Open Source Software solutions implementing a reference IoT architecture from the Things and Edge to the Cloud

Technical Report · May 2018

DOI: 10.13140/RG.2.2.12214.60488/1

Christos S. Tranoris
University of Patras

Network Architectures and Management Group
Electrical and Computing Engineering Department,
University of Patras,
Patras, Greece

Technical Report

Open Source Software solutions implementing a reference IoT architecture from the Things and Edge to the Cloud

May 2018

Authors

Christos Tranoris
tranoris@ece.upatras.gr (http://nam.ece.upatras.gr/tranoris)

Spyros Denazis
sdena@upatras.gr (http://nam.ece.upatras.gr/denazis)

Contents

1 Introduction

According to Gartner Hype Cycle 2017 [1], IoT Platforms (among other technologies like 5G, Digital Twins, Edge Computing) are one of the key mainstream technologies to create transformative impact in the next 2 to 5 years. Although the purpose of any IoT device, a “Thing”, is to connect with other IoT devices and applications (cloud-based mostly), the gap between the device sensors and applications is filled by an IoT Platform. Such a platform connects to the “Things” via a data network and provides backend services to support applications enabled from the data generated by hundreds of sensors.

Both Cloud and IoT technologies are now mature enough and widely adopted on various vertical domains such as Industry 4.0, Automotive, Smart Cities, eHealth etc. Edge computing lately emerged to cover near the field problems such as real-time processing and low latency. IoT, Edge and Cloud Computing technologies need to be delivered together in order to provide holistic solutions for these vertical domains having “things” for the sensing/actuating part, edge for local processing, data pre-processing and the cloud for application/business logic, data aggregation, big-data storage and analytics, managing the infrastructure etc.

Usually an IoT platform is considered to run on the cloud, but the picture is now more complete with 5G and Edge Computing. 5G will be a major IoT enabler given the well-publicized 5G promises of ultra-low latencies, 100% coverage, high capacity and five nines reliability. On the other hand (Mobile) Edge Computing arrived to raise the limitations of the centralized cloud computing architecture like ultra-short latency. Edge Computing aims to place storage and computation resources at the network edge, while data processing can be pushed from cloud to the edge, thus reducing the traffic bottleneck toward the core network. Besides, it helps shorten end-to-end latency, it enables the computation offload from power-constrained “Things” to the edge and it represents a fault resilient solution for its decentralized architecture [2].

This report is a current snapshot of available platforms, efforts and solutions based purely on Open Source Software that can be used to deploy an end-to-end IoT solution covering the three tiers of “Things”, the Edge and the Cloud. While there is a wide literature, technologies and market analysis available surveying IoT, the Edge and Cloud separately or combined, this work focuses on how OSS can be delivered to provide complete solutions and identify missing gaps and potential opportunities. To accomplish this, we define a high level architectural pattern and process, which tries to capture all the core functionalities of IoT solutions. This reference architecture is defined after analyzing several architectural IoT solutions and processes. Then we survey OSS examining features, maturity, community support, licensing models etc. by mapping OSS platforms and components that can implement all the parts and processes of this reference architecture.

1.1 The need for OSS IoT Cloud solutions

Open Hardware innovations, like the Raspberry Pi, Arduino, etc, are making it easier, faster and cheaper, even for individuals, to develop new devices for the following reasons:

- Interoperability by sharing source code
- Industry specific capabilities
- Benefit in general from OSS (user code reviews, improvements, etc)
- Based on Open Standards
- Loosely coupled and Modular via APIs
- Platform Independent: available to run in Public clouds (Amazon, Azure, etc) as well as in private cloud or in the case of Edge able to run in multiple microprocessors

A study of [3] claims that once an AWS IoT-based system processes beyond 17-18 messages per second, it’s time to start thinking about moving to another IoT platform to reduce operating cost. The cost per message drops dramatically after 20 messages per second for open-source IoT platform and continues to decrease with the growing message rate, while remaining constant for serverless IoT.

1.2 The research methodