead will amount up to 52% out of the 127 byte-long Maximum Transfer Unit (MTU) provided by the IEEE 802.15.4 standard. This will undoubtedly cause the system to be severely inefficient [Yijian2013].

Glowbal IP protocol is a solution whose purpose is to extend the IPv6 and 6LoWPAN support to all kind of sensors and legacy devices already deployed in the cities and industry sectors. In order to address the overhead problem described before, authors in [European2014] suggested an Access Address Identifier (AAID) and an AAID-IPv6 translation mechanism in order to adapt any Internet connected device to the IoT architecture via IPv6. The goal of the AAID is to simplify all connection parameters from IPv6 (source address and port and destination address and port) in a single 4-bytes communication identifier.

2.1.3.4. GS1 Identification Keys and Object Naming Service

The GS1 Identification Key is a naming scheme created by the GS1 organization. The key steps to assign a GS1 Identification Key include getting a company prefix, assigning numbers and selecting a bar code and its related parameters. The generated keys can be used to keep track the status and location of the object all over the world.

Later on, the GS1 also introduced the Object Name Service (ONS) in order to allow objects to be located using GS1 Identification Keys via Internet. Here, the GS1 takes advantage of the DNS transforming GS1 Identification Keys in a DNS usable format. Considering the fact that DNS is accessible worldwide and that ONS doesn't need to modify the DNS server, this solution could be adopted into a traditional structure without major changes, what can be translated in a cheaper integration, in comparison with other proposals [Yijian2013].

2.1.3.5. Communication Identifier System

The Communication Identifier (CID) System is a public user oriented Internet of Things identifier naming system. The CID system is responsible for distributing, managing, storing and querying CID identifiers, which are unique names for Internet of Things devices.

Each Communication Identifier is composed by three different parts: compatibility domain, type domain and information domain. While compatibility and type domains are optional fields, the information domain is a mandatory field [European2014].

2.1.3.6. Sensor Web Enablement and Sensor UID

Sensors are the main source of real-time data in the Internet. Due to its great diversity they can collect almost any kind of information from the environment they are deployed at. In order to find, share and access heterogeneous sensors systems across different infrastructures, the Sensor Web Enablement (SWE) standardizes mechanisms to discover sensors, determine sensors capabilities, access sensor parameters, retrieve real-time observations and publish alerts based on sensed data [European2014].

Among all major SWE standards, Sensor Model Language (SensorML) is the one directly related to the naming scheme for sensors [Yijian2013]. In general, SensorML is used to describe a sensor system as a process chain composed of sub processes with inputs and outputs. The standard also provides static metadata needed for discovery of sensors, such as name, type, manufacturer, location, etc.

2.1.3.7. Entity Code (ECode) System

The Entity Code (Ecode) System standards the coding structure and the distribution principle of Ecode identifiers, which is a uniform, compatible coding scheme for Internet-connected-objects. Each code identifier consists of three different parts, including Version (V), Numbering System Identifier (NSI) and Master Data (MD). The length of NSI and MD is decided by the V of Ecode [European2014].

2.1.3.8. Name Data Networking

Nowadays the Internet architecture is a host-centric network where devices are identified across the network using a unique IP address. Thus, the desired content in the current Internet is located by the address of the host it is stored.

The Name Data Networking (NDN) is a Content-Centric-Network (CCN) architecture under development that focuses on data and information rather than end-to-end communications [European2014]. In NDN, users send data packages out to the Internet with information about the interests they want to reach rather than a host address.

2.1.3.9. Object Identifier

The Object Identifier (OID) is a naming scheme that uses an encoding strategy for uniquely identifying objects within a given scope –either worldwide or limited. OID has a very good foundation for global applications and has been successfully deployed in many fields in China, such as information security, eHealth service, network management, sensor network and RFID [European2014].

The assigned identifiers to objects are string of numbers that are allocated in a hierarchical manner, so that, for instance, the only one that can say what “1.2.3.4” stands for is its parent, who in this case is the object “1.2.3” [Sandoche2017].

2.1.3.10. MobilityFirst

This is a future Internet architecture developed with the main purpose of effectively supporting the increasing need for mobility of the different Internet connected objects, such as, sensors, cars, smartphones, etc. The most remarkable characteristic in MobilityFirst is the differentiation between the host name and the host address. In MobilityFirst the end points are represented by IDs that are unique. Those IDs are assigned using the Global Unique Identification (GUID) naming scheme.

In the envisioned MobilityFirst architecture, despite the GUID and network address is separated from each other, the network still maintains the mapping between them both [Yijian2013].

The GUID is a string that concatenates the user’s public key and a hash string, function of a parameter *x*, where *x* can be any information chosen from the IoT device, for example, the serial number, MAC address, or other. The Global Unique Identification will be unique as long as the user’s public key used to generate it is also unique. A single user can use the same public key to generate GUIDs for many different objects, being then the differentiation factor the hash side of the string [Yijian2013].

2.2. Scientific trends coming from the HPC area

2.2.1. Data Management Trends

Storage systems are undergoing fundamental changes to keep up with the requirements of new applications and services. These new requirements arise from problems found in the HPC, cloud, and big data areas, both at the infrastructure level and at the software level.

The current trends regarding infrastructure are focused on accelerating access to data, either by improving storage technologies, or by proposing new architectures.

New storage technologies received much attention with the advent of Flash-based storage devices. Future disruptive changes to the I/O hierarchy will be from the use of emerging, byte addressable Non-volatile Memories (NVM) [Coburn2011, Wong2010]. NVM technologies like phase-changed RAM, MRAM, or RRAM will significantly shrink the current performance gap between non-persistent byte-addressable DRAM and persistent, block-addressable storage, reducing access times down to 10 ns [Queralt2015]. In this context, the NextGenIO project [NextGenIO] has the goal to develop a prototype HPC system that takes advantage of this technology to provide greatly improved I/O performance for applications.

The main architectural trend nowadays is bringing storage and computation closer in the data center. Traditionally, persistent storage has been placed behind a storage area network (SAN) for scaling and management purposes. With current technology trends, it becomes important that storage (and NVM) moves closer to the compute nodes to further improve performance and reduce energy footprint. This is a significant architectural shift and requires fundamentally different approaches to storing, caching, replicating, and moving data. These issues, among others regarding the convergence of HPC and cloud infrastructures, are investigated in the BigStorage project [BigStorage].

At the software level we find NoSQL databases, which are increasingly used to manage persistent data in HPC applications in several domains, such as life sciences or engineering applications. NoSQL databases provide a mechanism for storing and accessing data that overcomes the scalability problems of relational databases. There are different types of NoSQL databases depending on their data model, such as key-value, graph, or document databases. The most appropriate one depends on the particular problem to be solved.

2.2.2. Programming Models Trends

Related to the implementation of applications, there are critical challenges to solve from a data management perspective due to strong impact of the data volume, velocity, and variety. Despite the multiple and different solutions already available, it can be easily argued that there is no unique solution for big data applications; on the contrary, multiple classes of Big Data systems are emerging, that are better fitted to streaming data problems (e.g., Watershed, Storm), to general data mining/machine learning tasks (e.g, Spark, Anthill), to array-based data analysis and On-Line Analytical Processing on large datacubes (e.g. SciDB, Rasdaman, Ophidia), etc.

Among these frameworks, Hadoop [Hadoop] and Spark [Spark] have attracted the most interest from the general public; Spark in particular is becoming very popular and has created a community of contributors focused on its improvement and exploitation for a large number of different projects and objectives. This constitutes one of the key advantages of Spark and has resulted in an increasing ecosystem that makes it attractive for a larger public.

The low level of integration amongst all these systems represent a strong barrier to address the implementation of big data scenarios. The EUBra-BIGSEA project [BIGSEA] is addressing these issues through the development of an integrated platform that allows the user to specify applications that combine different types of data and processing elements and instantiate them in a cloud environment.

2.3. Applications in different science areas (health, etc.), science data centers, big data processing type in science

The changing scale of research data generation is ushering in new approaches to processing and analysing data. For large-facility based big science research, there are two specific challenges arising from the exponential growth in the volume of experimental data generated and the analysis chains become increasingly complex. Firstly, it is simply not practical in many cases for researchers to transport the data back to their home institution and perform analyses locally. Secondly, in high throughput experiments, the experimental pipeline needs to be fine-tuned at runtime to the data output by the initial runs.

To tackle the first challenge, many facility providers are exploring how best to provide additional computing resources to enable users to access and analyse their data remotely. At STFC Rutherford Appleton Laboratory, there is co-location of large-scale experimental facilities with a dedicated data-centre hosting large-scale data archives, computing capabilities and specialist expertise. This gives the opportunity to coordinate into one cloud-based service giving location independent access to in- and post-experiment computing support for facility users to help them analyse their experimental data [Barnsley2016]. West-Life (http://about.west-life.eu/) is another example of cloud-based virtual research platform targeting the structural biologist community funded by H2020; while PaNDaaS [PaNDaaS] is a community effort developing Data Analysis as a Service for the Photon and Neutron domain.

A response to the second challenge is the timely analysis of raw data output from the experiment to inform on the subsequent steps in the live experiment process. Both STFC ISIS and Diamond facilities are building dedicated data analysis platforms using cluster and High Performance computers (HPC) to enable on-the-fly processing of experimental data to provide live feedback to steer the experimental process. The ULTRA project exemplifies such efforts. It is delivering an HPC data analysis platform for tomographic image reconstruction which combines high-end computing technologies, including high speed data acquisition, high throughput data transfer, cluster computing, parallel rendering and remote visualisation to enable end-to-end fast parallel reconstruction workflows for STFC facility users [Yang2015].

Beyond the facility-based Big Science research, HPC, cloud computing and Big Data techniques are being increasing exploited to analyse the broad spectrum of digital data generated as we progress through our daily life. While businesses exploit the derived information to improve their productivity and profitability, researchers use the intelligence to deliver better study outcome and innovations to improve qualities of life. To illustrate, we provide two use cases in the area of e-Health that exploit the current digital landscape (e.g. mobile telemetry, IoT, HPC, cloud storage etc.) to deliver better patient care and value for the public health system.

In the UK, unscheduled hospitalization in long-term care is a major cost to the National Health Service (NHS) and which the authority is keen to prevent not just simply to cut cost but also to avoid catastrophic failures in a patient's treatment. Bucci *et al.* [Bucci2015] report on a smart health application employing mobile technology to deliver a cognitive behaviour therapy-informed intervention in early psychosis and to detect potential relapses of patients within the community. Cognitive behaviour therapy (CBT) is recommended for the treatment of psychosis; however, only a small proportion of service users have access to this intervention. The Manchester University School of Health Science is prototyping smart-phone based apps to deliver CBT to patients within the community. Using the apps, patients complete simple form-based self-reports several times a day while one of the apps passively monitors the patients' geolocations as proxies to social activities, e.g. visiting a Day Centre, cinema etc. The reports and geolocations are analysed in the cloud together with the patients' stored longitudinal data to evaluate treatment progress and, critically, to provide early warnings of relapses. If the background analyses reveal that a patient's condition is deteriorating, the system will issue an invitation for the patient to attend clinic and to alert the NHS care contact. If a patient is considered to be on the verge of a relapse, the system will trigger a pre-defined emergency procedure to mitigate risks to the patient and the community.

The second use case follows a similar, but more pro-active approach to customising patient-specific treatment of age-related gait instability and actively preventing falls in frail older people in the community. Two of the greatest challenges to healthcare worldwide, affecting both developing and advanced economies are the increasing number of falls which have accompanied changing age structures for national population and poor patient compliance with prescribed treatment regimes. Statistically, every person over 65 will have two falls each year; with each fall will sharply increase further risk in falling and the fear of falling. Falls alone cost the UK NHS £2.3 billion each year. Falls account for 50% of NHS accidental injury admissions for over-65's; primarily from ensuing hip fractures. 50% of hip fracture patients never recover full function and 20% die within 3 months. If frail older people are admitted to a care home, environmental factors further increase fall risk, and the lack of exercise and stimulation are predictive of dementia. Care in the community is therefore vital, and training or medical devices which assist older people stay independent and active will reduce lifelong care costs and increase wellbeing into old age.

To address these issues, Hunt *el al.* [Hunt2016] propose a smart IoT cloud-based system using ultra-smart wearable textile [Lin2011] knee braces with in-built micro- and flexible-sensors to continuously assess, prescribe and monitor compliance and progress of stratified (patient-specific) treatment of gait instability. The wearable device continuously streams the wearer's gait data which is analysed via machine-learning algorithms for longitudinal gait (locomotor activity) pattern and characteristics of instability that are precursor of an imminent fall [Preece2009]. The longitudinal analysis will form the basis of a low-cost remote diagnosis, prescription of stratified treatment and monitoring of compliance while detection of the latter to real-time interventions through triggering a change in the stiffness of the ultra-smart textiles and/or electro-stimulation of muscles to prevent the fall. It has been established that there is a 150 milliseconds time-window in which a likely fall can be prevented. Consequently, low latency computation is a critical requirement to achieve this vision of active intervention.

In summary, the trend in scientific research is moving beyond passively analysing data in the background; it is also about actively processing dynamic data, often remotely making use of distributed cloud-based resources. In the area of facility-based Big Science research, timely analysis of in-experiment raw data facilitates experiment steering to help deliver better quality results. In other research areas, the use of a broad spectrum of data from diverse sources, e.g. social, longitudinal and other archived data, in big data type analyses to extract intelligence on the fly to feedback into the data producing event to help mitigate risks in unfolding situations.

2.4. Security trends

The IoT comprises a multitude of concepts ranging from small devices to the powerful cloud, and the bridging gateways and networks in between. Every IoT system usually integrates large numbers of various interconnected devices that either communicate with each other or directly with the application end. Whether we consider cloud computing, edge computing or fog computing, the size of the attack surface is very big and several unique security challenges arise.

Analyzing some of the more recent assessments and reviews (e.g., [Vasilomanolakis2015], [Cam-Winget2016], [Bertino2016], [Hwang2015], and [Medwed2016]), we can summarize the most important requirements that should be addressed as follows:

- **Trust**: IoT systems are usually deployed in highly uncontrolled and untrustworthy environments, many times without any supervision. Therefore, security should be considered by design, and most importantly, the level of security should be continuously monitored and managed during runtime.
- **Constrained resources:** The majority of IoT devices have limited resources, thus security solutions should be supporting IoT systems that rely on constrained resources, low power, and low cost.
- **Scalability**: The increasing number and diversity of devices in IoT ecosystems require highly scalable and efficient security solutions.

In the recent years, different approaches, trust models, and security architectures have been proposed that enforce and/or monitor various security features and thus contribute to achieving these goals.

To achieve trustworthiness in IoT, we need to not only protect the data collected, processed, and stored by systems, but also protect the infrastructure supporting these procedures. This means that in order to ensure confidentiality, integrity, and availability of the data, we need to first ensure integrity and security of the software managing it. Only trusted devices that behave as expected (i.e., the deployed SW is not altered) can produce trustworthy data. To this end, the scientific community has put a lot of effort into developing various approaches for **software integrity verification**.

Despite of all available methodologies and technologies to protect the software controlling IoT systems, there is no such thing as perfect security. New software vulnerabilities are discovered on the daily basis, and different cyber-attacks occur just as fast. Some of them are insignificant and unnoticeable, but some of them are substantial and cause serious damage. Thus it is important to also consider situations where an IoT system carries vulnerabilities and is thus exposed to potential attackers, or worse, when it is already being attacked. To this end, the researchers have developed methodologies for **software vulnerability management**.

As in any other ICT system, also in the IoT, the efficiency of controls and mechanisms preventing or dealing with attacks relies on appropriate **identity and access management** and overall **secure communication**.

Some of these security techniques and technologies are complex and expensive; however, there are some scalable and efficient solutions that are suitable even for low-end devices. Below we present the ones that could be the most relevant for mF2C.

2.4.1. Software Integrity Verification

Integrity verification of the device's firmware/software (i.e., device attestation) is based on the following challenge-response protocol:

1. To prevent replay attacks, the *verifier* first sends a random nonce to the *prover*.
2. The prover computes a checksum over its entire memory and returns it to the verifier (data and unused memory is erased with a predictable value, memory is read in a pseudo-random traversal to prevent checksum precomputation, all interrupts are disabled during checksum computation, the device is reset after the checksum is returned).
3. The verifier checks the correctness of the result (verifier has a copy of the expected prover's memory content and compares the received value with its own, also checks that the checksum computation time was within expected/fixed bounds).

A variety of attestation schemes exist, and they are either software-based, hardware-based, or integrate both approaches.

Software-based attestation schemes (e.g., [Kennell2003], [Seshadri2011], [Li2011]) do not rely on cryptographic secrets stored in the secure hardware, which makes them particularly suitable for devices with constrained resources. However, these techniques can be uncertain and are prone to attacks (as demonstrated in [Castelluccia2009], [Kovah2012], and [Wurster2005]) due to the strong assumptions that are hard to achieve in reality, for example, that the attestation algorithm and its implementation are optimal and that the adversary is passive during the entire execution of the attestation protocol. Moreover, software-based attestation techniques work only if the verifier communicates directly to the prover.

As opposed to software-based algorithms, **hardware-based** attestation schemes improve the level of security, but are more suitable for general-purpose computing platforms and are often too complex and too expensive for IoT devices. An early approach for hardware-based attestations is **secure boot** [Arbaugh1997], which is a security standard that prevents unauthorized modifications of code and relies on a public key (or more public keys) stored in secure hardware storage which cannot be changed by a remote attack. Using cryptographic signatures over BIOS, bootloader, kernel, and other low-level components, each component is validated before it is executed. In particular, when a machine/device starts, the firmware checks the signature of each piece of boot software, including firmware drivers and the operating system. If the signatures are good, the machine/device boots, and the firmware gives control to the operating system. Signatures provide integrity and authenticity, but they do not provide hardware anchored attestations to a centralized monitoring component (which also means there are no scalability issues). While secure boot ensures that the low-level components have not been modified by an attacker, we need to be able to detect and prevent integrity attacks for other components as well. Moreover, secure boot does not detect changes in run-time memory.

Trusted boot is a module ([Pearson2005], [Kil2009], [Datta2009]) that uses hardware technology to perform a verified launch of an operating system kernel. It checks the integrity of every component of the start-up process before loading it into the operating system. The integral parts of the trusted boot (also known as the static root of trust) are the Core Root of Trust Measurement (CRTM) component, the Trusted Platform Module (TPM) chip [TPM], and the Platform Configuration Register (PCR). The CRTM calculates the hash of the initial boot code and commits the measurement to the TPM, which is a specialized computer chip that stores artifacts (e.g., encryption keys) specific to the host system for hardware authentication. The TPM then stores the hash in a PCR, which is a special register that cannot be set, but only extended with another measurement. A TPM usually has 23 PCRs, which are reset to zero during boot and can only be extended after that. TPM can then attest the measurement to a third-party (acting as a monitoring component that can detect unauthorized modifications on the servers in the infrastructure) by signing the measurement with a key which resides in TPM and cannot be extracted from there. The detection component verifies the signature and checks if the measurement is in a whitelist. In each attestation, a freshness challenge is included to prevent replay attacks. The measurements can be taken of all components, but this introduces scalability problems. Scalability problems related to signing and verifying the attestations are addressed in [Schiffman2012], [Schiffman2010], and [Schiffman2011], whereas scalability problems related to collecting and maintaining a whitelist are addressed in [Berger2015]. Another limitation of the trusted boot model is that it does not detect changes in run-time memory.

One extension of the trusted boot is the so-called **Integrity Measurement Architecture (IMA)**, which is a subsystem for detection of files that have been accidentally or maliciously modified. IMA intercepts all attempts to access files and appraises (*appraisal* is a measurement which is locally evaluated for validity, an example of measurement is file signature) a file's measurement against a good value. IMA can be integrated with a TPM chip to provide hardware anchored attestations to the central monitoring component and thus extend trusted boot to all files. VPN software strongSwan [Stefan2012] uses remote attestations based on TPM certified IMA measurements.

Scalable attestations [Berger2015] provide software integrity of the hypervisor's critical components by combining secure boot, trusted boot, Linux IMA, and TPM.

While trusted boot, secure boot, and scalable attestations provide software integrity, there is a complementary concept of **Trusted Execution Environment (TEE)** which aims to ensure that sensitive data (like keys and passwords) cannot be extracted from the application by the software components running on the same system, in particular by privileged code such as kernel or hypervisor code. Examples of TEE are SecureBlue++ [Williams2011], Intel SGX [McKeen2013], Secure Processor [Lee2005], ARM TrustZone [ARM], Kinibi [KINIBI].

Finally, the so-called **Dynamic Root of Trust (DRT)** enables attestation protocol after boot. The goal of the DRT is to create a trusted environment from an untrusted state, i.e. to create a clean state and provide a report for a piece of code someone wants to execute. This is accomplished by allowing the CPU and the chipset to reset the state of some PCRs, disable all but one CPU and blocking/stopping everything currently running, isolate a memory region, then execute the attestation protocol again. The DRT has been implemented by major vendors, for example, Intel [Intel2012] and AMD [AMD2005], and it has been used in many security architectures for various platforms ranging from web servers to embedded devices (e.g., [McCune2010], [Nie2007], [Parno2010]).

More recently, the focus has shifted towards software based techniques that at least partially exploit secure hardware by minimizing required hardware security features - so called **hybrid approaches** (e.g., [ElDefrawy2012], [Koeberl2014], [Brasser2015], [Asokan2015], [Francillon2014], [Ibrahim2016]). Therefore, they tend to offer better security than software only methods, though commensurate with limited security functionality compared to hardware-based techniques.

2.4.2. Software Vulnerability Management

Despite of the myriad of tools and technologies that enable software integrity verification and thus protect against code manipulation, software itself can have security vulnerabilities. And since internet connected devices are constantly subject to external probing and attacks, these vulnerabilities can quickly and easily be exploited. Moreover, manufacturers of IoT devices usually integrate chips into their products that are cheap, do as little engineering as possible, and put them on the market. This means that even though devices are new and sophisticated, the deployed software is insecure. Hence, there is an important need for solutions that enable secure remote software updates as means of mitigating software vulnerabilities discovered after deployment.

Considering the heterogeneity of technologies used in IoT and the number of different environments in which they are deployed, securely updating software in IoT is not trivial. Apart from the fact that updates have to be (i) *efficient* since they are performed on devices with constrained resources that (usually) operate 24/7, and (ii) *robust*, *atomic*, and *fail-safe* so that devices don't become unusable, there are two fundamental security issues that need to be addressed. Namely *who* is installing *what*. The software to be installed on a device has to come from a legitimate source and the issuer has to be sure that it is installed on a legitimate device. Additionally, from the receiver's side, the software to be installed has to be the correct one, and from the other side, the issuer might want to protect the confidentiality of the software. Cryptographic techniques can be used to overcome these issues (e.g., [Adelsbach2005], [Misra2013], [Ambrosin2014]).

In terms of the update process itself, it can be done in two ways. Devices can use an **update agent**, which is a piece of code on the device that receives an update from a local storage (e.g., USB) or a remote server and applies it (e.g., [SWUpdate] or [RAUC]). The problem with update agents is that the approach does not scale.

The other option is to perform **Over The Air (OTA)** updates where software is pushed to devices from a central server and are executed (semi-)automatic (e.g., [Mender] and [Resin]). However, with OTA approaches various concerns arise, for example, authenticity (*is the update legit?*) and integrity (*is the received update the one that was sent?*) of updates.

When dealing with software updates or patching vulnerabilities, it is important to first detect vulnerable devices and isolate them from other nodes in the system to minimize the damage of their potential exploit. It may also happen, that vulnerabilities are detected on some devices in a system, but patches are not yet available. In these cases, and in cases where some devices in a system have already been attacked, systems have to be able to automatically detect vulnerable/compromised devices and enable secure coexistence with other nodes. Recently, some solutions for this so-called **dynamic network segmentation** have been proposed [Miettinen2016].

This approach can be done in advance, as a preventative measure, before any device is compromised or becomes vulnerable to security attacks . The entire IoT network can be divided into isolated segments (for example, based on location), which prevents abnormal behaviour of one segment affecting others [Oltisk2014].

In any case, whether segmentation or device isolation is managed before or after deployment, the security and usability of this concept relies on identity and access management and secure communication, discussed below.

2.4.3. Identity and Access Management and Secure Communication

Authentication and access control are the most critical functionalities in the IoT systems for enabling controlled and secure communication between devices. In the IoT, every device is communicable and accessible through the Internet and needs to be uniquely identified.

Several approaches exist that provide with authentication and access management suitable for IoT environments/infrastructure. These range from a model that protects the IoT from man-in-the-middle, replay, and denial of service attacks [Mahalle2013], a lightweight multicast authentication mechanism for small scale IoT applications[Yao2013], authentication protocol based on zero-knowledge proofs suitable for low-end devices [Flood2014], to a framework that addresses scenarios where transactions linked to the same identifier must not be traceable [Alpar2016].

All services that are published externally need to be properly secured. Transmissions need to be protected from eavesdropping (confidentiality) and interference (integrity). Confidentiality and integrity are most commonly achieved by **Transport Layer Security (TLS).** There are other approaches, like encryption of data by applications, which does not protect data only in transit, but also at rest, however this requires key management techniques if data can be shared.

To establish a secure communication between internal services, which might reside on different devices, machines or even in different data centers, an internally-hosted **Certificate Authority (CA)** can be used for issuing TLS certificates. CloudFlare SSL (CFSSL) [Sullivan2015] is an open source PKI/TLS toolkit which can be used as a CA.

Recently, there were attempts to enable TLS for protocols that are used in IoT, like Datagram Transport Layer Security (DTLS) (presented in [McGrew2010] and [Keoh2014]), and the Logical Link Control protocol (LLCP) secured by TLS (LLCPS) [Urien2013]. DTLS aims to achieve stream-oriented TLS, while LLCPS provides multiplexed communications between two Near Field Communications (NFC) Forum Devices.

2.5. Resource management and QoS

This section provides an overview of architectures and tools proposed in order to provide Resource management and QoS assurance in Fog, as well as, Fog to Cloud in literature.

While still the concept of a Fog resource is under discussion, a reference work for this project in [Marin2016] defines a Fog Node (resource) as "a system that can, on the one hand control a specific set of edge devices, while on the other, access to clouds". This definition has a clear impact on mechanisms defined for Fog resource management in mF2C. In addition to this, as highlighted in [Vaquero2014], resource management in Fog Environments significantly differs to resource management in Cloud environments, requiring potential distributed and hierarchical structure due partial control over and volatility of underlying fog resources.

Fog computing was introduced in [Bonomi2012]. This initial work was later evolved into a Fog architecture in [Bonomi2014], which considers new requirements that IoT scenarios pose on Fog Computing with regards to big data analytics. Overall the approach is based on the fact that IoT platforms will be, in the short term generating large volumes of data, thus requiring analytics platforms to be geo-distributed in a way that processing is moved near the data". Therefore, creating the need for "distributed intelligent platform at the Edge (Fog Computing) that manages distributed compute, networking and storage resources". This work has proposed a high level architecture which deals with the following key objectives: Fog architecture has to enable transparent resource management considering heterogeneity in fog nodes and environments that range from core (cloud), edge (fog), access networks and endpoints; Heterogeneity has also to be supported at level of applications considering the diversity of vertical sectors and applications that can make use of the platform; Fog platform has to support distributed policy based orchestration, enabling scalable management of individual subsystems and for the overall service. In order to manage Fog Resource heterogeneity the proposed architecture considers a Fog Abstraction Layer which hides the heterogeneity of resources that compose the Fog set-up while providing a unified interact for seamless resource management, metering and control. The types of physical resources of Fog devices mentioned are CPU, memory and energy which aims to enable multi-tenancy though resource virtualization mechanism. QoS management is expected to be covered at Fog Service Orchestration layer. This layer aims to provide dynamic and policy based orchestration across the Fog infrastructure and services through a Policy based Orchestration framework. Within this framework administrators can define multiple policies that determine system behaviour with regards to QoS. Considered parameters in this environment are network, storage and compute linked to parameters such as minimum delay or maximum rate.

Another interesting work is ANGELS. ANGELS stands for "Available Network Gateways in Edge Locations for Sensors" and it is presented in [Mukherjee2014]. ANGELS presents on-going work and explores the idea of using smart edge devices - sensor gateways, personal laptops, play-stations, smartphones in order to perform parallel execution of data processing jobs in IoT, using idle capability of these devices, in an approach similar to Grid's volunteer computing. Resource capabilities at Edge / fog devices area provided through a capacity metric in the device, which is a linear function CPU, memory and communication bandwidth. Analysis jobs in the architecture are scheduled through HTCondor framework, an evolution of Grid's Condor scheduler adapted for process in edge devices. So far this architecture is working under the assumption that edge devices are "always" available. Next steps detail the consideration of dynamic availability patterns of edge devices.

[Cardellini2015] [Cardellini2016] proposed a distributed and self-adaptive QoS aware scheduler for Apache Storm Data Stream processing framework. These extensions allow exploiting local resources at Fog Computing infrastructures. In the proposed architecture the Storm's baseline has been enriched with an adaptive scheduler and a QoS monitor, executing at each Storm Supervisor, in charge of a number of Fog

Nodes; and WorkerMonitor, placed at each worker Fog node.[S1] While Worker monitor collects incoming and outgoing data rate focusing on network information, QoSMonitor centralises all available information structuring it in intra-node information (utilisation and availability) and inter-node information (network). Adaptive Scheduler performs a MAPE self-adaptation process based on optimisation algorithm for latency, availability and resource utilization.

[Skarlat2016] provides a systematic classification of fog resources. This classification comprises the following classes for resources: fog cells, single IoT devices that control a series of other IoT resources while offering virtualised resources; fog colonies, defined as micro-data centers build-up of a series of fog cells. In the proposed architecture a fog orchestration control node is available at each fog colony in order centralise and orchestrate available fog cells. Resource provision in this context is formulated as an optimisation problem that aims to optimise resource utilisation of fog cells and minimisation of delays. Evaluation of the proposed model has been performed thought extension of CloudSim simulation framework for Fog Computing, resulting in 39% delays reduction. Next steps will consider the implementation of the depicted architecture and its evaluation in real case scenarios.

MIST [Arkian2017] presents Fog-based data analytics scheme with cost-efficient resource provisioning applicable in IoT crowd-sensing applications. In the proposed architecture, the following typologies of resources, such as high-end servers, edge routers, access points, set-top boxes, and end devices like vehicles, sensors, mobile phones, etc. are considered as fog resources. These present different characteristics at storage, compute and software (OS and applications) levels. At network level also heterogeneity is widely present; from both high-speed links connecting enterprise data centres and core to diverse wireless access technologies (e.g. 3G, 4G, WiFi, etc.) at the edge. The QoS model proposed by this work aims to improve resource utilisation of limited fog resources by providing an optimisation approach. The optimisation presented defines QoS in terms of overall expected delay for an application which takes into account Upload, Transferring among fog resources and processing delay. Evaluation of the proposed model has been performed using real data from deployed Smart cities environment.

Finally [Masip2016] has defined a set of characteristics for QoS guarantee specifically applicable in the context of mF2C project. These extend current state of the art in the following aspects: Resource availability guarantees; minimisation of failures therefore increasing reliability of fog nodes; as well as minimisation of response time, not only considering latency optimisation (as present in previously evaluated research works) but considering the overall picture of fog resources computing and storage characteristic and application specific needs.

2.6. Convergence of AI and computing

Fog computing has brought a paradigm shift in the way centralized cloud-based management system works in today’s world. Although the fog computing can provide the same services as a cloud computing, it is closer to the access devices, giving rise to an increment on security to the sensitive data and enhanced efficiency for data handling [Masip2016]. However, the Fog-to-Cloud system—as a hybrid architecture consisting of a centralized cloud datacentre and distributed edge devices—also pose a large number of technical challenges like management, coordination, smart computing and processing, and security and privacy.

To address these challenges, the F2C management system must be intelligent enough to take decision on its own. Additionally, its constituents: cloud, fog and networking parts are best envisioned as self-managed, i.e., self-configured, self-optimized, self-healed, and self-protected [Kephart2003]. For instance, installing/updating software in a large cloud data centers is currently a nightmare, and error-prone. A self-managed (autonomous) and intelligent F2C system, on the other hand, can reconfigure, re-optimize the tunable parameters in the cloud and fog to achieve better system performance and serve more customers without compromising their quality of service. Moreover, an autonomous F2C system will be able to detect, diagnose and fix the cause of failures in the system and networks, as well as protect systems from malicious attacks.

The artificial intelligence techniques, including machine learning (ML), genetic algorithms (GA), fuzzy neural network, Markov decision process-based hidden models, can help achieve the F2C system to be intelligent as well as autonomous [Wang2015]. As many researchers have noted the importance of the AI in next generation cloud computing, many companies like Google, Amazon, Microsoft and IBM have incorporated the AI in their platform-as-a-service or software-as-a-service solutions. As the traditional ML libraries do not support well processing of huge datasets, parallel computing frameworks, such as MapReduce and Dryad, and distributed implementation of ML algorithms have gained momentum in the academia and industry, which is also a relevant trend for our F2C management framework that leverages parallel programming and distributed resources allocation [Pop2016]. Authors in [Aazam2014]propose a Smart Gateway, in the context of fog computing, that collects data from fog devices, filters and analyzes it, and transmits just relevant data. Genetic algorithms have been adopted to solve multi-objective optimization problems in the heterogeneous mobile networks. [Abdelkhalek2011] proposes self-optimization of antenna tilt and power using Fuzzy neural network optimization method proposes. Although Markov Models [Bengio1999] are not exactly AI techniques, they still provide statistical solutions for heterogeneous networks with certain ability of automatic optimization. As the resource sharing among cloud and fog is dynamic and cooperative in F2C system, the above mentioned AI technologies could be utilized in the three different parts of the F2C system. For example, traditional ML in the centralized cloud datacenters, distributed ML algorithms in the fog layer, and genetic or Markov models to optimize the network resources could make F2C system an AI-enabled and autonomous, such that the system would be able to do all the functions by itself, from the configuration of it until the recovery in case of failures. Therefore, the success and efficiency of an intelligent and autonomous F2C system lies in the selection of the best possible combination of AI techniques that can effectively be adopted in different parts of the F2C system to take an intelligent decision based on the processed data near the end users, providing low latency as well security as required by critical medical and many commercial applications.

As it is self-evident that the AI has the capability to make F2C management system as self-managed, however, there are still many open questions that we need to answer. For example, which AI technology be used to analyse and extract relevant information from a large set of data, collected from thousands of mobile devices connected to the fog devices, to make better decisions, as well as optimize the use of networks and achieve a self-adaptive approach in our F2C management system? How the intelligent fog devices set different parameters without compromising the quality of service of end users as well business interest of cloud operators? How much data should fog devices transfer to the cloud, and how does cloud influence the decision making of fog devices? There are other areas related to the resource sharing and service execution in F2C architecture where the AI will play a fundamental role using algorithms based on machine-learning, genetic algorithms, etc. to avoid mistakes that human-controlled system do. This approach will offer considerable improvements on the architecture of the F2C network, making the communication between devices faster and an easier development of applications and services.

2.7. Key Takeaways

- There are still unresolved issues regarding Service Management in cloud computing, e.g., energy consumption, VM migration, multi-tenancy and interoperability.

- Regarding Service Management in Fog Computing, the main unresolved issue is Service Orchestration: how to divide and allocate services, especially in a highly mobile scenario
- Current research in IoT management relates to proposed protocols only
- IoT management proposals are not addressing the IoT management from a whole perspective: network, data, etc.
- Current research addressing data and service management is only based on semantics and leverage ontologies
- There is a need to name and address all devices in a mF2C system: this will be a key prerequisite for the development of services in the mF2C architecture
    - List of different proposals for naming and addressing
    - Pros and Cons of the different proposals: high overhead, some of them supporting/non-supporting mobility, other focussed on sensors, etc.
    - NO unique naming scheme means F2C must adopt one of them or remain agnostic
- Trends coming from HPC are focused on accelerating access to data by means of new storage technologies, such as non-volatile memories, and also by means of new architectures that bring storage and computation closer in the data centre.
- NoSQL databases are being increasingly used in HPC scientific applications.
- Multiple classes of Big Data systems are emerging for different purposes (streaming, machine learning, data analytics and OLAP …). Research efforts are devoted to the development of a single cloud platform that enables the integration of different types of data and applications.
- In scientific research, advances in computing and technology is driving the exponential growth of data as well as enabling increasingly complex datasets to be generated and correlated in Big Data type of analyses.
- Research outside of the large facilities-based domain commonly employs a wide spectrum of third party data, e.g. social media, locational, to enhance veracity of observations.
- A trend in scientific research focuses on the timely analysis of dynamic data to extract intelligence on the fly to facilitate run-time responses to live events.
- Non-matured technologies and wide integration field present a huge attack surface area.
- Current surveys and reviews summarise three requirements to be considered :
    - Trust: how to obtain trust in highly uncontrolled and untrustworthy environment.
    - Constrained resources: privacy and security may be constrained in the same way that devices are capable of - from the perspective of power, simplicity, low cost).
    - Scalability: trust, privacy and security solution must be scalable.
- Software integrity is verified by software/hardware/hybrid approaches
- The vulnerable devices should be isolated.
- Authentication and access management are critical functionalities.
- All public or externally published services should use secure communications.
- Resource management with QoS assurance in fog should have a different approach than the cloud due to high distribution of resources, possible volatility of resources, heterogeneity of resources
- Review of works guaranteeing QoS in data analytics/big data processes by means of exploiting local resources (fog/edge) close to the devices producing the data
    - optimizing and performing a trade-off between the limited computational capacity of edge devices and their closeness and thus their low delay

3. Technology trends

3.1. Tools, platforms, IoT

3.1.1. Cloud management tools and platforms

Plenty of Cloud management solutions are available in the market. While some tools can enable the management of several features, such as storage, compute, machine instances, and even containers, other tools may focus on specific cloud functionalities [IEEEStan2017] [SelectHub] [Whatmatrix]. However, for each project, the optimal solution requires a full evaluation of the features provided by each Cloud management tool. In this section, we describe some important characteristics for a few available solutions.

RightScale [Rightscale] aims at bringing simplicity to business cloud operations, driving visibility with detailed reporting and history tracking. It allows configuration management, automated provisioning and scaling, monitoring, security, and reporting across public, private, and hybrid clouds environments. Workload liberation ensures users can manage heterogeneous applications and services in a rapidly changing market and the return of investment (ROI) calculator determines cost benefits and allows users to make informed business decisions. It offers support to Docker containers and several public clouds, including AWS, Google Cloud, IBM Softlayer, Microsoft Azure, and Rackspace.

Red Hat CloudForms [RedHat] is the downstream product of the open source project so-called ManageIQ [ManageIQ]. It gives choice and flexibility providing control of the virtualization environment, and allowing building and managing either private or hybrid clouds. As users' needs change, CloudForms evolves, preserving user investments and providing a continuum of capabilities as infrastructure progresses from traditional virtualization toward IaaS model. This platform offers support for VMware Vsphere, Microsoft Hyper-V, OpenStack and oVirt, and applications can be redeployed on distinct clouds, although it does not provide native cloud-to-cloud migration.

Embotics vCommander [Embotics] enables IaaS automation by seamlessly integrating private, public and hybrid clouds. Moreover, it supports cloud-to-cloud migration through the integration with 3rd party cloud migrations tools, and load balancers by using cloud formations in Amazon Web Services (AWS). It is an easy-to-use solution with low complexity in installation and configuration, combining automation and resource management in a single solution.

VMware vRealize suite [VMware] is a platform designed for hybrid cloud providing management for IT services on vSphere and other hypervisors such as Hyper-V 2008R2 SP1, Hyper-V 2012 and SCVMM 2012, as well as physical infrastructure and external clouds, providing a unified management by implementing a single and extensible management platform. Load balancers and SDN are supported through VMware NSX.

Oracle Enterprise Manager Cloud Control [Oracle] is an integrated business-driven enterprise cloud management solution which leverages the built-in management capabilities of the Oracle stack for traditional and cloud environments. Manager for Oracle Cloud Platform enables the user to employ his/her premises, a private cloud and Oracle Cloud in order to build, deploy, and operate application environments. The Management Pack for Oracle Database includes support for managing database cloud services for extreme database consolidation, whilst the capability necessary for deploying and managing middleware-centric PaaS clouds within the enterprise is offered by Management Pack for Middleware. Other benefits provided by this solution include live changes to VM resources with policy driven Scale up, scale down, live migration and power management.

Besides the mentioned proprietary cloud management platforms, some open source cloud management tools also are worth considering. Each one of them was developed with distinct focus and, therefore, there is no single solution meeting cloud management requirements for all users [Techbeacon].

Walmart OneOps [OneOps] is a solution developed by Walmart Labs under the DevOps model. It enables developers to code their products in a hybrid, multi-cloud environment leveraging OpenStack cloud environments and enables on demand migration of applications between distinct clouds, allow the use of the best cloud hosting in terms of cost and offered features. Besides OpenStack, it also supports Rackspace, MS Azure, AWS and Google Cloud platform.

Docker's Machine, Compose, and Swarm [Docker] is a set of tools proposed for orchestration of distributed applications. Docker Machine can automate the provisioning of a Docker taking the user from "zero-to-Docker" with a simple command and enabling the management of distributed Docker hosts. Docker Swarm allows the management of Docker clusters enabling scheduling and guaranteeing cluster scaling, high availability and automatic balancing. Finally, the Docker Compose can assemble multi-container distributed apps that run on top of the clusters provided by the Docker Swarm.

Ansible [Ansible] is a simple, powerful, and agentless open source framework providing automation capabilities for containers, networks and cloud services. The management service makes use of SSH in order to manage Unix nodes and PowerShell so that both get along with Windows servers. Moreover, Ansible Playbook can provide information about the state of servers and services that use the YAML language, allowing task or application orchestration, and supports public, private, and hybrid cloud service providers. Additionally, Ansible can also be extended through plugins

3.1.2. Fog management tools and platforms

In this section the existing fog management tools are presented. Currently, even when there are projects for managing the fog infrastructure (UniServer [UniServer] for example) under development, most of the available solutions are focused in the Internet of Things, which will be reviewed in the next section.

The two major ready-to-use solutions for managing fog are Vortex Fog, [PrismTech], and AWS Greengrass, [AWS]. According to Vortex Fog [PrismTech] enables the secure forwarding of data between Fog subsystems containing edge node applications communicating with each other on a Local Area Network (LAN), and other nodes and subsystems that are connected over a Wide Area Network (WAN). Vortex Fog can be configured to ensure that only "data of interest" is forwarded to the WAN in order to optimize network bandwidth. It can also transparently manage any impedance mismatches when forwarding data from a low latency UDP multicast LAN to a TCP endpoint. Vortex Fog enables secure data sharing between subsystems by supporting encrypted communications, combined with authentication and access control at the Fog subsystem boundary.

Among other, and according to PrismaTech, the creators of Vortex, the key features of Vortex Fog are the possibility for natively support both cloud and fog computing environments providing system wide support for automatic discovery of the underlying network topology and computing platform technologies. Also, Vortex provides implementations optimized for different device platforms (sensor, embedded, desktop, server and web), each providing the low latency, secure, QoS-enabled data connectivity required by IoT systems. Where there is a requirement to manage high velocity data, typically in the fog tier at the network edge, Vortex can take advantage of network capabilities such as UDP multicast to enable efficient, low-latency and reliable Device-to-Device data sharing between fog nodes, adapting to the underlying network capabilities when necessary.

On the other hand, AWS Greengrass, [AWS], is a software developed by Amazon that allows users to run locally compute, messaging and cache data storage in a secure way. The platform facilitates the communication with other devices and permits to respond to local events when the connectivity is intermittent and even without Internet connection. This solution extends Amazon Web Services to devices in a simple way, what allow them to act locally in function of the generated data while the cloud is used for administrative tasks, analysis and long term storage. According to Amazon, the main advantages of AWS Greengrass is the ability to respond to local events in near real-time, the capacity to operate offline, the ease to program devices leveraging the integration with AWS Lambda (an Amazon service for programming in the cloud) and the filtering of data to be sent to the cloud, what helps to reduce costs of running IoT applications.

3.1.3. IoT management tools and platforms

The enormous amount of Internet connected objects and the expected growth in the next years has led to the emergence of different IoT management tools with the sole purpose of get the most out of the envisioned IoT emerging paradigm. In this section we review some of the most prominent solutions aimed to manage the deployed Internet of Things devices.

3.1.3.1. Sofia2

SOFIA2 is a middleware that allows the interoperability of multiple systems and devices, offering a semantic platform to make real world information available to smart applications (IoT). It is multi-language and multi-protocol, enabling the interconnection of heterogeneous devices. It provides publishing and subscription mechanisms, facilitating the orchestration of sensors and actuators in order to monitor and act on the environment [SOFIA].

The key features of Sofia2 are the capability to transfer data through the network, which can be Wi-Fi, radio, satellite, 3G/4G, etc. Also, the platform collects and stores the IoT network information, what allows to process, extract and send knowledge to people, IT systems or IoT devices to perform actions.

3.1.3.2. AllJoyn Framework

AllJoyn is an open source software framework that makes it easy for devices and apps to discover and communicate with each other. Developers can write applications for interoperability regardless of transport layer, manufacturer, categories, OS and without the need for Internet access or the cloud.

The AllJoyn framework handles the complexities of discovering nearby devices, creating sessions between devices, and communicating securely between those devices. It abstracts out the details of the physical transports and provides a simple-to-use API. Multiple connection session topologies are supported, including point-to-point and group sessions [AllSeen].

3.1.3.3. IoTivity

IoTivity is an open source software framework enabling seamless device-to-device connectivity to address the emerging needs of the Internet of Things. The project was created to bring together the open source community to accelerate the development of the framework and services required to connect these billions of devices connected to Internet.

The IoTivity architecture aims to (i) reuse existing and establish new common communication protocols for discovery and connectivity across multiple transports, (ii) apply common approaches for security and identity, (iii) define common profiles, object models, and developer application programming interfaces (APIs), (iv) promote device and application interoperability across markets and use cases, (v) provide opportunities for innovation and allow for differentiation, (vi) define a communication and interoperability solution across multiple product markets and (vii) connect everything from the smallest wearable to the largest smart car [IoTivity].

3.1.3.4. Thread

The Thread stack is an open standard for reliable, cost-effective, low-power, wireless D2D (device-to-device) communication. It is designed specifically for Connected Home applications where IP-based networking is desired and a variety of application layers can be used on the stack [ThreadGroup].

The general characteristics of the Thread stack are:

- Simple network installation, start up and operation. The simple protocols for forming, joining, and maintaining Thread Networks allow systems to self-configure and fix routing problems as they occur.
- Secure. Devices do not join the Thread Network unless authorized and all communications are encrypted and secure.
- Small and large networks. Home networks vary from several devices to hundreds of devices communicating seamlessly. The network layer is designed to optimize the network operation based on the expected use.
- Range. Typical devices in conjunction with mesh networking provide sufficient range to cover a normal home. Spread spectrum technology is used at the physical layer to provide good immunity to interference.
- No single point of failure. The stack is designed to provide secure and reliable operations even with the failure or loss of individual devices.
- Low Power. Host devices can typically operate for several years on AA type batteries using suitable duty cycles.

3.1.3.5. FIWARE

FIWARE is a new European cloud platform that provides a simple set of APIs (Application Programming Interfaces) named Generic Enablers (GE) that ease the development of Smart Applications in 16 vertical sectors, including smart cities, multimedia, eHealth, Internet of Things Services Enablement, data/context management, security, cloud hosting, among others [FIWARE].

The specifications of these APIs are public and royalty-free. Besides, an open source reference implementation of each of the FIWARE components is publicly available so that multiple FIWARE providers can emerge faster in the market with a low-cost proposition.

3.2. Technology trends coming from the HPC area

3.2.1. Data Management Trends

Analogously to the scientific trends in data management and HPC, technology trends can also be classified into storage technologies, architectural solutions, and software platforms.

Regarding new storage technologies, the first NVM devices such as the Intel® 3D XPoint™ NVDIMM memory are becoming available. This device can be configured in different modes, acting as a volatile memory, as a block device, or as a byte-addressable device.

The main technology trends at the infrastructure level are focused on bringing storage close to computation. Active storage is an architectural concept that addresses the increasing costs of data transport between compute and storage systems. Therefore, computing power and storage are much more tightly integrated. In particular, IBM has extended the Blue Gene/Q architecture by integrating Flash into the node to enable a scalable, data-centric computing platform in the BGAS (Blue Gene Active Storage) system [Fitch2013]. Compute-in-storage is intended to enable the use of high performance (HPC) programming techniques (Message Passing Interface, MPI) to implement data-centric algorithms (e.g. sort, join, graph) that execute on processing elements embedded within a storage system.

At the software level, traditional parallel file systems such as Lustre [Lustre] and PVFS (OrangeFS) [PVFS] are still widely used in HPC. Recently, however, solutions born in the big data field such as HDFS [Shvachko2010] are also being adopted by the HPC community. HDFS is a Java-based file system that provides scalable and reliable data storage, and it was designed to span large clusters of commodity servers. HDFS supports the concept of computing close to data by means of MapReduce [Dean2004], which allows processing huge amounts of data where it is stored.

The concept of keeping computation close to data, and also the idea of byte-addressable NVMs that enable applications to interact directly with the data without the need of going through a file system or a database interface, are brought together at the software level in the dataClay storage platform [Marti2013, Marti2017]. dataClay is a distributed object store where applications can manipulate objects as they see them in their address space, without mapping them to any particular format and without worrying about their location. In this platform, the behavior associated to the objects is also stored, in such a way that it can be executed within the platform without the need of any data movements.

Regarding NoSQL databases, key-value Stores (KVS) [Li2015] are databases that use an associative array (similar to a map or a dictionary) as the fundamental data model, where each key is associated with one value in a collection. There exist many kinds of key-value stores, the most popular ones being HBase [George2011] and Cassandra [Lakshman2010], which support MapReduce, Berkeley DB [Olson1999] and Memcached [Memcached]. Document stores are also being used in HPC and scientific environments due to their flexibility. In a document database, data is semi-structured in some XML-based language, such as YAML or JSON. The most popular technologies of this kind are MongoDB [Bradshaw2016] and CouchDB [CouchDB]. Finally, there are applications that are based on a graph data model. Graph databases [Robinson2013] such as Neo4j [Neo4j] can be used for this purpose, and frameworks such as GraphX [Gonzalez2014] are becoming popular for large scale graph processing.

3.2.2. Programming Models Trends

Apart from improving data storage, another important trend in HPC is incrementing the number of computing devices integrated within a computing node. Not just in terms of the number of CPUs and cores in chip but also by including different accelerators like GPUs and FPGA in order to speed up key parts of the application algorithms. Every device has its own APIs or libraries which allow users to program applications for these heterogeneous devices; however it increases the programming complexity of applications. To solve the problem of using different device API, the openCL [openCL] and openACC [openACC] are working on providing a standard interface for interacting with accelerators. However, these solutions do not solve the extra programming effort to deal with data movements from main memory to device as well as spawning processes in the device. The OmpSs programming model and runtime [Ompss@GPU][Ompss@FPGA] tries to transparently manage this work instead of the developers, providing a high-productive task-based programming model for integrating accelerator computation with traditional CPU parallelism. In the framework of the TANGO [Tango] project, this programming model is also combined with COMPSs [COMPSs] which follows the same task-based concepts but in distributed computing environment with the aim to provide an integrated programming model which is able to manage heterogeneity in highly distributed computing platforms.

COMPSs is also used to implement the abstraction layer of the EUBra-BIGSEA [BIGSEA] platform in order to make users able to compose big data applications without the need to know the details of the specific data analytics framework. The core of such platform is the QoS infrastructure that includes a monitoring service that closely follows the execution of the applications running in a Mesos cluster, and that implements proactive and reactive elasticity mechanisms to adapt the system in order to guarantee the QoS of the applications.

3.3. Cloud Orchestration Platforms, Virtualization, Containers

In the area of fog computing, we find different proposal, from the initial work of Bonomi et al. in [Bonomi2014] and also other works such as [Aazam2014], proposing the virtualization of computer capacity of edge devices in form of virtual machines, VMs. However, other contributions, see [Willis2014] [Ismail2015]and [ZurichUblog], propose the use of containers to run applications in fog nodes –considering fog nodes as mini-clouds at the edge of the network– due to their reduced memory capacity, computing footprint, and small size. In other research area, IoT management in [Petrolo2017] it is proposed to run services in virtualized containers deployed in the proposed IoT gateway.

During the last years we have witnessed how the cloud orchestration platforms usage have been consolidated and how the adoption of hybrid clouds solutions by companies and users has been increased [CloudTrends2016]. This is the result of the emerging technologies and the new approaches in the use of them.

Each year new commercial and open source solutions are incorporated into the set of tools that allow the orchestration and management of simple applications and complex solutions in multi-cloud environments. Some of these tools were built taken as a basis different usage perspectives and concepts; although at the end most of them share common characteristics. In addition to that, the success of container-based solutions, such as Docker [Docker], is having a great impact on all these new technologies and tools, and the way companies and users make use of them [DockerCon16] [DockerEcosystem].

3.3.1. Cloud management and orchestration tools

Cloud Orchestration is the method for managing and automating manual IT processes such as provisioning (in physical or virtual resources), installation, configuration management, monitoring, scaling, etc. in a cloud environment, with no admin intervention. Or in other words, a Cloud Orchestrator is a software entity that manages the interconnections and interaction among other cloud-based entities. In the scope of Cloud computing environments, the orchestration refers to the automation of processes and workflows required to meet the application performance goals, minimizing the associated deployment and operation costs while maximizing the application performance.

Anyways this term can be very confusing at times. It’s usual to mix concepts when talking about Cloud Orchestration and Cloud Management. In relation to this, when we talk about managing applications in a cloud environment, we can differentiate between the different approaches that one can take to manage them. First, we can take an infrastructure-centric approach (**Cloud Management Platforms** – VMWare vRealize [VMwareCMP], Right Scale [RightScale], SlipStream [SlipStream]), where the main focus is put on the monitoring and management of the infrastructure resources, like virtual machines, network, storage, memory, etc. These platforms can be used indirectly to manage applications by combining some orchestration capabilities as part of them. Then, a more developer-centric approach can be taken by using **PaaS** (Cloud Foundry [CloudFoundry], Openshift [Openshift], Heroku [Heroku]), which are built as an abstraction layer that hides all the infrastructure and operational aspects to the developers, so they can focus only on the application. And finally, we have the **Cloud Orchestration** that, basically, includes features and characteristics of the previous concepts and approaches, in order to automate manual IT processes. But at the end, all these tools and platforms are used by companies and final users to achieve the same goals: to have a fully managed application on the cloud.

There exist several open-source Cloud Management Platforms (CMP) that provide IaaS solutions, like OpenStack [OpenStack] (released in 2010), Scalr [Scalr] (2008), CloudStack [CloudStack] (2010), Eucalyptus [Eucalyptus] (2008) and OpenNebula [OpenNebula] (2008), and also some commercial CMPs like VMWare vRealize [VMWareVRealize], Morpheus [Morpheus] (released in 2014) and Right Scale [RightScale]. Some of them have been integrated in commercial cloud orchestrator solutions as part of their core features, like IBM Cloud Orchestrator [IBMCloudOrch] (released in 2014) which relies on *Openstack*. Other commercial solutions that exist on the market include Microsoft Azure [MicrosoftAzure] (2010), Flexiant Cloud Orchestrator [Flexiant], VMware vRealize Orchestrator [VMwareVOrch] (which integrates *VMware vRealize Automation*, formerly called *vCloud Automation Center*), among many others. First open sourced in 2012, the multi-cloud, generic or agnostic orchestration and management platforms, SlipStream [SlipStream] by mF2C partner SixSq, is available under open source and proprietary licenses and used in collaborations such as Helix Nebula Science Cloud [HNSciCloud] and smart city CityZen [CityZen].

Finally, with the success of containers (mainly thanks to Docker container engine) new cloud containers orchestration tools have appeared in the market, like Docker Swarm [Swarm], Marathon [Marathon], Nomad [Nomad], Amazon EC2 Container Service [EC2ContainerServ], Azure Container Service [AzureContService] or Kubernetes [Kubernetes]. Many cloud solutions are incorporating containers (and containers clusters) management as part of their offering, including PaaS providers like Openshift (*Kubernetes*), Heroku (*Kubernetes*, *Swarm*, *Marathon*) and Cloud Foundry (*Swarm*).

3.3.2. Containers orchestration

Although the idea of containers is not new and has been around since the early days of Unix (in 1979 with the *chroot* command), they are enjoying now a renewed interest within IT world in general thanks to **Docker** (released in 2013) and other cloud containers solutions. Not much time ago, when talking about cloud solutions, companies and final users could only choose between dedicated servers (bare metal) and virtualization solutions. Bare metal solutions offer a better performance and reliability, but at a much higher cost among other cons. On the other side, virtualization offers many advantages in terms of costs, scalability and many others, at the expense of a worse performance. Virtualization led to the today's successful IaaS and PaaS solutions we can find in the market. And many experts see in the cloud containers a step further in this evolution.

While a Virtual Machine makes use of a lot of system resources (including a full copy of an operating system, and a virtual copy of all the hardware that this guest operating system requires to run), all that a container needs is enough system resources (files, environment variables and libraries) to run a specific program. As they sit on top of the virtual server and its host Operating System, it is possible that multiple containers can run within a single virtual machine. In other words, instead of having a Virtual Machine for one application, we can run multiple applications in a single Virtual Machine environment. These containers are isolated from each other and from the host, and are much easier to build than VMs, which make them "fast" and "light". And because they are decoupled from the underlying infrastructure and from the host filesystem, they can be ported across different clouds and Operating Systems distributions. Next picture depicts the main difference between these two architectures:

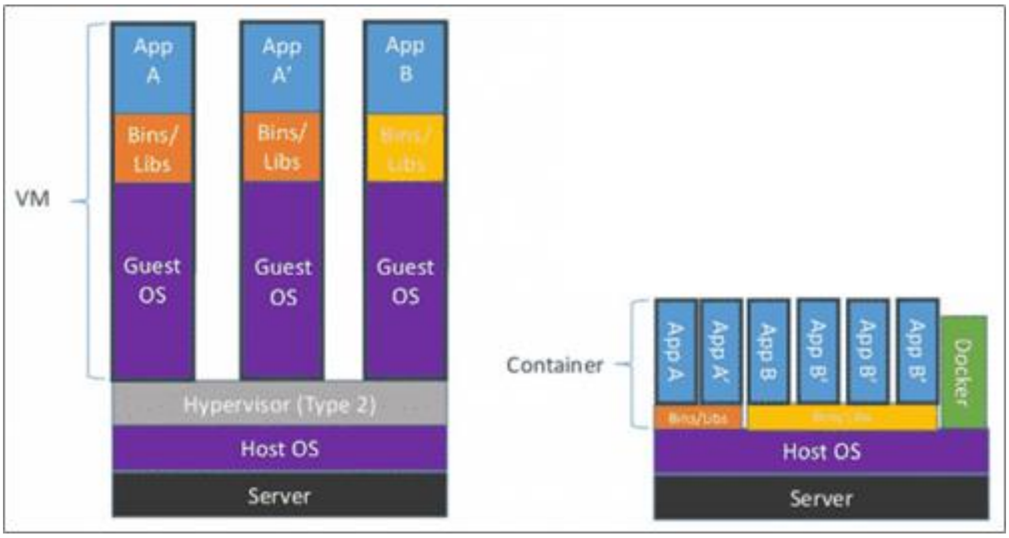


Figure 1 Containers' architectures

Other container manager technologies that have emerged during these years are the following:

- **Rocket [Rocket]**, an open source [RocketGithub] container runtime launched by CoreOS. It is supported by Kubernetes [RocketKubernetes] and Nomad [RocketNomad].

- **Singularity** [Singularity]: A container for HPC environments. It is similar to Docker, but solves some of the issues that Docker presents in HPC systems (security, mobility and parallel executions). Next image shows the main differences between Docker and Singularity containers:

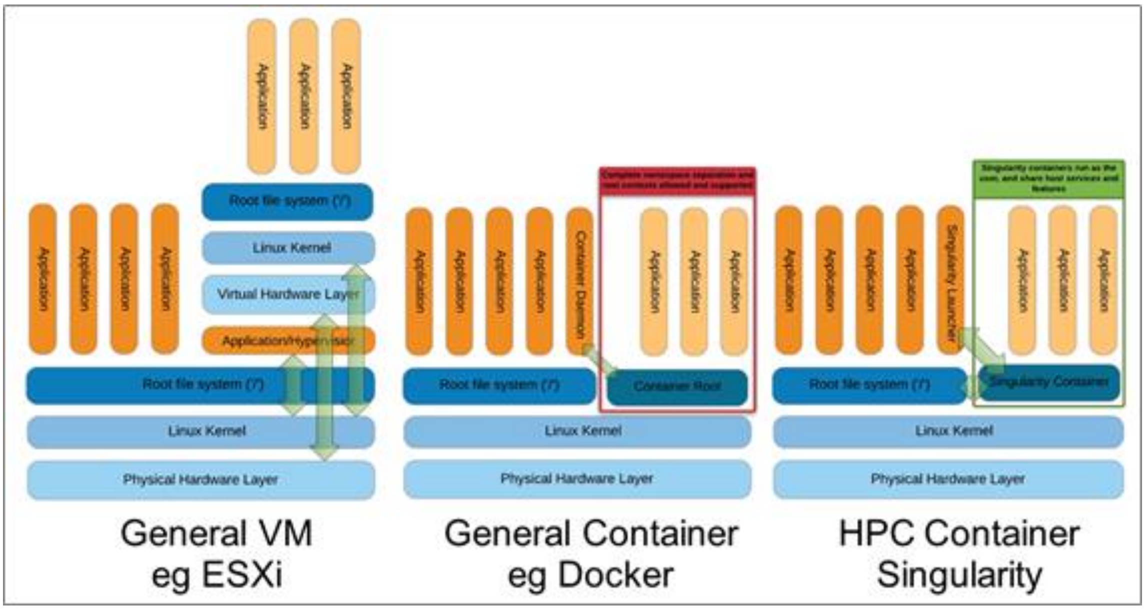


Figure 2 Docker vs. Singularity

- **runC** [RunC]: It is a CLI tool for spawning and running containers according to the OCI specification.

Containers' orchestration tools appeared to solve the problems (deployment, communication, scaling, life-cycle, etc.) derived from the management of multiple containers in a multi cloud environment.

Kubernetes [Kubernetes]

Released in 2014, Kubernetes is an open source system for Docker container management and orchestration designed by Google, which aims to provide a platform for automating deployment, scaling, and management of containerized applications in a clustered environment. These are some of the main features:

- Auto scaling capabilities; scale in or out containers on the fly
- Replication; automates the deployment and replication of containers
- Resource usage monitoring, volume management
- Easily roll out new versions of application containers
- Manages multiple containers as one entity (a pod); organise containers in groups and provide load balancing between them
- Provides container resilience, if a container dies it gets replaced immediately
- Containers in a pod run on the same host and can also communicate with each other

It works as follows: Kubernetes uses a single master server that manages multiple nodes using the command-line interface *kubectl* [kubectl CLI]. A **cluster** is a set of physical or virtual machines and other infrastructure resources used by Kubernetes to run the applications:

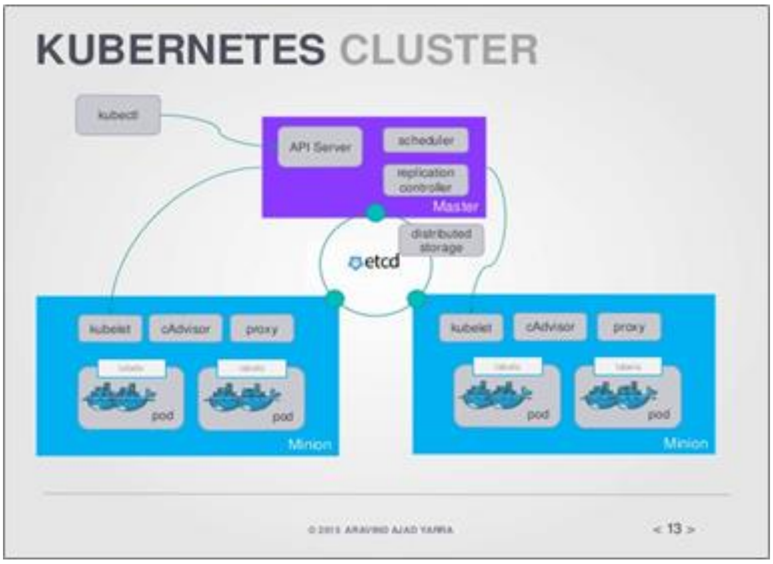

KUBERNETES CLUSTER

Figure 3 Kubernetes Cluster

A **node** is a worker machine in Kubernetes. A node may be a VM or physical machine, depending on the cluster. Each node has necessary to run pods.

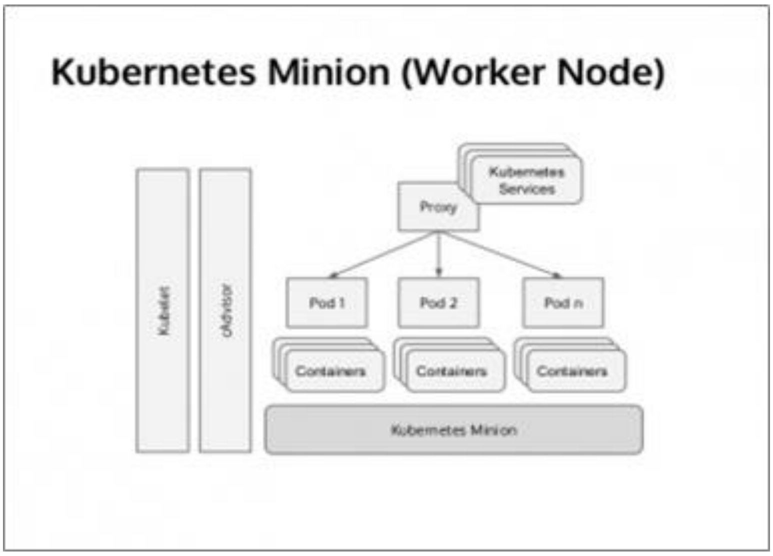


Figure 4 Kubernetes Minion

In Kubernetes, the basic unit of scheduling is a **pod**, a group of containers that are co-scheduled and deployed together on a single node in order to execute a particular task.

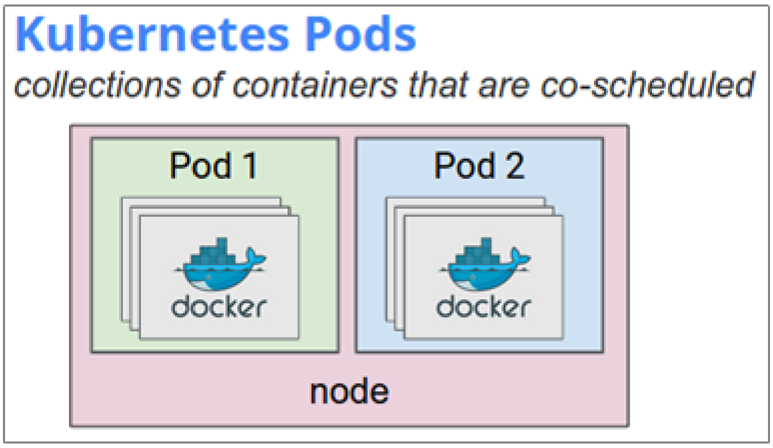


Figure 5 Kubernetes Pods

Pods are temporary – they may be generated and deleted at will while the system is running. Higher level concepts such as Deployments can be constructed as a set of pods. Users can set up custom health checks, including HTTP checks and container execution checks, on each pod in order to ensure that applications are operating correctly.

Kubernetes is supported on Google Compute Engine [GoogleCompEng], HP Helion Cloud, Rackspace, Microsoft Azure, Redhat Openshift and vSphere environments, among many others.

CoreOS Tectonic [CoreOS]
Released in 2015, it is basically a commercial Kubernetes platform that combines Kubernetes and the CoreOS stack. Tectonic is compatible with both the Docker and CoreOS Rocket containers.

Docker Swarm [Swarm]
Released in 2015, Swarm is the Docker's open source own tool for cluster management and orchestration, and it is distributed together with Docker. Swarm uses the same Docker interface enabling transparent

scalability from Docker use to Swarm use. These are some of its main features:

- Compatible with Docker tools; Cluster management integrated with Docker Engine (CLI)
- Auto scaling capabilities
- Load balancing
- Integrated networking and volumes
- Failover and high availability
- Secure by default; each node in the swarm enforces TLS mutual authentication and encryption
- Flexible container scheduling

Swarm uses a **manager** responsible for the entire cluster, which also manages the resources of multiple Docker hosts.

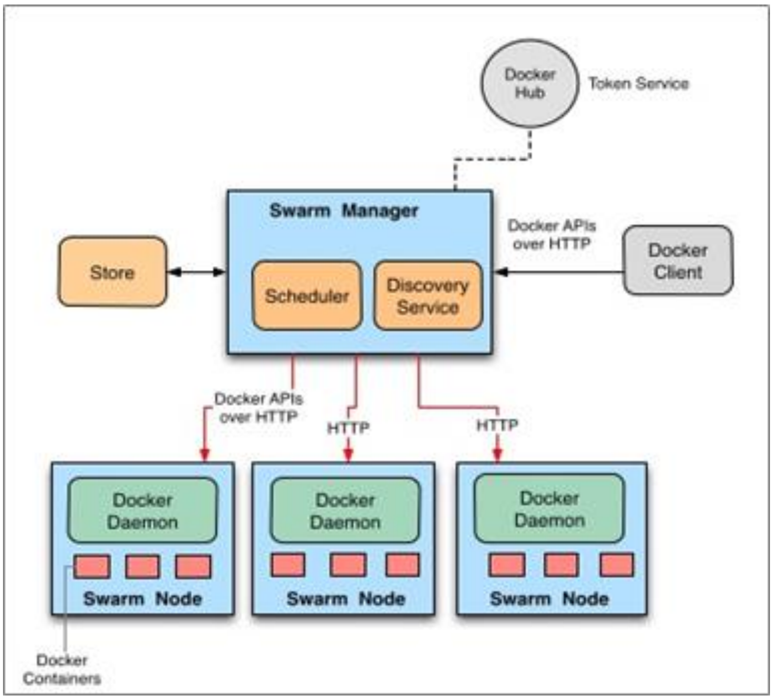


Figure 6 Swarm architecture

To deploy an application to Swarm, it is necessary to first submit a service definition to a manager node. Then, the manager dispatches units of work called tasks to worker nodes. Each node can be seen as a Docker node. You can run one or more nodes on a single physical computer or cloud server.

Compared to Kubernetes, Docker Swarm offers an easy and fast setup, and a lightweight installation. But on the other side, the API offers a limited functionality. As Kubernetes is backed by years of expert experience it's a more mature tool, although the setup and installation offer more difficulties than Swarm.

Mesosphere Marathon [Marathon]
Apache Mesos [ApacheMesos] is a cluster manager that provides efficient resource isolation and sharing across distributed applications or frameworks, which was developed at the University of California, Berkeley.

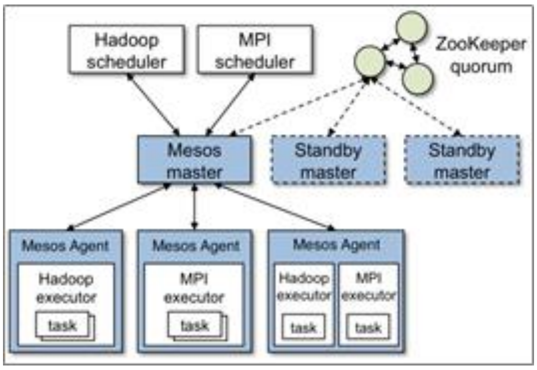


Figure 7 Apache Mesos architecture

DC/OS derives from **Mesosphere's Datacenter Operating System**, a distributed operating system based on the Apache Mesos distributed systems kernel. It enables the management of multiple machines as if they were a single computer. It automates resource management, schedules process placement, facilitates inter-process communication, and simplifies the installation and management of distributed services.

Marathon is a container orchestration platform for Mesos and DC/OS. These are the features offered by this platform:

- High Availability
- Multiple container runtimes. Marathon has first-class support for both Mesos containers (using cgroups) and Docker.
- Stateful apps
- Web UI
- Constraints. e.g. Only one instance of an application per rack, node, etc.
- Service Discovery & Load Balancing
- Health Checks
- Event Subscriptions; let you supply an HTTP endpoint to receive notifications, for example to integrate with an external load balancer
- Basic Auth and SSL
- Metrics. Query them at /metrics in JSON format or push them to systems like graphite, statsd and Datadog.
- JSON REST API for easy integration and scriptability.
- Orchestrates both apps and frameworks
- Scaling and fault recovery
- Supports Docker

Amazon EC2 Container Service [EC2ContainerServ]
Amazon EC2 Container Service is a container management service that offers support for Docker containers within AWS. Although any containers managed by Amazon ECS will be run only on instances of Amazon Web Services EC2, it offers access to AWS features such as elastic load balancing and CloudTrail, a logging and monitoring application.

Azure Container Service [AzureContService]
This is the Microsoft's container orchestration solution for its Azure cloud computing platform, and it is based on the open-source Apache Mesos cluster manager. It lets users to choose between three container

orchestration tools: Apache Mesos, Docker Swarm, and Kubernetes.

Nomad [Nomad]
Nomad is a free and open-source solution from software company HashiCorp. It is a cluster manager and scheduler. Whereas offerings like Kubernetes are intended specifically for Docker containers, Nomad is a more general-purpose solution that supports Docker as well as other applications that are virtualized, containerized, and standalone.

3.3.3. SlipStream

SlipStream, by mF2C partner SixSq, is a multi-cloud application management platform, available under both open source (community) and proprietary (enterprise) versions. The solution provides a complete environment for capturing, deploying and managing any applications, in any cloud. High-level features include a single Dashboard, AppStore, workspace for private development, unified usage monitoring and advanced auto-scaling.

Recently, the solution has been extended to manage edge computing appliances, such as the NuvlaBox also developed by SixSq, to build hybrid environments for distributed applications. The following figure illustrates how the two solutions can be interfaced.

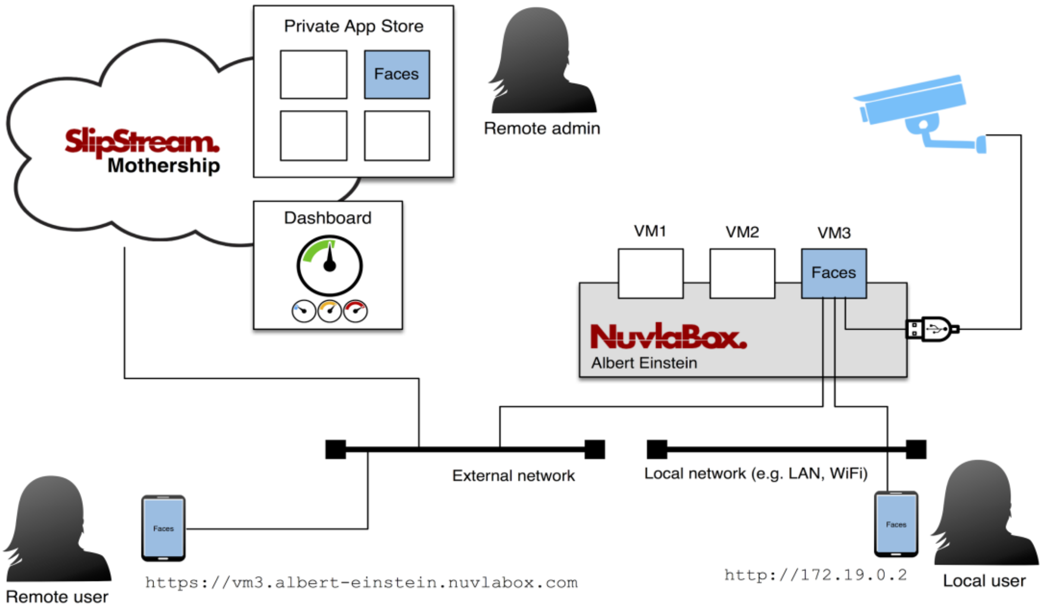


Figure 8 SlipStream platform

This example shows how cloud orchestration and management solutions can be extended to bridge standard cloud or data centre environment, with fog or edge computing environment.

3.4. Role of standards in technologies

mF2C occupies both the established world of cloud computing, and the emerging paradigm of fog computing. Cloud computing is a maturing compute model for which dedicated standards have started to emerge. Fog computing is a very recent concept and is just beginning to be considered by standardisation initiatives.

Although an immature space, there are some observations that can already be made regarding standards in cloud and fog computing. International standards with formal international recognition are defined by working groups and subcommittees of ISO/IEC JTC1 [ISO/IEC JTC1]. ISO working groups and subcommittees typically author their own standards for high level concepts. For more technical standards such as APIs they often accept and ratify standards developed in more-focused standards development organisations and industry groups such as DMTF [DMTF] and SNIA [SNIA]. Standards relevant to Cloud Computing such as OVF and WS-Management have come through this ratification process. Contributions to ISO/IETC standardisation efforts are via national standardisation bodies.

Sometimes reference is made to large commercial or open-source software implementations when discussing cloud and fog standards. When there is an absence of relevant de-jour standards, interested parties can look to large commercial or community-driven efforts (e.g. Amazon Web Services, OpenStack) as a source of de-facto standards. It should be noted however that such commercial and community-driven projects have their own goals and limited resources, and backward-compatibility, versioning and generalisations that may be necessary for holistic standards may not be something that can be facilitated or supported.

3.4.1. ISO/IEC JTC1 SC38 Cloud Computing and Distributed Platforms

ISO/IEC JTC1 SC38 [ISO/IEC JTC1 SC38] is responsible for defining and agreeing international standards in Cloud Computing. To date, standards describing concepts, terminology and reference architecture have been published. Standards addressing Service Level Agreements, Interoperability and Portability, and Data Flow have all been drafted and are very advanced in the standards preparation process with publication expected during 2017. The vast majority of SC38 standards are descriptive in nature, and are designed to allow a shared understanding of the relevant subject areas. They generally do not describe technical APIs or models or specific implementations. SC38, and its overseeing technical committee, have established a process that allows standards developed outside of SC38 to be approved by by SC38 if appropriate conditions have been met and processes have been followed. Standards from DMTF (e.g. OVF and WS-Management) have been ratified through this process.

Some aspects of individual SC38 standards that may be of particular relevance to mF2C include:

- ISO/IEC 17788:2014 Information technology -- Cloud computing -- Overview and vocabulary: Provides definitions for common terms that should be used when discussing cloud computing
- ISO/IEC CD 19086-2 Information technology -- Cloud computing -- Service level agreement (SLA) framework -- Part 2: Metric Model: includes a general model for metrics and an XML definition of an example metric.
- ISO/IEC DIS 19941 Information technology -- Cloud computing -- Interoperability and portability: describes considerations for interoperability and portability at different layers of the cloud stack.
- ISO/IEC DIS 19944 Information technology -- Cloud computing -- Cloud services and devices: data flow, data categories and data use: includes definition of a structure for statements that describe how data is used.

At the time of writing SC38 is considering what aspects of Cloud Computing should next be standardised. One possible area of focus is inter-connected multi-tenant data for global clouds, which may be of some relevance to mF2C.

3.4.2. ISO/IEC JTC1 WG10 Internet of Things

As of the time of writing Internet of Things related standards are being progressed by ISO/IEC JTC1 WG10 Internet of Things. Work is progressing on standardised IoT Reference Architecture (ISO/IEC 30141), and IoT Definitions and Vocabulary (ISO/IEC 20924). Additional sub-groups are investigating standardisation gaps, use cases, cyber-physical systems for IoT, and networking frameworks for IoT.

3.4.3. ISO/IEC JTC1 SC41 Internet of Things and Related Technologies

At the time of writing ISO/IEC JTC1 has decided to create and dedicate a sub-committee, number 41, to internet of things and related technologies. It is expected that this sub-committee will launch in summer 2017. Two initial working groups will be created within this subcommittee. One dedicated to Sensor Networks (previously JTC1/WG7), the other to Internet of Things (formally JTC1/WG10). A study group will be created to investigate the need for standards in Wearable technologies.

3.4.4. ETSI

The European Telecommunications Standards Institute [ETSI] develops global standards for Information and Communications technologies. It focuses much of its efforts on telecommunications, with some of its clusters dedicated to Interoperability, Connecting Things, Wireless Systems, and Networks. ETSI Groups that may be of particular interest to mF2C include:

- NFV - Network Function Virtualisation
- OSM - open Source NFV management and orchestration (MANO)
- SmartM2M - machine to machine communication in IoT

ETSI is also the official European standards organisation, and has been engaged by the European Commission to drive European standardisation initiatives, such as the Cloud Standards Coordination effort [ETSI-CSC] chartered with coordinating cloud standards efforts in Europe. The European Cloud Standards initiatives are influenced by the EC's Cloud-Special Interest Group [EC C-SIG]. Previous efforts have focused on Certification Schemes, Code of Conduct, and Service Level Agreements.

3.4.5. OGF OCCI

The Open Grid Forum [OGF] is an open community dedicated to developing best practices and standards for advanced, applied, distributed computing. It includes a working group dedicated to Open Cloud Computing Interfaces: OCCI [OCCI]. OCCI have developed a set of interoperability standards to enable infrastructure management tasks. Based on the OCCI Core Model, customised extensions can be defined to target functionality in particular areas. Arbitrary renderings of the interfaces can also be formalised. The OCCI standards are now at version 1.2.

Of particular relevance to mF2C are:

- [GFD.221] – Open Cloud Computing Interface – Core: describing the formal definition of the OCCI Core Model.
- [GFD.224] – Open Cloud Computing Interface – Infrastructure: defines an OCCI Infrastructure extension for the IaaS domain. The document defines additional resource types, their attributes and the actions that can be taken on each resource type
- [GFD.228] – Open Cloud Computing Interface – Service Level Agreements: defines the OCCI extension for handling Service Level Agreements for cloud infrastructure

The OCCI working group is developing an extension to support monitoring. Researchers (e.g. at University of

Pisa) are exploring how this might also be useful in an IoT scenario.

3.4.6. OpenFog Consortium

The OpenFog Consortium [OFC] was founded in November 2015 to address the technical challenges in Fog computing. They define Fog Computing as "A horizontal, system-level architecture that distributes computing, storage, control and networking functions closer to the users along a cloud-to-thing continuum". With a strong industry focus, in January 2017 the OpenFog Consortium published their OpenFog Reference Architecture Technical Paper [OFC-RA]. It describes eight OpenFog pillars: Security, Scalability, Open, Autonomy, RAS (reliability, availability, serviceability), Agility, Hierarchy and Programmability. It covers medium-to-high level considerations that address security, cognition, agility, latency and efficiency.

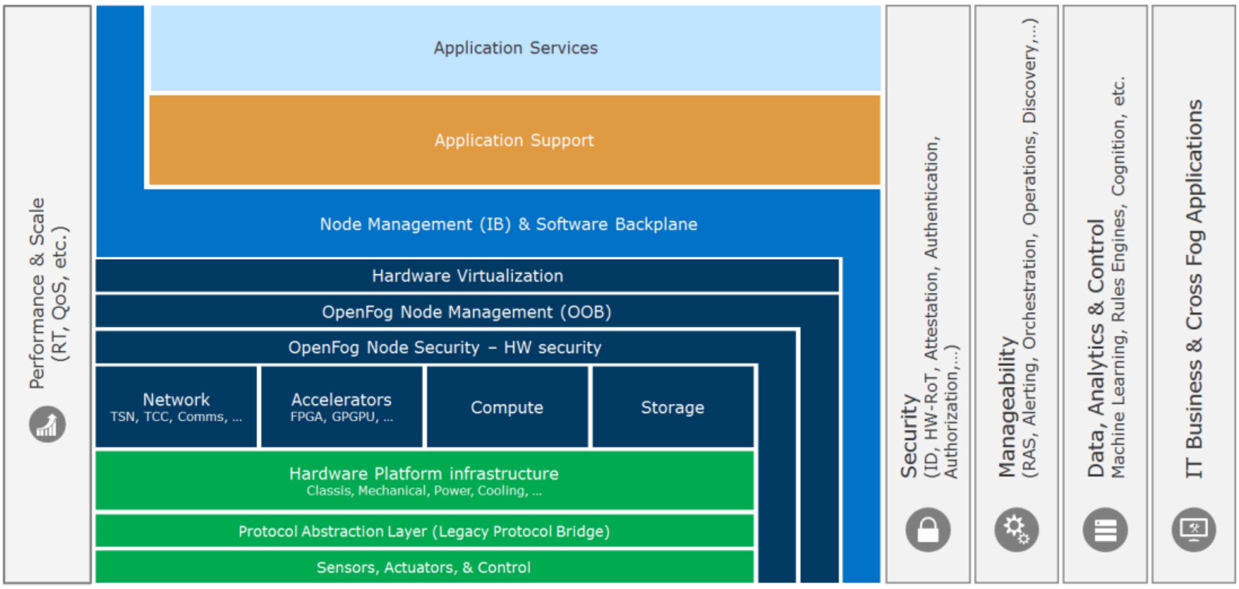

Performance & Scale
(RT, QoS, etc.)
Application Services
Application Support
Node Management (IB) & Software Backplane
Hardware Virtualization
OpenFog Node Management (OOB)
OpenFog Node Security – HW security
Network
TSN, TCC, Comms, ...
Accelerators
FPGA, GPGPU, ...
Compute
Storage
Hardware Platform infrastructure
Classis, Mechanical, Power, Cooling, ...
Protocol Abstraction Layer (Legacy Protocol Bridge)
Sensors, Actuators, & Control
Security
(ID, HW-RoT, Attestation, Authentication, Authorization,...)
Manageability
(RAS, Alerting, Orchestration, Operations, Discovery,...)
Data, Analytics & Control
Machine Learning, Rules Engines, Cognition, etc.
IT Business & Cross Fog Applications

Figure 9 The OpenFog Reference Architecture and Perspectives

The OpenFog Consortium has numerous working groups defined, these include Architecture, Communications, Manageability, and Software infrastructure.

3.4.7. Distributed Management Taskforce

The Distributed Management Task force [DMTF] is an industry group founded in 1992 dedicated to developing technologies and standards to simplify the management of devices accessible over a network. Standards it has developed include

- OVF - Open Virtualization Format: a standardised packaging format for virtualized resources
- WS-MAN - Web Services Management: a SOAP protocol for management of data centre infrastructure, services and applications
- CIMI - Cloud Infrastructure Management Interface: an API for management of cloud infrastructure (see below for details).

DMTF is currently advancing Redfish [REDFISH], a standard for addressing and interacting with all physical components in a data centre.

3.4.8. DMTF CIMI

As part of its Cloud Management Initiative, DMTF released in 2013 its first version of the Cloud Infrastructure Management Interface [CIMI]. The "Cloud Infrastructure Management Interface (CIMI) Model and RESTful HTTP-based Protocol" provides a well-structured REST framework to interact with cloud systems.

"The DMTF's Cloud Management Working Group models the management of cloud services and the operations and attributes of the cloud service lifecycle through its work on the Cloud Infrastructure Management Interface (CIMI).

The CIMI specification describes the model and protocol for management interactions between a cloud Infrastructure as a Service (IaaS) provider and the consumers of an IaaS service. The basic resources of IaaS (machines, storage, and networks) are modelled to provide consumer management access to an implementation of IaaS and facilitate portability between cloud implementations that support the specification."

This standard is one of the few neutral standards defining how to manage cloud and virtualized resources. SlipStream is currently migrating its REST API to CIMI, in order to standardise its API, but also benefit from the rigorous REST framework it promotes.

The CIMI standard v2.0 was released in August 2016.

3.4.9. Open Connectivity Foundation

The Open Connectivity Foundation [OCF] was announced in February 2016 to drive the specification of standards to enable connected devices communicate with each other. It is an amalgamation of the Open Interconnect Consortium (OIC), Universal Plug and Play (UPnP) Forum and many key industry players. Devices in scope include computers, mobile phones, sensors and the full range of IoT devices. The standards consider the device technology stack from silicon through software, platform and finished-goods.

OGF has published the OIC Specification 1.1 in February 2017 [OCF-OIC]. It includes a base resource schema, and specifications for OIC core (architecture, interfaces, protocols and services), security and smart home devices are defined.

OCF sponsors IoTivity [IoTivity], an open-source reference implementation of the standard published under the Apache 2.0 license. The base architecture for IoTivity v1.2 is illustrated below.

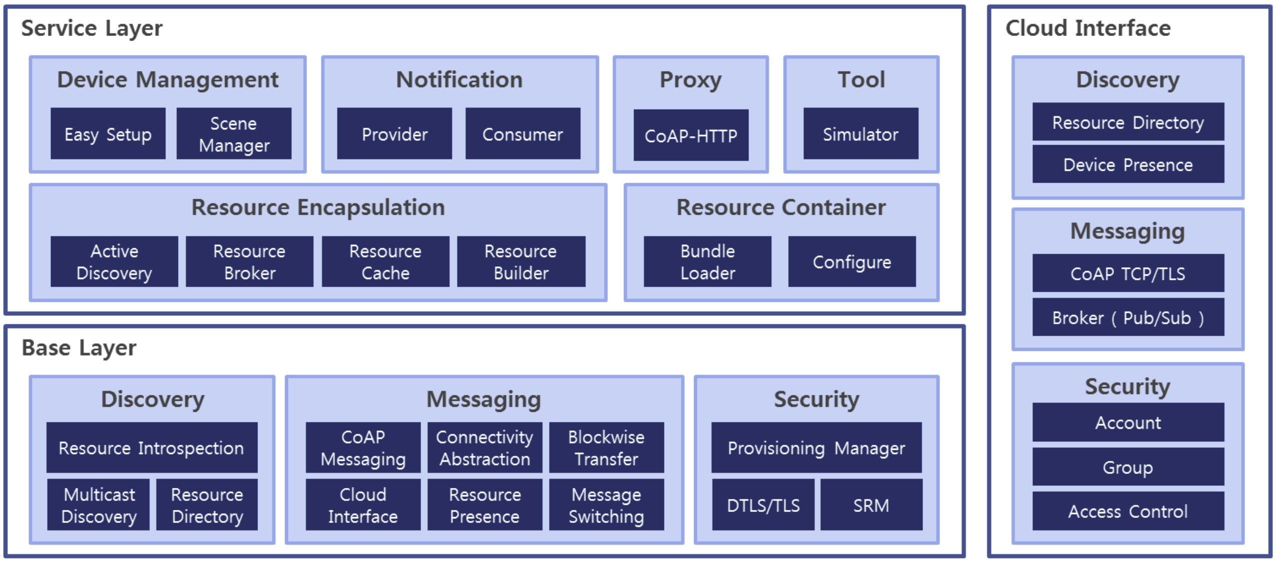


Figure 10 IoTivity Architecture v1.2

3.4.10. LoRA Alliance

The LoRA Alliance [LoRA] is driving development of a low power wide area network communications protocol, LoRaWAN, as a global standard for secure, carrier-grade, IoT connectivity. LoRaWAN is targeted at wireless battery-powered devices that may be deployed at a national scale. It includes several layers of encryption and supports bi-directional communication, multicast, mobility and localisation services. An Adaptive Data Rate (ADR) scheme is employed to maximise the battery life of all end devices, and to maximise the overall capacity of the network.

3.5. Technology trends in edge computing

In this section, we review the current trends in the field of edge computing technology. We start with a standard definition to set a reference taxonomy. We then review examples of contributions from grassroot, open source and community initiatives, as well as commercial and proprietary development. We continue with a review of companion terms to edge computing to position edge computing in a wider context, and conclude with key findings relevant to the mF2C project, its architecture and potential foundation to start from.

Wikipedia [WIKI-EDGE] proposes the following definition for 'Edge computing':

"Edge computing is pushing the frontier of computing applications, data, and services away from centralized nodes to the logical extremes of a network. It enables analytics and knowledge generation to occur at the source of the data. This approach requires leveraging resources that may not be continuously connected to a network such as laptops, smartphones, tablets and sensors. Edge Computing covers a wide range of technologies including wireless sensor networks, mobile data acquisition, mobile signature analysis, cooperative distributed peer-to-peer ad hoc networking and processing also classifiable as local cloud/fog computing and grid/mesh computing, dew computing, mobile edge computing, cloudlet, distributed data storage and retrieval, autonomic self-healing networks, remote cloud services, augmented reality, and more."

This definition is broadly in agreement with the current trends. This section provides a non-exhaustive, but representative, view of the current technology trends in edge computing.

As for most fast moving technologies like edge computing, several terms carry similar semantic meaning, such as:

- Fog computing, popularised by CISCO,
- Cloudlet, popularised by Carnegie Mellon University, and
- Smart grid (computing).

Most important is to understand the differences and overlaps of edge and fog. According to [FOG-V-EDGE], "*The key difference between the two architectures is exactly where that intelligence and computing power is placed [..]*

- *Fog computing pushes intelligence down to the local area network level of network architecture, processing data in a fog node or IoT gateway.*
- *Edge computing pushes the intelligence, processing power and communication capabilities of an edge gateway or appliance directly into devices like programmable automation controllers (PACs).*"

Whether or not we agree with these distinctions between edge and fog computing, mF2C includes both aspects in its scope.

In order to build a system linking cloud computing, mobile, sensors and actuators, the placement of intelligence must include both the surrounding networking environment of the edge (aka fog), as well as the edge itself.

Therefore, in the context of mF2C, both fog and edge advancements are relevant.

However, for the purpose of the study of technology trends in this section, our scope for edge computing will limit itself to the edge. This means that the management aspect of edge devices is less prominent in this review.

To explain the technology push towards edge computing, it is useful to understand the business drive. Analysts predict that edge computing will represent the 'next billion-dollar market' [EDGE-MARKET]. This explains why many tech companies are positioning themselves in relation to this segment.

While most development seems driven by private organisations, open source projects are also contributing to this trend.

3.5.1. Reference solutions

In the open source world, the 'Do It Yourself' movement has created a lot enthusiasm with platforms like the Raspberry Pi and Arduino, along with a flurry of open source software tools that take advantage of these devices. While these platforms are only one aspect of what would compose an edge computing platform, they have helped build communities and possibly democratise this sector, which was dominated by proprietary solutions before their appearance.

In this section, we highlight reference solutions that deliver edge computing solutions in the broadest sense, including both open source and proprietary products and services.

3.5.1.1. BRCK

An interesting example of solutions emerging from community and open source efforts is the 'BRCK' project [BRCK]. According to their website, "BRCK was designed and prototyped in Nairobi, Kenya. We wanted a connectivity device that fit our needs, where electricity and internet connections are problematic both in urban and rural areas." The BRCK project is currently working on a second version of its original product, which is expected to be released in Q1 2017.

These grassroot movements show the need for low power and low cost edge platforms, promoted by local communities. One particularity of the BRCK is its emphasis on a ruggedised design, able to support rough and challenging environments (e.g. Africa and India). It is based on an Arduino core, integrates mobile communication and can be managed from a cloud application via an API.

3.5.1.2. Open Edge Computing

Open Edge Computing [OEC] is a community effort building on Carnegie Mellon University's early work on Cloudlet. The community effort, mostly USA based, is focusing on the networking challenges of edge computing. The association of several large companies to the community suggest that it is gaining traction. OEC is also publishing reference architectures and APIs that mF2C should review during its architecture and design definition phases.

3.5.1.3. NuvlaBox

The NuvlaBox from mF2C partner SixSq, is an edge computing platform, built from open source components with integrated remote control capabilities. The box acts as a local private cloud, where applications are dynamically deployed inside virtual machines. The current versions (Standard and Mini) are based on a fanless system containing an Intel x86 platform, supporting full Linux and Windows operating systems.

While the current NuvlaBox portfolio is applicable for middle tier mF2C applications, adding a machine with a smaller form factor and corresponding lower price would extend its applicability to the micro agents needed at the very edge.

On the proprietary side, several companies offer edge platforms, such as Cisco, Riverbed and Hewlett Packard Enterprise (HPE). For example, HPE provides an Edge Line Server product line, targeting edge applications, with a range of form factors, from rack mounted blades to fanless devices (similar to the NuvlaBox).

3.5.1.4. Nebbiolo Fog Computing Platform

Nebbiolo Technologies a recent startup led by the main fog computing promoter, F.Bonomi (former at Cisco), has recently delivered its solution for fog management. The proposed solution is built on three main concepts, the fogNode, the fogOS and the fogSM. The fogNode is a hardware platform embedding all envisioned functionalities for fog computing, supported by a fogOS, a software stack enabling secure fog functionalities deployment, and managed by the fogSM, devoted to manage computing and networking systems, end-to-end wise, under a secure management framework.

Our preliminary review of these commercial products indicates that while the hardware is mature. However, most of the software has been inherited from either data center or networking type applications, neither of which supports or embraces the dynamic application environment mF2C is targeting.

The next few sections explore companion technologies and trends relating to edge computing.

3.5.2. Containers in the edge

Traditionally, cloud execution environments have been based on virtualisation, where users are provided a complete operating system, boxed inside a virtual machine. More recently, this model has been relaxed somewhat with the introduction of containers, which offer a lighter weight packaging alternative.

While the security context of virtual machines and containers are significantly different, containers come with the benefit of requiring fewer resources, compared to virtual machines. In the context of edge computing, this means that resource-constrained platforms can still deliver users with a semi-virtualized environment to host applications.

This trend has been explored already by mF2C partners, such as ATOS, STFC and SixSq. Preliminary internal results seem to confirm the potential of containers in constrained devices and should be further explored.

3.5.3. Edge and IoT

Edge computing predates *Internet of Things* (IoT). Pushed by new communication protocols (e.g. LoRa, Sigfox, NB-IoT) to allow low power devices to communicate over large distances, new use cases have appeared that extend the reach of connected objects. An initial, perhaps naive, push to have all connected object to communicate directly to a cloud based server is now seen as unrealistic. Indeed, given the projected explosion of connected devices coupled with security and privacy concerns, the industry is realising that building gateways and a hierarchy of devices for the aggregation of data is needed.

An interesting trend currently pushed by companies such as [Orbiwise] is to build local IoT gateways to allow more local processing of IoT sensor data. This is similar to the vision of coupling IoT long range connectivity to local processing, as proposed by the edge computing architecture.

This tiered architecture seems to bring the best of both worlds, delivering reach at scale and a sufficient level of local control.

3.5.4. Edge and Function as a Service

Another potential companion of edge computing coming from an extension to the [NIST] model of cloud computing (* as a Service) is *Function as a Service* (FaaS). This was popularised recently by Amazon Web Services, under the name *AWS Lambda*. The idea of FaaS is that instead of defining a complete runtime environment, such as a virtual machine or a container, the *function* is defined, to be executed on a managed environment, based on a pre-defined trigger or action.

Where edge computing and FaaS could (or should) intersect is in the case where edge platforms, or fog layers, start supporting FaaS. In this case, running functions would potentially be easier to move around systems, such as mF2C, in order to gain further flexibility and dynamism.

3.5.5. Edge and Industry 4.0

The application or fusion of trends such as edge computing, IoT and fog, in the domain of manufacturing and industrial processes is now referred to as *Industry 4.0*. A key challenge to Industry 4.0 is the need to coordinate several fast changing processes, in order to efficiently control these processes, while maintaining high-level of security and generating relevant information to feed high-level functions (aka *cognitive computing* according to IBM), to produce diagnostic and trend analysis.

With its ability to move compute to the edge of the network and therefore close to machines and robots, edge computing is potentially a key ingredient in delivering Industry 4.0 solutions.

3.5.6. Transformation of data into information

Next generation systems, embedded with high density connected components, promise to be a source of enormous data. For example, the C-Series, the latest airplane by Bombardier, could be producing over 844 TB of data per 12 hours flight [C-SERIES]. Airlines operating fleets of this type of craft will be faced with a potential deluge of data.

It is not realistic to expect to transfer, store and process this much data. Nor does it make sense. Analysis applications need structured information instead, in order to predict important event and derive significant trends. This information, using edge computing, could be produced from the raw data, closer to the components or system generating the data.

In the example of the C-Series, the aircraft itself is probably the right place to perform this type data transformation. With expected results from mF2C, the ability to feedback high-level analysis down to the edge layers to optimise the transformation of data into information would potentially address the risk of losing relevant data, in the case where the transformation engine was not optimally calibrated.

3.5.7. Conclusion

In conclusion, the latest trends in edge computing are rich and provide interesting and relatively mature foundations for the mF2C project. The open source domain, including previous development performed by mF2C project members, seems interesting as potential starting point.

However, this trend analysis has not covered the management aspects more related to cloud orchestration and fog computing, which will need to integrate harmoniously with the edge layer (or layers), in order to deliver on the mF2C vision.

Finally, companion technologies are also showing interesting avenues for exploration, in order to leverage the potential and environment of edge computing technologies.

3.6. Key Takeaways

- Review of available platforms and tools (Open Source, Commercial, from European projects) for managing:

  - Cloud: plenty of solution, only listed some of them
  - Fog: only two solutions coming projects still under development
  - IoT: different solutions, middlewares, platforms, etc. trying to take advantage and easy the development of apps with the already connected to Internet heterogeneous edge devices.

- Regarding HPC hardware, the first non-volatile memories are available in the market (Intel), and also compute-in-storage architectures that enable the execution of data-centric algorithms within the storage system (IBM).
- Regarding HPC software platforms, traditional parallel file systems coexist with newer solutions that bring computation closer to data (HDFS, dataClay).
- Heterogeneity of computing devices increases the programming complexity of applications. Task-based programming models and runtimes transparently manage this heterogeneity and allow to exploit parallelism, both at the CPU level (OmpSs) and in distributed environments (COMPSs).
- Review of available cloud orchestration tools (for IaaS and PaaS), including containers solutions based on docker and other technologies
- Although formal international standards are defined by geographical or nationally driven bodies such as ISO/IETC and ETSI, the actual technical content of standards is often developed by industry or community-driven standards groups such as DMTF, OGF, OCF and the OpenFog Consortium.
- Standards groups of particular relevance to mF2C include ISO/IEC JTC1 SC38, ISO/IEC JTC1 SC41 (to be formed in 2017), DMTF, OCF, OpenFog Consortium, OGF, ETSI and LoRA Alliance.
- Technical standards of particular relevance to mF2C include DMTF's CIMI and OGF's OCCI for interoperability; OCF's OIC for IoT device intercommunication; the Open Fog Consortium's Reference Architecture, and the LoRA Alliance's LoRaWAN protocol for low-energy wireless communication.
- Certain open-source projects have been formed to complement the technical specification of standards. Of particular relevance to mF2C is IoTivity from the Open Connectivity Foundation.
- The latest trends in edge computing are rich and provide interesting and relatively mature foundations for the mF2C project. The open source domain, including previous development performed by mF2C project members, seems interesting as potential starting point.
- This trend analysis has not covered the management aspects more related to cloud orchestration and fog computing, which will need to integrate harmoniously with the edge layer (or layers), in order to deliver on the mF2C vision.
- Companion technologies are also showing interesting avenues for exploration, in order to leverage the potential and environment of edge computing technologies.

4. Business trends

Since 2012, when the European Commission adopted the European Cloud Computing Strategy [CloudStrategy] in order to provide a common framework all over the State Members to bypass several issues surrounding the European cloud adoption, such as incompatibility of standards and security issues, the cloud computing paradigm has evolved to become a key driver for innovation. To this respect, the European Cloud Partnership [CloudPartnership] has been integrated into the Digital Single Market strategy [DSM] as the pillar for transforming the European ICT landscape until 2020. Since then, cloud computing has been evolved to become firmly positioned in the IT market. However, the rise of the Internet of Things and the need of processing data at local devices instead of in a remote data centre has resulted in an architectural pattern called Fog computing. Cisco, who introduced the term, defines it as a paradigm that extends the current cloud and its services to the edge of the network. Thus, fog computing takes advantage of cloud offering improving its efficiency and reducing the amount of data transferring.

The following subsections contain an overview of cloud, fog and IoT trends with respect to business needs.

4.1. Cloud computing

In 2013 Gartner predicted that nearly half of large enterprises will have hybrid cloud deployments by 2017, as virtualization is expected to reduce capital expenses, while standards and automation will reduce operational expenses [Gartner2013]. In 2016, Gartner's predictions were even one step further, assuring that by 2020 a no-cloud policy will be as unusual as a no-internet policy nowadays [Gartner2016]. Depicted by offering, IaaS market is expected to reach $173B in 2026, while PaaS and SaaS portion will grow to $55B in the same period [Statista2015].

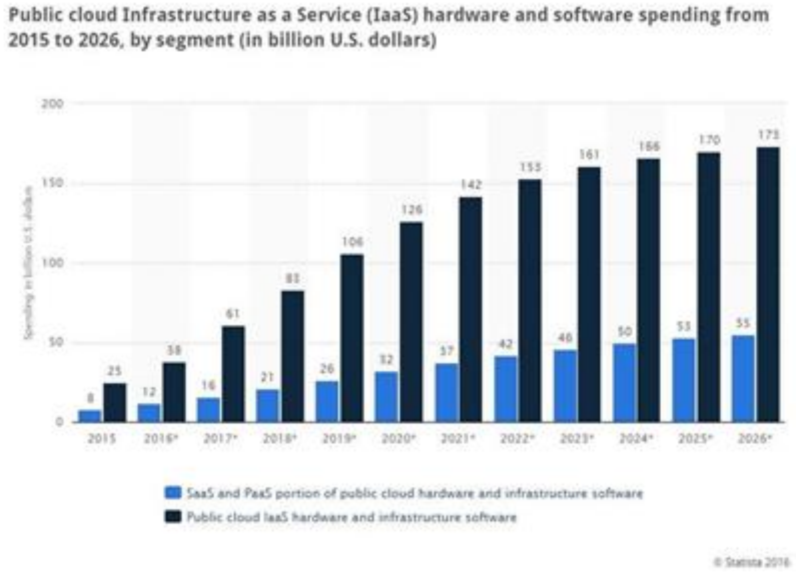
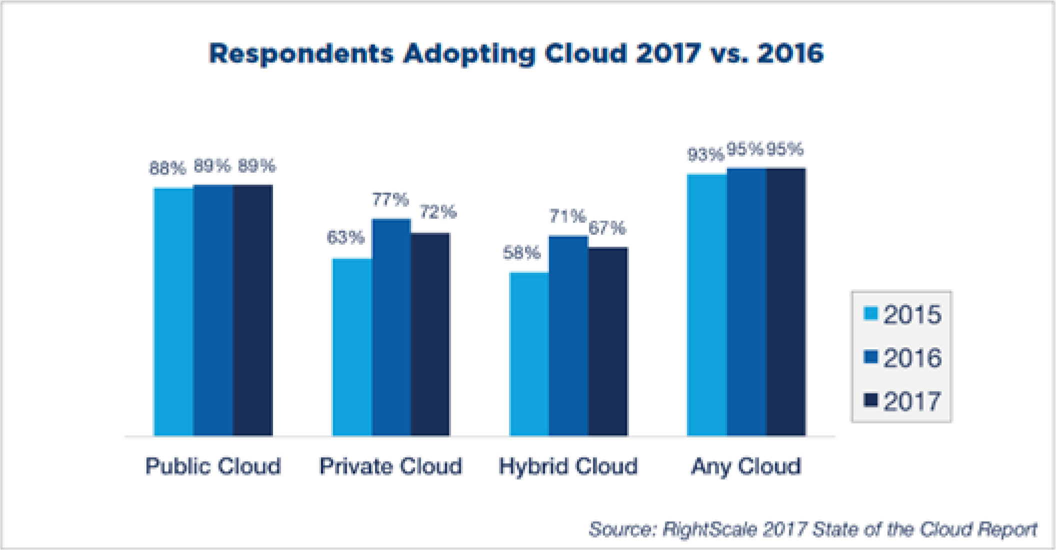


Figure 11 Public cloud IaaS spent 2015-2026

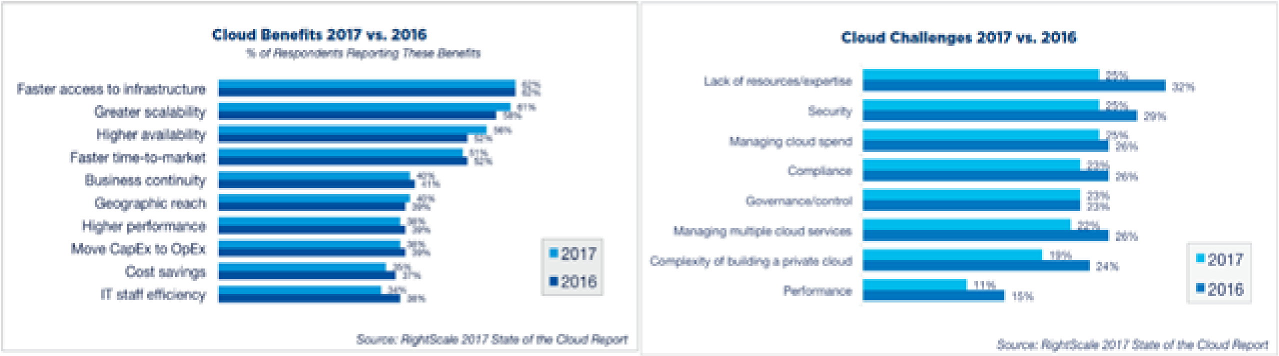
In January 2017, RightScale [RightScale2017] conducted a survey about the positioning on cloud computing adoption. Results showed that a 95% of the surveyed organizations are experimenting with IaaS, bearing in mind the optimization of cloud costs (53%) considering an estimation of a 30% of wasted spend. As it can be seen in the figure below, this percentage of cloud adopters hasn't varied too much in the last three years.



Source: RightScale 2017 State of the Cloud Report

Figure 12 Cloud adopters by type of cloud

The same survey shows that the expected benefits of cloud adoption are flat, compared with 2016. However, it is significant that potential adopters expect increased scalability and availability of resources while, at the same time, aren't reluctant to sacrifice cost savings or staff efficiency. Also the perception of the pain adoption points has significantly change, as security, lack of resources or cloud spend have been reduced.



Source: RightScale 2017 State of the Cloud Report

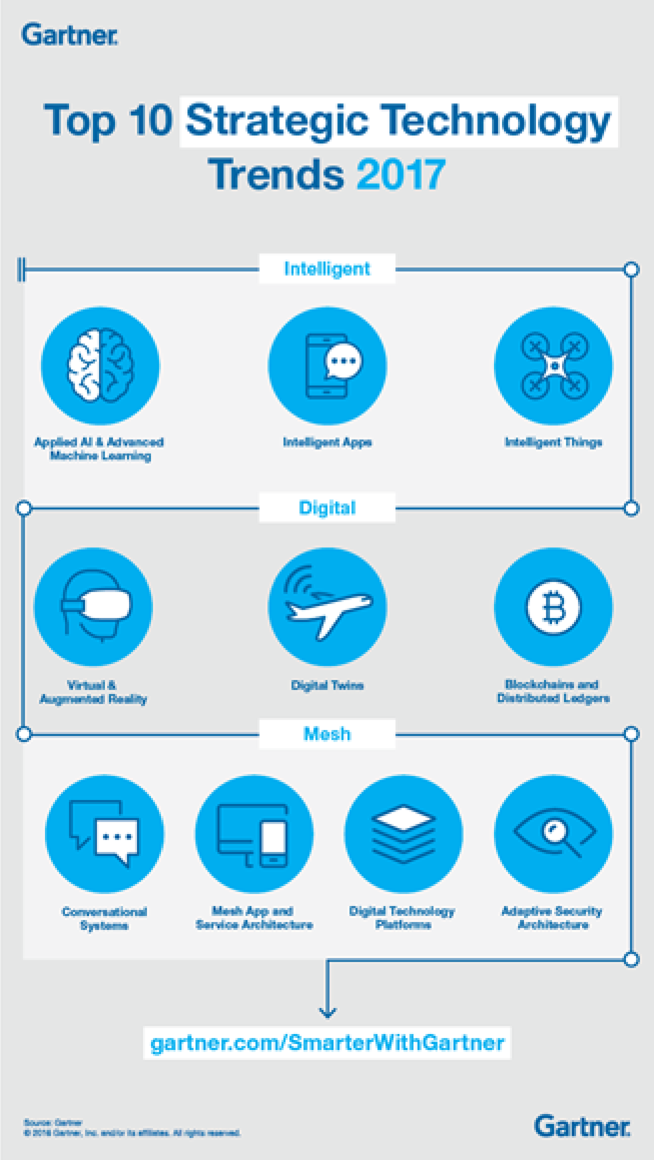
Source: RightScale 2017 State of the Cloud Report

Figure 13 Cloud benefits and challenges 2017-2026

EuroCloud defines cloud computing as "*one of the most important drivers of knowledge based society, where physical resources are optimised and shared resources are common*" [EuroCloud]. The normalisation in the usage of cloud-based technologies, together with the expansion of IoT into a digital mesh has completely changed the traditional point of view on cloud computing trends. Gartner [Gartner10] identifies 10 strategic technology trends for 2017, based on intelligent, digital and mesh concepts.

Connecting people, devices and services into one single digital ecosystem is a real need for the following years, and new solutions must be able to adapt to specific user needs in a dynamic way.

These trends are expected to defeat the barriers between physical and digital world, to create new business opportunities, as it will be explained in the following subsections.


Gartner
Top 10 Strategic Technology Trends 2017
Intelligent
Applied AI & Advanced Machine Learning
Intelligent Apps
Intelligent Things
Digital
Virtual & Augmented Reality
Digital Twins
Blockchains and Distributed Ledgers
Mesh
Conversational Systems
Mesh App and Service Architecture
Digital Technology Platforms
Adaptive Security Architecture
gartner.com/SmarterWithGartner
Gartner

Source: Gartner
© 2016 Gartner, Inc. and/or its affiliates. All rights reserved.

Figure 14 Top Ten Strategic Technology Trends 2017

This situation is leading the cloud services market to the third big wave [Enescu2014], where the concept of fog computing is of special relevance. Both, cloud computing and IoT aims to increase the efficiency of daily tasks. However, while IoT generates massive amounts of data, cloud computing provides the path to process it. In this context, fog computing paradigm is born with the aim of extending cloud services to the edge of the network, bridging the gap between data and locality to improve efficiency in a more secure way.

ABI Research [abiresearch2016] states that the most significant IoT trend is the shifting balance from cloud computing to edge computing. According to BI Intelligence [businsider], the biggest benefit of IoT for businesses is the data generated by sensors and devices. There are millions of IoT devices connected to an edge solution nowadays, and this trend will continue growing in the future as it is shown in the figure below.

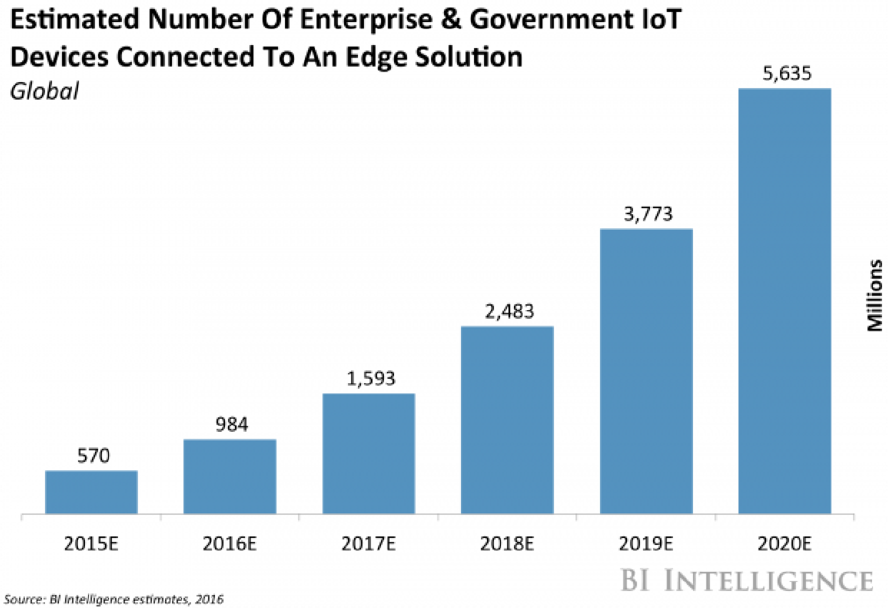


Figure 15 Enterprise devices connected to the edge

The same report also shows to relevant takeaways: standardisation is a real need to unify deployments; and manufacturing, utilities, energy and transportation industries will be the early adopters of this kind of technology based on their needs. PrimsTech [embedcompute2017] supports this assumption, based on the incompatibility of Industrial Internet of Things (IIoT) applications with cloud-centric architectures.

Gartner [Gartner10] recognises micro and edge computing environments as one of the top technology trends that is impacting IT operations. Furthermore, in its Hype Cycle for Emerging Technologies [GartnerHype2017] IoT platforms are expected to set the basis for new business models with more interaction between humans and technology. While, at the same time, in its Hype Cycle for Infrastructure Strategies [GartnerStrategies2016] recognises edge computing and IoT edge architectures as an innovation trigger for market realisation in 2-5 years. Finally, in The Edge Manifesto [GartnerEdge2016] Gartner supports the movement to the edge, closer to users in order to support consumerisation and democratisation of IT. This movement is expected to increase user experience as a master piece of core new digital businesses, which will complement traditional ones. As IDC predicts [idc2017] a 30% of IT assets will be owned or operated in edge locations and micro data centres by 2018. Moor Insights and Strategy goes one step further assuring that "*the concept of a more intelligent flexible cloud can help guide carriers towards better business outcomes*" [moorinsightsstrategy2015]. Thus, moving to the edge can speed deployment driving the most value to flexible compute environments.

4.2. Internet of Things

Many areas of our lives have changed with the exponential growth of connected devices, and in this context the industrial sector is not any exception. In the field of smart solutions such as the Internet of Things (IoT), innovations are changing the delay life and industrial processes by blending the real and the virtual worlds via novel ICT solutions. In the near future is expected that machines, products, systems, and people will be able to communicate locally and in real-time so that they can manage their needs in an efficient fashion.

To guarantee quality and replicability as well as increasing technology acceptance, future networks will require open and standard solutions prior to be implemented. Moreover, novel smart solutions will also need an easy interface to the connected devices to request services with strict requirements related to bandwidth, delay, jitter, packet loss or redundancy. In response, the network will grant the requested resources automatically and program the intermediate networking devices based on dynamic profiles and privileges assignation. Similar requirements come from business applications where smart services request for particular network resources.

Taking into account mF2C project, novel technologies as FOG devices and Cloud management layers are cornerstone of automatic applications, and they are of great interest for this project since they provide a mandatory combination of programmable connectivity, rapid service provisioning and service chaining. The real-time control of devices, machines and products will provide direct benefits regarding: **i) Productivity:** automation fulfils the aim of the customers to constantly run and monitor processes over the time; **ii) Quality:** with innovative services as continuous monitoring, applications are permanently optimized to perform activities with precision and high repeatability; **iii) Flexibility:** the creation of virtualized services and functionalities provides the basis for the advent of novel more flexible solutions in a short period of time; **iv) Information Accuracy:** adding automated data collection will allow gathering key production information, improving data accuracy, and reducing costs. Accordingly, customers will be able to take the right decisions at any time; and **v) Safety:** the presence of technicians on hazardous operations should be avoided as virtualized services limit the need of human workload in undesired environments. Such benefits may be then translated to interesting business opportunities for utilities and manufactures in order to reduce OPEX via CAPEX investments.

In this context taking into account the market forecast for IoT, several reports shown the potential numbers of this sector in the next years:

1. ABI Research’s: IoT-related value-added services are forecast to grow from $50B in 2012 to $120B in 2018, attaining a 15.71% CAGR in the forecast period.

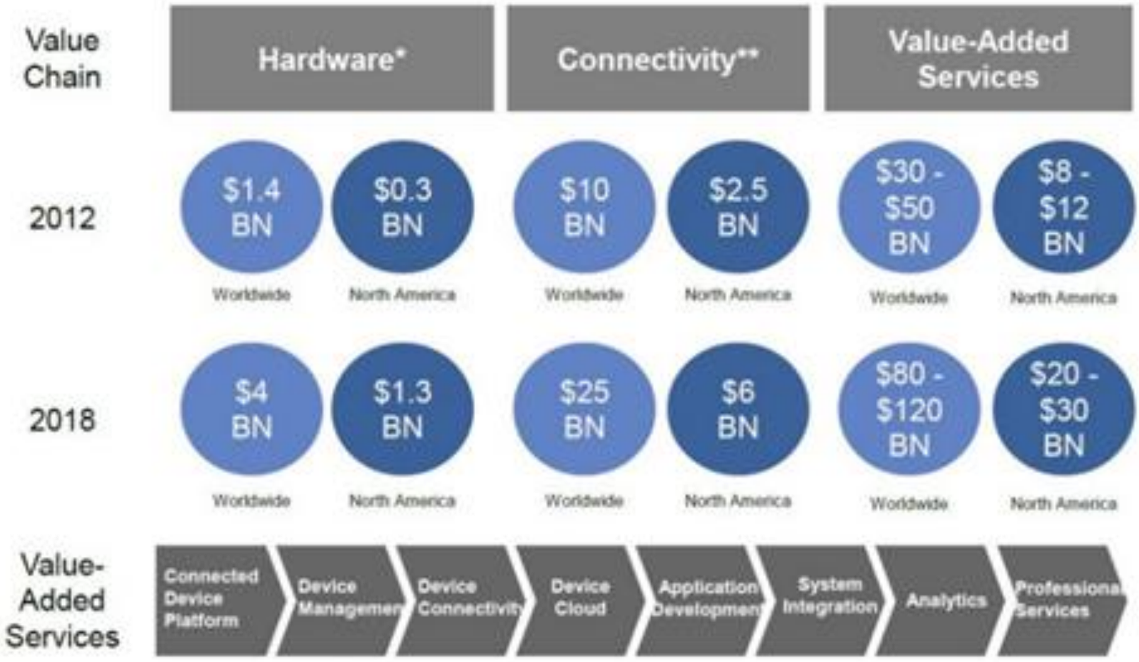


Figure 16 Growth predictions in the IoT market

2. Cisco predicts the global Internet of Things market will be $14.4 trillion by 2022, with the majority invested in improving customer experiences. Additional areas of investment including reducing the time-to-market ($3T), improving supply chain and logistics ($2.7T), cost reduction strategies ($2.5T)

and increasing employee productivity ($2.5T).

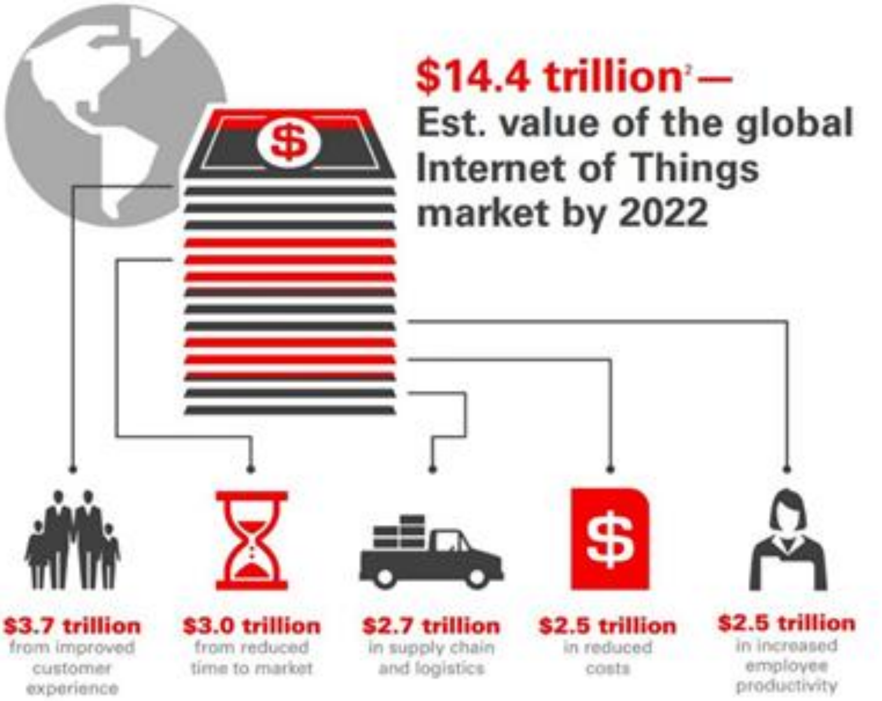


Figure 17 Estimated value of the IoT market by 2022

3. BI Intelligence: Software and services will be a $600B market by 2019, attaining a 44% CAGR from 2015 to 2019. BI Intelligence also predicts the number of devices connected via IoT technology will grow at a 35% CAGR from 2014 to 2019.

Moreover, the innovations that will be enable the aforementioned economic forecasts are following presented subdivide them considering the main technology sectors of the IoT value chain:

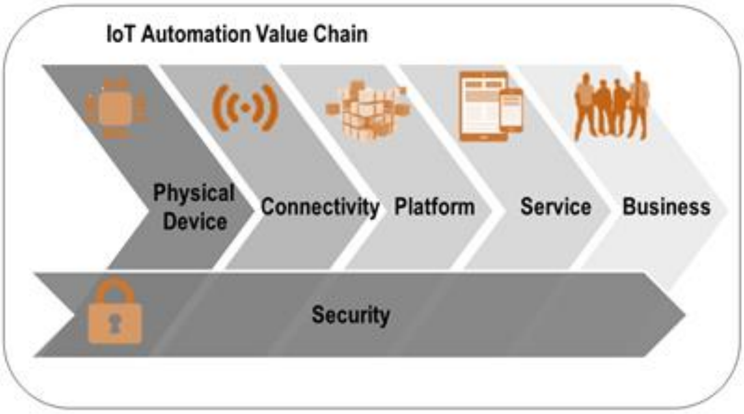


Figure 18 IoT value chain

i) **Physical Devices:** embedded systems are intelligent central control units that typically operate as information-processing systems inside an enclosing product for a set range of device-specific applications. These devices are low-power and are connected with the outside world allowing physical systems to be increasingly interconnected with each other and with the end-users through Internet; ii) **Connectivity:** the evolution of embedded systems into futuristic cyber physical systems for *Internet of Things* depicts the vision of a potential global market of services. Through the evolutionary development of more powerful embedded system, customers will have the opportunity to operate more complex processes locally. However, such devices need to be networked over the Internet or corporate networks to take full advantage of all the benefits of the Internet of Service. Indeed, novel connectivity solutions able to serve Industrial IoT (IIoT applications are currently coming out as: NB IoT, LoRaWAN, etc; iii) **Platform:** As automation relays on specific and heterogeneous data, this sector has identified the need for designing system integrator platforms able to allow interoperability functionalities managing high volume of data as well as different products in a centralized fashion; iv) **Service able to monetize the collected data:** the digital transformation of today operations will be possible only if, on one side, technologies are ready and trustable, and, on the other side, business models are appropriate to facilitate its adoption. Regarding business models, novel services need to be designed to translate the collected data and interconnected devices to a monetary value. Following, we present a short list of novel services that are subdivided considering their complexity: **1-Monitoring -** Real-time condition monitoring: supervision of the real-time operations and processes. **2-Control** - Work force control: coordination and control of the work force deployed in the field. **3-Optimization** - Predictive maintenance: anticipate possible operational problems by identifying common patterns and cluster set of data. **4-Autonomy** - Autonomous personalized operation: automatic reporting and analytic tools to ad-hoc personalize operation and processes.

4.3. Big data and IoT

The evaluation of IoT applications in the environments where these systems will be deployed (e.g. cities, offices, shopping centers, hospitals, etc.) shows a broader view of potential benefits and challenges, highlighting how various IoT systems can maximize value, in particular where they interact [McKinsey2015] .Interoperability between IoT systems is critically important: when IoT systems communicate each other their value is multiplied, so interoperability is an enabler for maximizing benefits.

Most IoT data are not currently used nor stored, the current use is mostly limited to address anomaly detection and real-time control, so a great deal of additional value remains to be captured, by using more data, as well as deploying more sophisticated applications such as using performance data for predictive maintenance to predict and prevent breakdowns, or to analyze workflows to optimize operating efficiency. IoT can be a key source of big data to be analyzed to capture value [Forrester2016].

Business-to-Business (B2B) applications can create more value than pure consumer applications. While consumer applications such as fitness and e-Health monitors, home automation and self-driving cars attract the most attention and have tremendous potential for creating value significant value, there is even greater potential value from IoT use in business-to-business applications. In many instances, such as in worksite applications (mining, oil and gas, and construction), there is no direct impact for consumers. A great deal of additional value can be created when consumer IoT systems, such as connected consumer health-care products, are linked to B2B systems, such as services provided by health-care providers and clients. B2B market is expected to generate nearly 70 percent of potential value enabled by IoT [McKinsey2015] .

Customers will capture most of the potential value over time, the users of IoT (business, other organizations, consumers) could capture 90 percent of the value that IoT applications generate.

In many settings, customers will capture value in both direct and indirect ways, such as being able to buy more efficient machinery that is designed using IoT data from older products in use. Of the value opportunities created by the Internet of Things that are available to technology suppliers, in general the largest share will likely go to services and software and less will likely go to hardware.

The industry is evolving around IoT technology, and brand new opportunities for both incumbents and new players are being created. The Internet of Things will change the bases of competition and drive new business models for user and supplier companies. The Internet of Things will enable—and in some cases force—new business models. For example, with the ability to monitor machines that are in use at customer sites, makers of industrial equipment can shift from selling capital goods to selling their products as services. Sensor data will tell the manufacturer how much the machinery is used, enabling the manufacturer to charge by usage. Service and maintenance could be bundled into the hourly rate, or all services could be provided under an annual contract. Performance from the machinery can inform the design of new models and help the manufacturer cross-sell additional products and services. This "as-a-service" approach can give the supplier a more intimate tie with customers that competitors would find difficult to disrupt [McKinsey2015] .

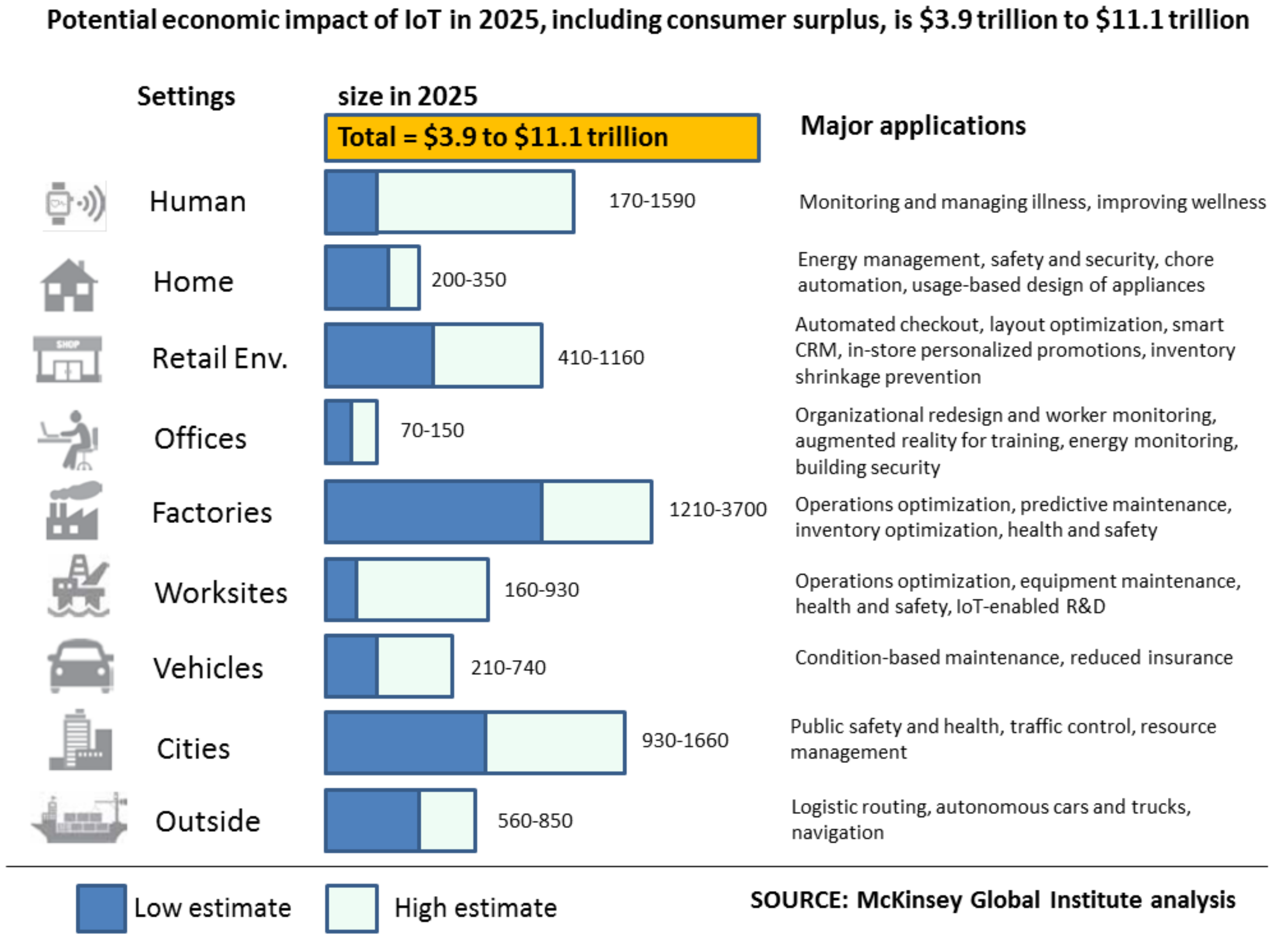


Figure 19 Economic impact of IoT in 2025

IoT is expected to change the way datacenters are designed, deployed and sited [451Research2015]. Two levels of compute, core and edge processing, are emerging, where each layer demands different kind of datacenters. The same differentiation is seen at processing level [HPE2016] where processing needs to be dynamically distributed between core and edge. New strategies aim at moving computing power, data acquisition and data management to the edge of the network, allowing faster access to relevant data and using available bandwidth not for raw data stream but only for critical one.

To realize the full potential from IoT applications, technology will need to continue to evolve, providing more sophisticated and robust data analytics. In particular services and IoT analytics and applications are expected to capture 60% of the growth from IoT in the next 5 years [BCG2017]. An enabling factor for this seems to be transparency to users about what data is being collected, how is being shared and ability to check it in real time [Deloitte2016].

To realize the expected impact and potential market for IoT, providers will have to work together within the IoT provider ecosystem of infrastructure, hardware, software, and other vendors to develop solutions that have greater potential to drive significant business value for enterprises [Deloitte2016].

In almost all deployed or foreseen settings, IoT systems raise questions about data security and privacy, solution providers and enterprises need to work together to protect break points, as well as enables rapid detection and mitigation of security breaches [Deloitte2016].

In most organizations, taking advantage of the IoT opportunities will require leaders to fully adopt data-driven decision making [McKinsey2015] . This point is rising among directors that see the Big Data framework, as an investment priority for the coming years [PoliMI2016].

This investment needs to address the following:

- **Strategy**, how the organization will manage the data in terms of strategic plan,
- **Data Management**, how the data are distributed among the enterprise systems and available for Service Providers,
- **Governance**, the whole set of structures, rules and strategies that guide the company,
- **Technology**, the adopted technological approach to manage the Big Data and the analysis on it.

A data-driven decision making have to follow a narrow path from Descriptive Analytics, what's happened, to Predictive Analytics, what will happen, to Prescriptive Analytics, what can be done, and to Automated Analytics, with automatic actions without human control when fast response is mandatory, e.g. in finance or health scopes [Gartner2016].

This path needs to be sustained by an adequate technological framework, where the analysis of heterogeneous data sources requires new architectural approaches [ForresterWav2016]. The management of the whole life cycle of the data, from collection to its conservation, passing to the analysis, requires the use of new innovative technologies.

Thus a scalable infrastructure able to process large amount of data in real time is needed, minimizing the potential information leaks. The analysis phase requires an evaluation of complex architectures that combine capabilities of real time and batch processing. The retention of data requires the use of technologies that go beyond traditional relational database to manage new information sources.

The IoT will speed up this evolution path because IoT produces huge quantities of a type of asset that can be sold or exchanged: the data. The ability to identify facts, hidden relations in the data available to organizations, not only allows to optimize processes and increase competitiveness, but also can open new opportunities for value creation. Data monetization is the process of generating new revenues through the sale or exchange of data in the possession of the organization and through the exploitation of these for the generation of new products and services.

Big Data Architectures and Technologies promise to be the main driver for this.

4.4. Security

There have been a number of IoT and "smart device" compromises, some of which are listed in the proposal, some have occurred since - turning car alarms off, brakes on, or open locks; or using IoT devices for DDoS attacks. Eventually, the public (or perhaps less likely regulator) will wake up and require some kind of security for IoT devices. If compromised devices end up giving the manufacturer a bad reputation, and hence lower sales, eventually security will make sense.

Adding features vs adding security is the perennial battle between marketing or eager coders and the security engineers. It is the feature that sells the device, not the security. Flaws can tend to get hidden or ignored in the hope that no one will discover it and exploit it, but if history teaches us anything, it is that "security through obscurity" will not work, particularly in recent scenarios where the attacker may be a well resourced organised crime mafia, or a foreign government.

There is a close analogy in the cloud world. Commercial CSPs are keen to provide services to government, to businesses, and to individuals, but ran into issues with trust and security. These issues have largely been solved, or are solvable. Both IoT and "smart" devices, likewise, are likely to face an issue of trust - which may be well placed, as the industry should be able to convince users that their devices are safe, by providing full access to both device, hardware, and software/firmware to independent security researchers (one notable difference with the cloud world is that IoT and smart devices are generally mobile, and an adversary could buy them and subject them to analysis; and in current global IT security one will have to assume that this attacker is very resourceful indeed.) It may be useful to develop industry standards or at least best practices, and there is in fact already an IoT working group in the Cloud Security Alliance, and there is an IoT group in ISO/IEC JTC1 (namely WG10 - IoT). Indeed, recognising the importance of the IoT market, ARM has worked with Intercede, Solacia and Symantec to develop the Open Trust Protocol (OTrP [OTrP2016]) for connected devices. The protocol combines a secure architecture with trusted code management and is currently a IETF draft. OTrP is a high level management protocol and uses the Public Key Infrastructure (PKI) and Certificate Authority-based trust architectures, enabling service providers, app developers and OEMs to use their own keys to authenticate and manage trusted software and assets in a Trusted Execution Environment (TEE).

Other obvious security factors include the cost of implementing security - if millions of devices are manufactured, the manufacture and `onboarding' of a secure device had better not be too expensive -- RFID is a good example here as it has been designed to a very low cost and with limitations to its security; there are also obviously technical limitations as identified in the mF2C proposal, where a device may not be capable of implementing strong cryptography. Outside of a regulatory framework or explicit procurement requirements, the implementation of security remains a commercial decision (i.e., will we recover the costs, will it improve or reduce our market share, etc.)

A specific examples occurs with devices in the field: once in the field, getting hold of them to do security upgrade of their firmware can be extremely expensive - think of product recalls where products have been withdrawn from the market, usually for safety reasons. Relying on people to upgrade their own (personal) devices is also tricky; for example most homes own a router but few people upgrade the router firmware. Thus, a company should be pretty sure of their device security when putting a product on the market, thus requiring a significant effort and investment in both design and testing before the product is released.

4.5. Standardisation

Standards facilitate the interoperability and portability of data and systems across products – be they software or hardware - developed by different providers. This is generally seen as important by customers, preventing vendor lock-in, and facilitating innovation and the integration of devices from a range of suppliers increasing the value of their investments. Significant providers, however, can have concerns that the portability and interoperability that standards deliver facilitate customers moving to a competitor's offering. Depending on the design of the standard, the unique value-added features that a provider may offer, may not be exposable.

Thus although standards such as OCCI [OCCI] (by OGF [OGF]) and CIMI [CIMI] (by DMTF [DMTF]) have been developed to standardise cloud management, their uptake by cloud providers is relatively low.

These concerns appear to be less of an issue when dealing with interoperability of, and communication between, hardware products, however, and standards to discover and manage data centre infrastructure (such as DMTF's RedFish [REDFISH]) and IoT intercommunication (as driven by the Open Connectivity Forum [OCF]) have gained significant industry support. The OCF, for example, has at the time of writing over 325 members spanning industry and academia (12 diamond level members, 24 platinum, 147 Gold, 15 academic and 127 basic). The consortium has successfully produced v1.1.1 of the Open Interconnect Consortium standard [OCF-OIC] and is actively enhancing and extending the standards. Curiously though, at the time of writing only 3 products have been certified as OCF compliant as per the Certified Product Registry [OCF-CPR]. Regarding standards for Fog Computing specifically, the OpenFog Consortium [OFC] now contains 56 members at the time of writing, including many key competing players. As discussed previously these members see sufficient business value to actively pursue standards in fog architecture, communications, manageability and software infrastructure.

4.6. Digital Business

Digital business is a relatively new concept, which refers to the way a company interacts with their customers. A digital business model is fully customer oriented, bearing in mind that the key for success is to understand their needs. This kind of model is agile, cost-effective, global and scalable [DimensionData]. These objectives are presented in the evolution of platforms, more flexible and dynamic than traditional ones [GartnerDigital].

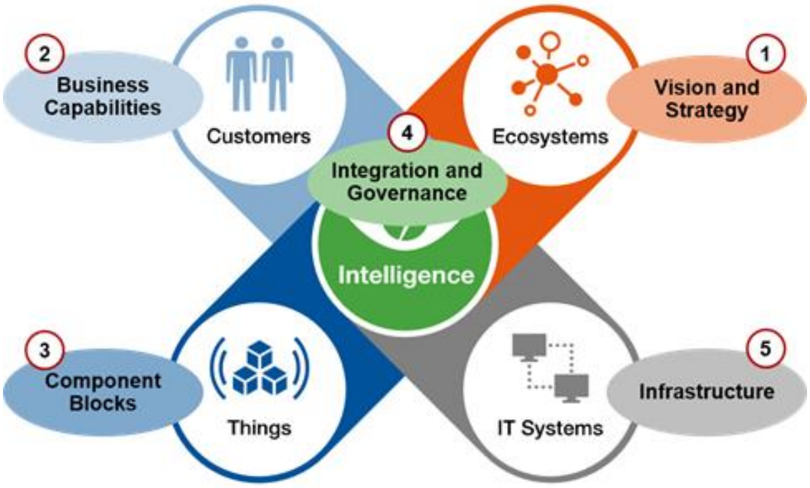


Figure 20 Digital Business ecosystem

As it has been presented in the previous sections, the evolution of technology has a direct impact on the business side. Traditional business models applied to cloud computing cannot cover the dynamicity of user needs as the range of scenarios has significantly increased with the combination of cloud computing, big data and IoT. Disruptive digital business may appear to improve customer experience on these scenarios taking innovation as the next frontier to be reached [GartnerTop].

According to Gartner [GartnerDecade], cloud computing can be considered as a vehicle for next-generation digital businesses, mainly due to the fact that many objections to cloud adoption are being discredited. In this sense, strategic relationships and partnerships with cloud technology and service providers are becoming more and more important. This does not mean the traditional relationship between one customer and one provider, but between one customer and several providers that can cover all its needs. These relationships are supported mainly through subscription and pay-per-use models in order to contain costs. This situation is also supported by Forrester predictions [Forrester2017], which consider that cloud continue will disrupt traditional models through 2020. One example of this is AWS, who is letting companies to use service abstraction to specific application areas going one step further from the traditional virtual and pay-per-use model to a most accessible one. Market is moving to cloud vendors who are software and solution enablers, offering an integrated framework to allow customers' abstraction from the infrastructure layer.

Customers are expecting an evolution of the cloud offering to support them reducing their time to market and simplifying their work [InformationAge]. Here appears the concept of serverless architectures, as they can be easily and fast developed and deployed to solve specific business cases.

In general lines, digital transformation is reshaping every business aspect being considered as the core component of business strategy. According to Forbes [Forbes2017], the evolution of technology is expected to change the way organisations are functioning nowadays:

- Cloud computing is not only a paradigm to be applied at technical level , but a mean to transform the company culture. Remote operations must be progressively included in daily activities.
- User experience must be included in any business as a way to involve and engage customers. This can also be done using virtual or augmented reality to enhance customer experience.
- Innovate fast to remain competitive in the market.
- Understand the potential of IoT and big data to exploit their value.

4.7. Key Takeaways

- The evaluation of IoT applications in the environments where these systems will be deployed shows a broader view of potential benefits and challenges, highlighting how various IoT systems can maximize value.
- Most IoT data is not currently used nor stored, IoT can be a key source of big data to be analyzed to capture and monetize value.
- A great deal of additional value can be created when consumer IoT systems, such as connected consumer healthcare products, are linked to B2B systems, such as services provided by health-care providers and clients. B2B market is expected to generate nearly 70 percent of potential value enabled by IoT.
- Customers will capture most of the potential value over time, the users of IoT (business, other organizations, consumers) could capture 90 percent of the value that IoT applications generate.
- The industry is evolving around IoT technology, and brand new opportunities for both incumbents and new players are being created. IoT will change the bases of competition and drive new business models for user and supplier companies.
- Services and IoT analytics and applications are expected to capture 60% of the growth from IoT in the next 5 years, where an enabling factor for this seems to be transparency to users about what data is being collected, how is being shared and ability to check it in real time.
- In the majority of foreseen settings, IoT systems raise questions about data security and privacy, solution providers and enterprises need to work together to protect break-points, as well as enables rapid detection and mitigation of security breaches.
- Business incentives for implementing IoT security (as opposed to more features) - regulatory compliance, reputation, industry best practice (e.g. Cloud Security Alliance IoT WG), mitigating against future costs (product recall, lawsuits)

5. Conclusions

This deliverable provides an overview of all the scientific, technology and business trends in the area of fog computing relevant to the mF2C project. This initial version of the deliverable (v1) is aligned to iteration 1 (IT-1) of the project. Further versions will be submitted later in year 2 and year 3. The technologies surveyed in this document will provide a basis for the project's architecture and system design by identifying state-of-the-art technologies, software, and standards that will feed the project. Also reviewed were existing security standards to include in the architecture design and potential security features required by the mF2C framework. Finally, existing and novel business models for mF2C were discussed.

The large data sets being generated by clients, IoT devices, things, and machine-to-machine connections are expected to overwhelm legacy networks, centralized data centres and today's cloud computing infrastructure. Therefore, real-time decision making at source is key to reducing high network round trip costs and this will rapidly drive demand for localized data analytics, storage & processing that cannot be met by legacy cloud technologies. These are the new 5G usecases such as Autonomous Driving Cars, Manufacturing, High Frequency Trading, etc, driving data processing closer to edge devices. The framework to be created in the mF2C project aims to extend data center and cloud computing to efficiently process, store & navigate relevant data across both datacenter & IoT models.

The actual usecases selected for mF2C will generate new data streams requiring localised analysis. These were specifically selected due to their low latency requirements, the potential high cost of networking, and the high volume of data. These create a value proposition for intermediate analysis locations between the cloud and the device. It will require the project to create new algorithms to run either on the cloud, edge server, micro-edge to enable the business models that will collectively push more relevant data to both the edge and datacentre.

The scientific trends reviewed in chapter 2 shows that while Cloud and Fog computing are conceptually similar, the differences in service orchestration and management for Fog require solutions that incorporate the constraints of edge devices, ie, energy, mobility, reliability, security and heterogeneity. To ensure Service Assurance (eg, QoS, QoE, Latency), the management of resources in the Fog requires a different approach to the Cloud due to the high distribution, heterogeneity and volatility of resources. IoT management solutions appears limited to proposed communication protocols only, and not to the entire stack (ie, network, data, and storage). As such, the mF2C framework will require a naming and address solution for all devices to support development of services. Solutions coming from HPC can be leveraged here due to similarities. These are also focused on accelerating access to data by means of new storage technologies, and by means of new architectures that brings storage and computation closer in the data center. Among these include NoSQL being increasingly adopted as database solution for Big Data systems. The exponential growth of data is generating increasingly complex datasets which in turn require analysis algorithms that take this into account. These include no longer analysing data in the background to discover patterns and lngitudinal changes, but to real-time analysis of dynamic data to support immediate decision making. The mF2c framework should be encouraged to find the trade-off between potentially limited computational capacity of edge devices and their immediate proximity to the data sources to reduce delayed decisions.

The technology trends reviewed in chapter 3 included the available platforms and tools, both opensource and proprietary, that would provide solutions for managing Cloud, Fog and IoT, with most of the latter still developmental. While cloud orchestration solutions for IaaS and PaaS, which also include containers, are well established, the opensource projects for Fog management are still maturing. Again, HPC provides suggested solutions to these problems. Both non-volatile memory and compute-in-storage architectures are now available that enable the execution of data-centric algorithms within the storage system itself. These complement traditional parallel file systems that co-exist with newer solutions to bring computation closer to the data. However, it is the heterogeneity of devices at the edge increases the programming complexity of application development. Some task-based programming models and runtimes can transparently manage this allowing parallelism, both at the CPU level and in distributed environments and should be included as part of the mF2C solution. To ensure the longevity of results of the project, the mF2C architecture should remain compatible with existing standards in this area ensuring success and adoption of the designed architecture. A number of opensource projects have formed, complementing these technical specifications, which could provide initial codebases and supporting libraries for the framework.

Finally, the business trends reviewed in chapter 4 evaluated research from different business analysts groups to feed the business value that the mF2C framework could generate. These include reports that all large enterprises will have hybrid cloud deployments by end of year, as virtualization reduces CapEx and automation reduces OpEx. Other areas include the evaluation of IoT applications in their deployed environments show a broader view of potential benefits and challenges, highlighting how various IoT systems can maximize value. With most IoT data not currently retained due to storage issues, these could become a source of big data for analysis and monetized. Business-to-Business systems is expected to generate 70% of potential value by IoT. As a new industry is evolving around IoT technologies, new business opportunities are appearing. This will change the basis for competition and drive the potential for new business models. Among these are services, IoT analytics, and applications which are expected to capture up to 60% of the growth in this area in the next 5 years. The enabling factor appears to be a lack of awareness from users of what data is being collected and shared. This raises questions about data security, privacy with solution providers and enterprises needing to work together to protect attack surfaces to enable rapid detection and mitigation of security risks. The business incentive for implementing IoT security will need to come from a series of regulatory and industry best practices to mitigate against future costs.

With this deliverable, the project has generated an understanding of what the current scientific, technical and business trends are within the domain of Fog and Cloud computing that will allow it to drive a first version of an architecture for the mF2C framework. Awareness of these trends should help to focus the project on areas that require new solutions and steer it away from areas that solutions already exist. Indeed, these pre-existing solutions may actually help to accelerate the development of both the Controller and Gearbox modules of the project by importing these codebases or reference architectures, therefore adding value.

References

[Arkian2017] Hamid Reza Arkian, Abolfazl Diyanat, and Atefe Pourkhalili. 2017. 2-MIST: Fog-based Data Analytics Architecture with Cost-Efficient Resource Provisioning for IoT Crowdsensing Applications. J. Netw. Comput. Appl. 82, December 2016 (2017), 152–165. DOI:http://dx.doi.org/http://dx.doi.org/10.1016/j.jnca.2017.01.012

[Barnsley2016] Frazer Barnsley *et al.* Building a prototype Data Analysis as a Service : the STFC experience. In: Proceedings of the 11th New Opportunities for Better User Group Software Conference (NOBUGS 11), Copenhagen, Denmark, 17-19 Oct 2016: 23-28. DOI: 10.17199/NOBUGS2016.proc. URL: https://indico.esss.lu.se/event/357/material/4/0.pdf

[Bonomi2012] Flavio Bonomi, Rodolfo Milito, Jiang Zhu, and Sateesh Addepalli. 2012. Fog Computing and Its Role in the Internet of Things. Proc. first Ed. MCC Work. Mob. cloud Comput. (2012), 13–16. DOI:http://dx.doi.org/10.1145/2342509.2342513

[Bonomi2014] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. 2014. Fog computing: A platform for internet of things and analytics. In Big Data and Internet of Things: A Roadmap for Smart Environments. Springer, 169–186.

[Cardellini2015] Valeria Cardellini, Vincenzo Grassi, Francesco Lo Presti, and Matteo Nardelli. 2015. Distributed QoS-aware scheduling in storm. Proc. 9th ACM Int. Conf. Distrib. Event-Based Syst. - DEBS ’15 (2015), 344–347. DOI:http://dx.doi.org/10.1145/2675743.2776766

[Cardellini2016] Valeria Cardellini, Vincenzo Grassi, Francesco Lo Presti, and Matteo Nardelli. 2016. On QoS-Aware scheduling of data stream applications over fog computing infrastructures. Proc. - IEEE Symp. Comput. Commun. 2016–Febru (2016), 271–276. DOI:http://dx.doi.org/10.1109/ISCC.2015.7405527

[Hunt2016] John Hunt *et al.* Bionic smart textile robotics. An application to the Shenzhen Peacock Innovation Award.

[Marin2016] Eva Marin Tordera et al. 2016. What is a Fog Node A Tutorial on Current Concepts towards a Common Definition. (2016). http://arxiv.org/abs/1611.09193

[Masip2016] X. Masip-Bruin, E. Martin-Tordera, G. Tashakor, A. Jukan, and G.J. Ren. 2016. Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. IEEE Wirel. Commun. 23, 5 (2016), 120–128. DOI:http://dx.doi.org/10.1109/MWC.2016.7721750

[Mukherjee2014] Arijit Mukherjee, Himadri Sekhar Paul, Swarnava Dey, and Ansuman Banerjee. 2014. ANGELS for distributed analytics in IoT. In 2014 IEEE World Forum on Internet of Things (WF-IoT). IEEE, 565–570. DOI:http://dx.doi.org/10.1109/WF-IoT.2014.6803230

[Lin2011] Yinglei Lin *et al.* An optimized design of compression sportswear fabric using numerical stimulation and the response surface method. Textile Research Journal 82(2): 108-116. DOI: 10.1177/0040517511424531

[Preece2009] Sally Preece, Yannis Goulermas, Lawrence Kenney, David Howard, K. Meijer and Robin Crompton. Activity identification using body-mounted sensors – a review of classification techniques. Physiol. Meas. 2009; 30:R1-R33

[Skarlat2016] Olena Skarlat, Stefan Schulte, and Michael Borkowski. 2016. Resource Provisioning for IoT Services in the Fog Resource Provisioning for IoT Services in the Fog. 2016 IEEE 9th Int. Conf. Serv. Comput. Appl. Resour. , November (2016). DOI:http://dx.doi.org/10.1109/SOCA.2016.10

[Yang2015] Erica Yang *et al.* Data Optimised Computing for Heterogeneous Big Data Computing Applications. In 2015 IEEE International Conference on Big Data (IEEE BigData 2016), Santa Clara, CA, USA, 29 Nov 2015 - 1 Dec 2015, (2015): 2817-2819. DOI: 10.1109/BigData.2015.7364087

[Vaquero2014] Luis M. Vaquero and Luis Rodero-Merino. 2014. Finding your Way in the Fog. ACM SIGCOMM Comput. Commun. Rev. 44, 5 (2014), 27–32. DOI:http://dx.doi.org/10.1145/2677046.2677052

[OTrP2016] The Open Trust Protocol (OTrP). Internet Engineering Task Force Internet-Draft version 0.3. Url: https://www.ietf.org/id/draft-pei-opentrustprotocol-03.txt

[Beloglazov2012] Anton Beloglazov, Jemal Abawajy, Rajkumar Buyya. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems, v. 28, n. 5, p. 755-768, 2012.

[Wolke2015] Andreas Wolke et al. More than bin packing: On dynamic resource allocation strategies in cloud computing. Information Systems, 51(C):83-95, 2015.

[Xiao2013] Zhen Xiao, Weijia Song, Qi Chen. Dynamic resource allocation using virtual machines for cloud computing environment. Parallel and Distributed Systems, IEEE Transactions on, v. 24, n. 6, p. 1107-1117, 2013.

[Hummaida2016] Abdul R. Hummaida, Norman W. Paton, Rizos Sakellariou. “Adaptation in cloud resource configuration: a survey”. In: Journal of Cloud Computing v. 5, n. 1, p. 1-16, 2016.

[Singh2016] Sukhpal Singh, Inderveer Chana. “QoS-aware autonomic resource management in cloud computing: a systematic review”, In: ACM Computing

Surveys (CSUR) v. 48, n. 3, p. 42, 2016

[Chaisiri2012] Sivadon Chaisiri, Bu-Sung Lee, Dusit Niyato. “Optimization of resource provisioning cost in cloud computing”. In: IEEE transactions on service computing, vol.5, no. 2; 2012. p. 67–78.

[Simoens2015] Pieter Simoens et al. Challenges for orchestration and instance selection of composite services in distributed edge clouds. In: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on. IEEE, 2015. p. 1196-1201

[Mukherjee2015] Arjun Mukherjee et al. Angels for distributed analytics in iot. In: IEEE World Forum On Internet of Things (WF-IoT), 2014. IEEE, 2014. p. 565-570

[Kwon2012] Young-Woo Kwon, Eli Tilevich. Energy-efficient and fault-tolerant distributed mobile execution. In: IEEE 32nd International Conference on Distributed Computing Systems (ICDCS), 2012. IEEE, 2012. p. 586-595.

[Alam2016] Md. Golam Rabiul Alam, Yan Kyaw Tun, Choong Seon Hong.2 “Multi-agent and reinforcement learning based code offloading in mobile fog. In: 2016 International Conference on Information Networking (ICOIN). IEEE, 2016. p. 285-290.

[Souza2016] Vitor Barbosa Souza, Xavi Masip-Bruin, Eva Marín-Tordera, Wilson Ramírez. “Towards Distributed Service Allocation in Fog-to-Cloud (F2C) Scenarios”, In: IEEE Global Communications Conference, Globecom 2016.

[Skala2015] Karolj Skala, Davor Davidovic, Enis Afgan, Ivan Sovic, Zorislav Sojat. “Scalable distributed computing hierarchy: Cloud, fog and dew computing”, Open Journal of Cloud Computing (OJCC), v. 2, n. 1, p. 16-24, 2015.

[Hypercat] http://www.hypercat.io/standard.html

[Xmpp] https://xmpp.org/

[Zeng2011] Deze Zeng, Song Guo, and Zixue Cheng, "The Web of Things: A Survey", Journal of communications, Vol. 6, No. 6, September 2011

[Zigbee] http://www.zigbee.org/

[CoAP] http://coap.technology/spec.html

[EnOcean] https://www.enocean.com/en/

[W3] https://www.w3.org/

[iotdb] https://iotdb.org/

[osgi] https://www.osgi.org/about-us/working-groups/internet-of-things/

[openmobilealliance] http://www.openmobilealliance.org/wp/overviews/lightweightm2m_overview.html)

[Cai2014] Hongming Cai, Li Da Xu, Boyi Xu, Cheng Xie, Shaojun Qin, and Lihong Jiang “IoT-Based Configurable Information Service Platform for Product Lifecycle Management” IEEE Transactions on Industrial Informatics, Vol. 10, No. 2, May 2014

[Främling2014] Kary Främling, Sylvain Kubler, and Andrea Buda, “Universal Messaging Standards for the IoT From a Lifecycle Management Perspective”, EEE

Internet of Things Journal, Vol: 1, Issue 4, Aug. 2014.

[Petrolo2017] Riccardo Petrolo, Roberto Morabito, Valeria Loscrì and Nathalie Mitton, “The design of the gateway for the Cloud of Things”, Annals of Telecommunications, Volume 72, Issue 1, February 2017.

[Kim2015] Seong-Min Kim, Hoan-Suk Choi, Woo-Seop Rhee, “IoT Home Gateway for Auto-Configuration and Management of MQTT Devices”, 2015 IEEE Conference on Wireless Sensors.

[Vögler2015] Michael Vögler, Johannes M. Schleicher, Christian Inzinger, Stefan Nastic, Sanjin Sehic and Schahram Dustdar, “LEONORE – Large-Scale Provisioning of Resource-Constrained IoT Deployments”, 2015 IEEE Symposium on Service-Oriented System Engineering

[Wu2015] Di Wu, Dmitri I. Arkhipov, Eskindir Asmare, Zhijing Qin, Julie A. McCann “UbiFlow: Mobility Management in Urban-scale Software Defined IoT”, INFOCOM 2015, Hong Kong, May 2015.

[Chen2016] Ing-Ray Chen, Jia Guo, and Fenye Bao, ”Trust Management for SOA-Based IoT and Its Application to Service Composition”, IEEE Transactions on Services Computing, Vol. 9, No. 3, May/June 2016.

[European2014] European Research Cluster on the Internet of Things, "EU-China Joint White Paper on Internet-of-Things Identification," 2014.

[Al-Fuqaha2015] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of Things: A Survey on Enabling Technologies, Protocols and Applications. IEEE Commun. Surv. Tutorials PP, 99 (2015), 1–1

[Yijian2013] L. Yijian y J. Raj, «Naming in the Internet of Things,» 2013.

[Sandoche2017] B. Sandoche, «Why DNS should be the naming service for Internet of Things?,» 16 January 2017. [En línea]. Available: https://ant.isi.edu/events/dinr2016/P/p72.pdf.

[IEEEStan2017] EEE Standards Association, «What is an Object Identifier (OID)?,» [En línea]. Available: https://standards.ieee.org/develop/regauth/tut/oid.pdf. [Last access: 16 January 2017].

[SelectHub] SelectHub Webpage: https://selecthub.com/categories/cloud-management-software?page=1

[Whatmatrix] WhatMatrix Webpage: https://www.whatmatrix.com/comparison/Cloud-Management-Platforms#

[Embotics] Embotics vCommander: http://www.embotics.com/embotics-vcommander-hybrid-cloud-management

[VMware] VMware vRealize: http://www.vmware.com/products/vrealize-suite.html

[Rightscale] RightScale: http://www.rightscale.com

[RedHat] RedHat Cloud Forms: http://www.redhat.com/en/technologies/cloud-computing/cloudforms

[ManageIQ] ManageIQ WebPage: http://manageiq.org/

[Oracle] Oracle: https://www.oracle.com/solutions/index.html

[Techbeacon] Techbeacon Webpage: https://techbeacon.com/7-essential-open-source-tools-cloud-management

[OneOps] OneOps: http://oneops.com/

[Docker] Docker: https://blog.docker.com/2014/12/announcing-docker-machine-swarm-and-compose-for-orchestrating-distributed-apps/

[Ansible] Ansible: https://www.ansible.com/

[UniServer] UniServer Consortium, «A Universal Micro-Server Ecosystem by Exceeding the Energy and Performance Scaling Boundaries,» [Online]. Available: http://www.uniserver2020.eu/. [Last access: 30 January 2017].

[PrismTech] PrismTech, «Vortex Fog,» [Online]. Available: http://www.prismtech.com/vortex/vortex-fog. [Last access: 1 February 2017].

[AWS] https://aws.amazon.com/es/greengrass/?nc1=h_ls

[FIWARE] FIWARE, «FIWARE Catalogue,» [Online]. Available: https://catalogue.fiware.org/. [Last Access: 1 February 2017].

[SOFIA] Indra Platform, «Sofia2,» [Online]. Available: http://sofia2.com/home_en.html. [Last access: 1 February 2017

[AllSeen] AllSeen Alliance, «Framework - AllSeen Alliance,» [Online]. Available: https://allseenalliance.org/framework/documentation/learn. [Last access: 1 February 2017].

[IoTivity] IoTivity, «Architecture Overview,» [Online]. Available: https://www.iotivity.org/documentation/architecture-overview. [Last access: 1 February 2017].

[ThreadGroup] Thread Group, «Thread,» 13 July 2015. [Online]. Available: http://www.threadgroup.org/Portals/0/documents/whitepapers/Thread%20Stack%20Fundamentals_v2_public.pdf.

[Last Access: 1 February 2017]

[Aazam2014] M. Aazam, E. N. Huh, Fog Computing and Smart Gateway Based Communication for Cloud of Things, 2014 International Conference on Future Internet of Things and Cloud (FiCloud), August 2014, Barcelona, Spain.

[Willis2014] D. Willis, A. Dasgupta, S. Banerjee, ParaDrop: A Multi-tenant Platform to Dynamically Install Third Party Services On Wireless Gateways, Proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture, MobiArch '14, September 2014, Maui, Hawaii.

[Ismail2015] B. I. Ismail, E. Mostajeran Goortani, M. B. Ab Karim, W. Ming Tat, S. Setapa, J. Y. Luke, O. Hong Hoe, Evaluation of Docker as Edge Computing Platform, 2015 IEEE Conference on Open Systems (ICOS), August 2015, Melaka, Malaysia.

[ZurichUblog] Zurich University of Applied Sciences at https://blog.zhaw.ch/icclab/making-fog-computing-real-research-challenges-in-integrating-localized-computing-nodes-into-the-cloud/

[McKinsey2015] McKinsey Global Institute, The Internet of Things: mapping the value beyond the hype, june 2015

[Forrester2016] Forrester, Simplifying the complexity of IoT, june 2016

[Deloitte2016] Deloitte, Internet of Things Ecosystem: Unlocking the Business Value of Connected Devices, 2016

[BCG2017] Boston Consulting Group, Winning in IoT: It's All About the Business Processes, 2017

[451Research2015] 451 Research, 2016 Trends in Datacenter Technologies, December 2015

[HPE2016] Hewlett Packard Enterprise, How to Get the Most From the Internet of Things, 2016

[PoliMI2016] Politecnico di Milano, Big Data Analytics & Business Intelligence Observatory : final report 2016

[ForresterWav2016] The Forrester Wave : Big Data Fabric, Q4 2016

[Gartner2016] Gartner Magic Quadrant for Data Warehouse and Data Management for Analytics 2016

[CloudTrends2016] Right Scale - Cloud Computing Trends: 2016 State of the Cloud Survey at: http://www.rightscale.com/blog/cloud-industry-insights/cloud-computing-trends-2016-state-cloud-survey#hybridcloudadoption

[DockerCon16] Docker Orchestration: Survey Results from DockerCon16 at: https://platform9.com/blog/docker-orchestration-survey-dockercon16/

[DockerEcosystem] Survey: The State of Containers and the Docker Ecosystem 2015 at: http://containerjournal.com/2015/10/29/survey-the-state-of-containers-and-the-docker-ecosystem-2015/

[VMwareCMP] VMware Cloud Management Platform: http://www.vmware.com/solutions/virtualization/cloud-management.html

[RightScale] Right Scale: http://www.rightscale.com/

[CloudFoundry] Cloud Foundry: https://www.cloudfoundry.org/

[Openshift] Openshift: https://www.openshift.com/

[Heroku] Heroku: https://www.heroku.com/

[OpenStack] OpenStack: https://www.openstack.org/

[Scalr] Scalr: http://www.scalr.com/

[CloudStack] CloudStack: http://cloudstack.apache.org/

[Eucalyptus] Eucalyptus: https://www.eucalyptus.com/ https://github.com/eucalyptus/eucalyptus/wiki

[OpenNebula] OpenNebula: http://opennebula.org/

[VMWareVRealize] VMWare vRealize: http://www.vmware.com/solutions/virtualization/cloud-management.html

[Morpheus] Morpheus: https://www.morpheusdata.com/

[IBMCloudOrch] IBM Cloud Orchestrator: http://www-03.ibm.com/software/products/en/ibm-cloud-orchestrator

[MicrosoftAzure] Microsoft Azure: http://azure.microsoft.com/en-us/documentation/services/automation/

[Flexiant] Flexiant Cloud Orchestrator: http://www.flexiant.com/flexiant-cloud-orchestrator/

[VMwareVOrch] VMware vRealize Orchestrator: http://www.vmware.com/products/vrealize-orchestrator.html

[Docker] Docker: https://www.docker.com/

[Swarm] Docker Swarm: https://www.docker.com/products/docker-swarm

[Marathon] Marathon: https://mesosphere.github.io/marathon/

[Nomad] Nomad: https://www.nomadproject.io/

[EC2ContainerServ] Amazon EC2 Container Service: https://aws.amazon.com/es/ecs/

[AzureContService] Azure Container Service: https://azure.microsoft.com/en-us/services/container-service/

[Kubernetes] Kubernetes: https://kubernetes.io/

[kubectl CLI] kubectl CLI: https://kubernetes.io/docs/user-guide/kubectl-overview/

[GoogleCompEng] Google Compute Engine: https://cloud.google.com/compute/

[ApacheMesos] Apache Mesos: https://mesos.apache.org/

[CoreOS] CoreOS Tectonic: https://coreos.com/tectonic/

[Rocket] Rocket: https://coreos.com/blog/rocket.html

[RocketGithub] Rocket project: https://github.com/coreos/rkt

[RocketKubernetes] Rocket support by Kubernetes: https://rocket.readthedocs.io/en/latest/Documentation/using-rkt-with-kubernetes/

[RocketNomad] Rocket support by Nomad: https://github.com/coreos/rkt/blob/master/Documentation/using-rkt-with-nomad.md

[Singularity] Singularity: http://singularity.lbl.gov/index.html

[RunC] RunC: https://runc.io/

[CityZen] http://cityzen.ch/cityzen_en.html

[HNSciCloud] http://www.helix-nebula.eu

[SlipStream] http://sixsq.com/products/slipstream

[Coburn2011] J. Coburn, A.M. Caulfield, A. Akel, L.M. Grupp, R.K. Gupta, R. Jhala, and S. Swanson, "NV-Heaps: Making Persistent Objects Fast and Safe with Next-Generation, Non-Volatile Memories", Proc. of the 16th Intl. conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS’11), 2011, pp. 105-118.

[Wong2010] H.S. Wong, S. Raoux, S. Kim, J. Liang, J.P. Reifenberg, B. Rajendran, M. Asheghi, and K.E. Goodson, "Phase Change Memory", Proceedings of the IEEE, 98(12), 2010, pp. 2201-2227.

[Queralt2015] A. Queralt, J. Martí, H. Baars, A. Brinkmann, T. Cortes, “Fusing Storage and Computing for the Domain of Business Intelligence and Analytics - Research Opportunities”, Proc. of the 48th Hawaii Intl. Conf. on Systems Sciences (HICSS’15), 2015, pp. 4752-4761.

[NextGenIO] NextGenIO: Next Generation I/O for the Exascale, http://www.nextgenio.eu/.

[BigStorage] BigStorage: Storage-based convergence between HPC and Cloud to handle Big Data, http://bigstorage-project.eu/.

[BIGSEA] EUBra BIGSEA Project, http://www.eubra-bigsea.eu

[Fitch2013] B. G. Fitch, “Exploring the Capabilities of a Massively Scalable, Compute-in-Storage Architecture by Close Integration of Solid State Storage (Flash) into the IBM Blue Gene/Q System”. 22nd Intl ACM Symposium on High-Performance Parallel and Distributed Computing (HPDC’2013), 2013.

[Lustre] Lustre™ File System: High-Performance Storage Architecture and Scalable Cluster File System, White Paper, 2007, http://www.csee.ogi.edu/~zak/cs506-pslc/lustrefilesystem.pdf

[PVFS] PVFS, http://www.pvfs.org

[Shvachko2010] K. Shvachko, H. Kuang, S. Radia, and R. Chansler “The Hadoop Distributed File System”. In Proc. of the IEEE 26th Symposium on Mass Storage Systems and

Technologies (MSST'10), 2010, pp. 1-10.

[Dean2004] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," in 6th Symposium on Operating System Design and Implementation (OSDI 2004), San Francisco, 2004.

[Marti2013] J. Martí, D. Gasull, A. Queralt, T. Cortés. Towards DaaS 2.0: Enriching Data Models. IEEE 2013 International Workshop on Formal Methods in Services and Cloud Computing, (FM-S&C, SERVICES 2013): 349-355, 2013.

[Marti2017] J. Martí, A. Queralt, D. Gasull, A. Barcelo, J.J. Costa, T. Cortes, "dataClay: a distributed data store for effective inter-player data sharing", Journal of Systems and Software (under review).

[Li2015] S. Li, H. Lim, V.W. Lee, J.H. Ahn, A. Kalia, M. Kaminsky, D.G. Andersen, O. Seongil, S. Lee, and P. Dubey "Architecting to achieve a billion requests per second throughput on a single key-value store server platform", In Proc. of the 42nd Annual Intl. Symposium on Computer Architecture (ISCA '15), 2015, pp. 476-488.

[George2011] L. George, "HBase: the definitive guide", O'Reilly Media, Inc., 2011.

[Lakshman2010] A. Lakshman and P. Malik. Cassandra: a decentralized structured storage system. ACM SIGOPS Operating Systems Review, 44(2), 2010.

[Olson1999] M.A. Olson, K. Bostic, and M.I. Seltzer, "Berkeley DB", in Proceedings of the USENIX Annual Technical Conference (USENIX '99), 1999, 183-191.

[Memcached] Memcached, https://memcached.org/.

[Bradshaw2016] S. Bradshaw and K. Chodorow, "MongoDB: the definitive guide", O'Reilly Media, Inc., 2016.

[CouchDB] CouchDB, http://couchdb.apache.org/.

[Robinson2013] I. Robinson, J. Webber, E. Eifrem, "Graph databases", O'Reilly Media, Inc., 2013.

[Neo4j] Neo4j, https://neo4j.com/.

[Gonzalez2014] J.E. Gonzalez, R.S. Xin, A. Dave, D. Crankshaw, M.J. Franklin, I. Stoica, "GraphX: Graph Processing in a Distributed Dataflow Framework", 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI'14), 2014, pp. 599-613.

[OpenCL] Open CL https://www.khronos.org/opencl/

[OpenACC] Open ACC http://www.openacc.org/

[OmpSs@GPU] Bueno, J., Planas, J., Duran, A., Badia, R. M., Martorell, X., Ayguade, E., & Labarta, J. (2012, May). Productive programming of GPU clusters with OmpSs. In *Parallel & Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International* (pp. 557-568). IEEE.

[OmpSs@FPGA] Filgueras, A., Gil, E., Jimenez-Gonzalez, D., Alvarez, C., Martorell, X., Langer, J., and Vissers, K. (2014, February). OmpSs@ Zynq all-programmable SoC ecosystem. In *Proceedings of the 2014 ACM/SIGDA international symposium on*

Field-programmable gate arrays (pp. 137-146). ACM.

[COMPSs] Badia, R. M., Conejero, J., Diaz, C., Ejarque, J., Lezzi, D., Lordan, F., ... & Sirvent, R. (2015). COMP Superscalar, an interoperable programming framework. *SoftwareX*, *3*, 32-36.

[TANGO] Djemame, K., Armstrong, D., Kavanagh, R., Deprez, J. C., Ferrer, A. J., Perez, D. G., ... & Georgiou, Y. (2016). TANGO: Transparent heterogeneous hardware Architecture deployment for eNergy Gain in Operation. PROHA Workshop, *arXiv preprint arXiv:1603.01407*.

[Vasilomanolakis2015] Vasilomanolakis, E., Daubert, J., Luthra, M., Gazis, V., Wiesmaier, A., Kikiras, P. (2015). On the Security and Privacy of Internet of Things Architectures and Systems. In *Proceedings of the 2015 International Workshop on Secure Internet of Things (SIoT)*. IEEE.

[Cam-Winget2016] Cam-Winget, N., Sadeghi, A.-R., Jin, Y. (2016). INVITED: Can IoT be Secured: Emerging Challenges in Connecting the Unconnected. In *Proceedings of the 53rd Annual Design Automation Conference (DAC'16)*. IEEE.

[Bertino2016] Bertino, E. (2016). Data Security and Privacy in the IoT. in *Proceedings of the 19th International Conference on Extending Database Technology (EDBT'16)*.

[Hwang2015] Hwang, Y. H. (2015). IoT Security & Privacy: Threats and Challenges. *In Proceedings of the 1st ACM Workshop on IoT Privacy, Trust, and Security (IoTPTS '15)*. ACM.

[Medwed2016] Medwed, M. (2016). IoT Security Challenges and Ways Forward. In *Proceedings of the 6th International Workshop on Trustworthy Embedded Devices (TrustED '16)*. ACM.

[Kennell2003] Kennell, R., Jamieson, L. H. (2003). Establishing the genuinity of remote computer systems. In *Proceedings of the 12th USENIX Security Symposium*.

[Seshadri2011] Seshadri, A., Luk, M., Perrig, A. (2011). SAKE: Software Attestation for Key Establishment in Sensor Networks. *Ad Hoc Networks* 9(6), pp. 1059–1067.

[Li2011] Li, Y., McCune, J. M., Perri, A. (2011). VIPER: Verifying the Integrity of PERipherals' Firmware. In Proceedings of the 18th ACM conference on Computer and communications security (CCS'11), pp. 3-16. ACM.

[Castelluccia2009] Castelluccia, C., Francillon, A., Perito, D., Soriente, C. (2009). On the Difficulty of Software-Based Attestation of Embedded Devices. In *Proceedings of the 16th ACM conference on Computer and communications security (CCS'09)*, pp. 400-409. ACM.

[Kovah2012] Kovah, X., Kallenberg, C., Weathers, C., Herzog, A., Albin, M., Butterworth, J. (2012). New Results for Timing-Based Attestation. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy (SP'12)*. pp. 239-253. IEEE.

[Wurster2005] Wurster, G., Van Oorschot, P., Somayaji, A. (2005). A generic attack on checksumming-based software tamper resistance. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy (SP'05)*. pp. 127-138. IEEE.

[Arbaugh1997] Arbaugh, W. A., Farber, D. J., Smith, J. M. (1997). A secure and reliable bootstrap architecture. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy (SP'97)*. pp. 65, IEEE.

[Pearson2005] Pearson, S., Mont, M. C., Crane, S. (2005). Persistent and Dynamic Trust: Analysis and the Related Impact of Trusted Platforms. In *Proceedings of the 3rd International Conference on Trust Management (iTrust'05)*, pp. 355-363, Springer.

[Kil2009] Kil, C., Sezer, E. C., Azab, A. M., Ning, P., Zhang, X. (2009). Remote Attestation to Dynamic System Properties: Towards Providing Complete System Integrity Evidence. In *Proceedings of the IEEE/IFIP International Conference on Dependable Systems & Networks (DSN'09)*. IEEE.

[Datta2009] Datta, A., Franklin, J., Garg, D., Kaynar, D. (2009). A Logic of Secure Systems and its Application to Trusted Computing. In *Proceedings of the 30th IEEE Symposium on Security and Privacy (SP'09)*, pp. 221-236. IEEE.

[TPM] Trusted Platform Module (TPM) Summary, https://trustedcomputinggroup.org/trusted-platform-module-tpm-summary/

[Schiffman2012] Schiffman, J., Vijayakumar, H., Jaeger, T. (2012). Verifying System Integrity by Proxy. In *Proceedings of the 5th International Conference on Trust and Trustworthy Computing*, pp. 179–201. Springer.

[Schiffman2010] Schiffman, J., Moyer, T., Vijayakumar, H., Jaeger, T., McDaniel, P. (2010). Seeding clouds with trust anchors. In *Proceedings of the 2010 ACM Workshop on Cloud Computing Security Workshop (CCSW'10)*, pp. 43–46. ACM.

[Schiffman2011] Schiffman, J., Moyer, T., Jaeger, T., McDaniel, P. (2011). Network-based Root of Trust for Installation," IEEE Security and Privacy 9(1), pp. 40-48. IEEE.

[Berger2015] Berger, S., Goldman, K., Pendarakis, D., Safford, D., Valdez, E., Zohar, M. (2015). Scalable Attestation: A Step Toward Secure and Trusted Clouds. *In Proceedings of the 2015 IEEE International Conference on Cloud Engineering (IC2E'15)*, 2015. IEEE.

[Stefan2012] Steffan, A. (2012). The Linux Integrity Subsystem and TPM-based Network: Endpoint Assessment. In *Proceedings of the Linux Security Summit*.

[Williams2011] Williams, P., Boivie, R. (2011). CPU Support for Secure Executables. In *Proceedings of the 4th International Conference on Trust and Trustworthy Computing (TRUST'11)*, pp. 172-187. Springer.

[McKeen2013] McKeen, F., Alexandrovich, I., Berenzon, A., Rozas, C. V., Shafi, H., Shanbhogue, V., Savagaonkar, U. R. (2013). Innovative Instructions and Software Model for Isolated Execution. In *Proceedings of the 2nd International Workshop on Hardware and Architectural Support for Security and Privacy*. p. 10. ACM.

[Lee2005] Lee, R. B., Kwan, P. C., McGregor, J. P., Dwoskin, J., Wang, Z. (2005). Architecture for Protecting Critical Secrets in Microprocessors. In *Proceedings of the 32nd International Symposium on Computer Architecture (ISCA'05)*. pp. 2–13. IEEE.

[ARM] ARM TrustZone. https://developer.arm.com/technologies/trustzone

[KINIBI] Trustonic Kinibi Trusted Execution Environment (TEE).

https://www.trustonic.com/products/kinibi.

[Intel2012] Intel (2012). Intel Trusted Execution Technology - Hardware-based Technology for Enhancing Server Platform Security. http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/trusted-execution-technology-security-paper.pdf

[AMD2005] AMD (2005). Secure Virtual Machine Architecture Reference Manual. https://www.mimuw.edu.pl/~vincent/lecture6/sources/amd-pacifica-specification.pdf

[McCune2010] McCune, J. M., Li, Y., Qu, N., Zhou, Z., Datta, A., Gligor, V., Perrig, A. (2010). TrustVisor: Efficient TCB Reduction and Attestation. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy (SP'10)*. pp. 143-158. IEEE.

[Nie2007] Nie, C. (2007). Dynamic Root of Trust in Trusted Computing. TKK Seminar on Network Security. http://www.tml.tkk.fi/Publications/C/25/papers/Nie_final.pdf

[Parno2010] Parno, B. J., McCune, J. M., Perrig, A. (2010). Bootstrapping Trust in Commodity Computers. In *Proceedings of the 2010 IEEE Symposium on Security and Privacy (SP'10)*. pp. 414-429. IEEE.

[ElDefrawy2012] El Defrawy, K., Francillon, A., Perito, D., Tsudik, G. (2012). SMART: Secure and minimal architecture for (establishing a dynamic) root of trust. In *Proceedings of the Network and Distributed System Security Symposium (NDSS'12)*.

[Koeberl2014] Koeberl, P., Schulz, S., Sadeghi, A.-R., Varadharajan, V. (2014). TrustLite: a security architecture for tiny embedded devices. In *Proceedings of the 9th European Conference on Computer Systems (EuroSys'14)*. ACM.

[Brasser2015] Brasser, F., El Mahjoub, B., Sadeghi, A.-R., Wachsmann, C., Koeberl, P. (2015). TyTAN: tiny trust anchor for tiny devices. In *Proceedings of the 52nd Annual Design Automation Conference (DAC'15)*. ACM.

[Asokan2015] Asokan, N., Brasser, F., Ibrahim, A., Sadeghi, A.-R., Schunter, M., Tsudik, G., Wachsmann, C. (2015). SEDA: Scalable Embedded Device Attestation. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)*. pp. 964-975. ACM.

[Francillon2014] Francillon, A., Nguyen, Q., Rasmussen, K. B., Tsudik, G. (2014). A Minimalist Approach to Remote Attestation. in *Proceedings of the conference on Design, Automation & Test in Europe (DATE'14)*. EDAA.

[Ibrahim2016] Ibrahim, A., Sadeghi, A.-R., Tsudik, G., Zeitouni, S. (2016). DARPA: Device Attestation Resilient to Physical Attacks. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks (WiSec'16)*. pp. 171-182. ACM.

[Adelsbach2005] Adelsbach, A., Huber, U., Sadeghi, A.-R. (2005). Secure Software Delivery and Installation in Embedded Systems. In: Deng, R. H., Bao, F., Pang, H., Zhou, J. (eds.). Information Security Practice and Experience (LNCS, vol. 3439). pp. 255-267. Springer.

[Misra2013] Misra, S., Tourani, R., Majd, N. E. (2013). Secure content delivery in information-centric networks: design, implementation, and analyses. *In Proceedings of the 3rd ACM SIGCOMM Workshop on Information-centric networking (ICN'13)*. pp. 73-78. ACM.

[Ambrosin2014] Ambrosin, M., Busold, C., Conti, M., Sadeghi, A.-R., Schunter, M. (2014). Updaticator: Updating Billions of Devices by an Efficient, Scalable and Secure Software Update Distribution Over Untrusted Cache-enabled Networks. In *Proceedings of the 19th European Symposium on Research in Computer Security*. pp. 76-94. Springer.

[SWUpdate] SWUpdate. https://github.com/sbabic/swupdate

[RAUC] RAUC. https://github.com/rauc/rauc

[Mender] Mender. https://docs.mender.io/.

[Resin] Resin. https://resin.io/

[Miettinen2016] Miettinen, M., Sadeghi, A.-R., Marchal, S., Asokan, N., Hafeez, I., Tarkoma, S. (2016). IOT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. https://arxiv.org/abs/1611.04880.

[Oltisk2014] Oltsik, J. (2014). The Internet of Things: A CISO and Network Security Perspective.https://pdfs.semanticscholar.org/e0d5/1ed03f09ded7eab813fba45c08bc1bf3800c.pdf.

[Mahalle2013] Mahalle, P. N., Anggorojati, B., Prasad, N. R., Prasad, R. (2013). Identity Authentication and Capability Based Access Control (IACAC) for the Internet of Things. Journal of Cyber Security and Mobility 1(4). pp. 309–348. River Publishers.

[Yao2013] Yao X., Han X., Du X., Zhou X. (2013). A lightweight multicast authentication mechanism for small scale IoT applications. IEEE Sensors Journal 13(10). pp. 693-701. IEEE.

[Flood2014] Flood, P. (2014). Securing the internet of things – A ZKP based approach. http://www.osna-solutions.com/wp-content/uploads/Securing-the-Internet-of-Things-A-ZKP-Approach-Thesis-Extract.pdf.

[Alpar2016] Alpar, G., Batina, L., Batten, L., Moonsamy, V., Krasnova, A., Guellier, A., Natgunanathan, I. (2016). New directions in IoT privacy using attribute-based authentication. In *Proceedings of the ACM International Conference on Computing Frontiers*. pp. 461 – 466. ACM.

[Sullivan2015] Sullivan, N. (2015). How to build your own public key infrastructure. https://blog.cloudflare.com/how-to-build-your-own-public-key-infrastructure/.

[McGrew2010] McGrew, D., Rescorla, E. (2010). Datagram Transport Layer Security (DTLS) Extension to Establish Keys for the Secure Real-time Transport Protocol (SRTP). IETF.

[Keoh2014] Keoh, S. L., Kumar, S., Tschofenig, H. (2014). Securing the Internet of Things: A Standardization Perspective. Internet of Things Journal 1(3). pp. 265–275. IEEE

[Urien2013] Urien, P. (2013). LLCPS: A new security framework based on TLS for NFC P2P applications in the Internet of Things. In *Proceedings of the 2013 IEEE Consumer Communications and Networking Conference (CCNC'13)*. IEEE.

[CloudStrategy] European Cloud Strategy 2012. https://ec.europa.eu/digital-single-market/node/10565

[CloudPartnership] European Cloud Partnership. https://ec.europa.eu/digital-single-market/node/609

[DSM] Digital Single Market. https://ec.europa.eu/commission/priorities/digital-single-market_en

[Gartner2013] Gartner 2013. “Private Cloud Matures, Hybrid Cloud is Next”.

[Gartner2016] Gartner 2016. “Market Insight: Cloud Computing’s Drive to Digital Business Creates Opportunities for Providers.”

[Statista2015] The Statistics Portal. “Public Cloud Infrastructure as a Service (IaaS) hardware and software spending from 2015 to 2026, by segment (in billion U.S. dollars). https://www.statista.com/statistics/507952/worldwide-public-cloud-infrastructure-hardware-and-software-spending-by-segment/

[RightScale2017] RightScale 2017. “State of the Cloud Report”. http://assets.rightscale.com/uploads/pdfs/RightScale-2017-State-of-the-Cloud-Report.pdf

[EuroCloud] EuroCloud Europe. https://www.eurocloud.org/about.html

[Gartner10] Gartner’s Top 10 Strategic Technology Trends for 2017. http://www.gartner.com/smarterwithgartner/gartners-top-10-technology-trends-2017/

[Enescu2014] Michael Enescu. “From Cloud to Fog & The Internet of Things”. https://cdn.oreillystatic.com/en/assets/1/event/115/From%20Cloud%20to%20Fog%20Computing%20and%20the%20Internet%20of%20Things%20Presentation.pdf

[GartnerTop] Gartner. “Top Strategic Predictions for 2017 and Beyond: Surviving the Storm Winds of Digital Disruption”.

[GartnerDecade] Gartner. “Predicts 2017: Cloud Computing Enters Its Second Decade”.

[Forrester2017] Forrester. “Predictions 2017: Customer-Obsessed Enterprises Launch Cloud’s Second Decade”.

[DimensionData] Dimension Data. “Top IT trends in 2017: digital business”. http://www2.dimensiondata.com/en/IT-Trends/Digital-business

[GartnerDigital] Gartner. “Architect Digital Platforms to Deliver Business Value and Outcomes”.

[InformationAge] Information Age. “Predictions for cloud in 2017: Unlocking creativity with platform thinking”. http://www.information-age.com/unlocking-creativity-platform-thinking-123463663/

[Forbes2017] Forbes. “Top 10 Trends For Digital Transformation in 2017”. https://www.forbes.com/sites/danielnewman/2016/08/30/top-10-trends-for-digital-transformation-in-2017/#6f9444fd47a5

[ISO/IEC JTC1] https://www.iso.org/isoiec-jtc-1.html

[ISO/IEC JTC1 http://www.iso.org/iso/iso_technical_committee%3Fcommid%3D601355

SC38]

[DMTF] https://www.dmtf.org

[CIMI] https://www.dmtf.org/standards/cloud

[REDFISH] https://www.dmtf.org/standards/redfish

[SNIA] https://www.snia.org/

[OGF] https://www.ogf.org

[OCCI] http://occi-wg.org/

[ETSI] http://www.etsi.org/

[ETSI-CSC] http://csc.etsi.org/

[EC C-SIG] https://ec.europa.eu/digital-single-market/en/cloud-computing-strategy-working-groups

[OFC] https://www.openfogconsortium.org/

[OFC-RA] https://www.openfogconsortium.org/ra/

[OCF] https://openconnectivity.org/

[OCF-OIC] https://openconnectivity.org/resources/specifications

[OCF-CPR] https://openconnectivity.org/certified-product

[LoRA] https://www.lora-alliance.org/

[WIKI-EDGE] https://en.wikipedia.org/wiki/Edge_computing

[EDGE-MARKET] http://uk.businessinsider.com/edge-computing-is-the-next-multi-billion-tech-market-2016-12

[BRCK] https://www.brck.com/

[OEC] http://openedgecomputing.org

[FOG-V-EDGE] https://www.automationworld.com/fog-computing-vs-edge-computing-whats-difference

[C-SERIES] http://aviationweek.com/connected-aerospace/internet-aircraft-things-industry-set-be-transformed

[EdgeNetwork] http://www.networkworld.com/article/3148330/virtualization/will-the-real-edge-computing-please-stand-up.html

[Orbiwise] https://www.orbiwise.com

[HPE-EDGE] https://www.hpe.com/us/en/servers/edgeline-iot-systems.html

[CONT-EDGE] http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.726.832&rep=rep1&type=pdf

[Edge-IoT] http://www.govtech.com/transportation/is-edge-computing-key-to-the-internet-of-things.html

[FaaS] http://www.cbronline.com/news/internet-of-things/5-future-tech-trends-to-watch-out-for-serverless-computing-platform-economy-device-mesh-edge-analytics-and-ambient-user-experience-4949014/#5-future-tech-trends-to-watch-out-for-serverless-computing-platform-economy-device-mesh-edge-analytics-and-ambient-user-experience-4949014

[NIST] https://www.nist.gov/programs-projects/cloud-computing

[Masip2016] Masip-Bruin, X., Marín-Tordera, E., Tashakor, G., Jukan, A., & Ren, G. J. (2016). Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. *IEEE Wireless Communications*, *23*(5), 120-128.

[Wang2015] Wang, X., Li, X., & Leung, V. C. (2015). Artificial intelligence-based techniques for emerging heterogeneous network: State of the Arts, Opportunities, and Challenges. *IEEE Access*, *3*, 1379-1391.

[Kephart2003] Kephart, J. O., & Chess, D. M. (2003). The vision of autonomic computing. *Computer*, *36*(1), 41-50.

[Pop2016] Pop, D. (2016). Machine learning and cloud computing: Survey of distributed and saas solutions. *arXiv preprint arXiv:1603.08767*.

[Aazam2014] Aazam, M., & Huh, E. N. (2014, August). Fog computing and smart gateway based communication for cloud of things. In *Future Internet of Things and Cloud (FiCloud), 2014 International Conference on* (pp. 464-470). IEEE.

[Abdelkhalek2011] Abdelkhalek, O., Krichen, S., Guitouni, A., & Mitrovic-Minic, S. (2011, October). A genetic algorithm for a multi-objective nodes placement problem in heterogeneous network infrastructure for surveillance applications. In *Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP* (pp. 1-9). IEEE.

[Bengio1999] Bengio, Y. (1999). Markovian models for sequential data. *Neural computing surveys*, *2*(199), 129-162.

[Hadoop] Apache Hadoop. Web page at http://hadoop.apache.org/, (Date of last access: 15th November, 2016).

[Spark] M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster Computing with Working Sets. In Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing, HotCloud'10, Berkeley, CA, USA, 2010. USENIX Association.

[PaNDaaS] http://pan-data.eu/node/103

[abiresearch2016] Edge analytics in IoT, https://www.abiresearch.com/market-research/product/1021642-edge-analytics-in-iot/

[businsider] Edge computing in IoT, https://www.businessinsider.com/intelligence/research-store?IR=T&utm_source=businessinsider&utm_medium=report_teaser&utm_term=report_teaser_store_text_link_edge-computing-in-the-iot-forecasts-key-benefits-and-top-industries-adopting-an-analytics-model-that-improves-processing-and-cuts-costs-2016-7&utm_content=report_store_report_teaser_text_link&utm_campaign=report_teaser_store_link&vertical=IoT#!/Edge-Computing-in-the-IoT/p/68220396/

[embedcompute2017] 2017 IIoT Prediction: Edge computing goes mainstream, http://embedded-computing.com/guest-blogs/2017-iiot-prediction-edge-computing-goes-mainstream/

[GartnerHype2017 Gartner's 2016 Hype Cycle for Emerging Technologies Identifies Three Key

]	Trends That Organizations Must Track to Gain Competitive Advantage, http://www.gartner.com/newsroom/id/3412017
[GartnerStrategies 2016]	Gartner, “Hype Cycle for Infrastructure Strategies, 2016”.
[GartnerEdge2016]	Gartner, “The Edge Manifesto: Digital Business, Rich Media, Latency Sensitivity and the Use of Distributed Data Centers”
[idc2017]	IDC FutureScape: Worlwide Datacenter 2016 Predictions
[moorinsightsstrategy2015]	Moor Insights & Strategy, “Bringing Intelligence to the Cloud Edge”, http://www.moorinsightsstrategy.com/wp-content/uploads/2015/02/Bringing-Intelligence-To-Cloud-Edge-by-Moor-Insights-and-Strategy.pdf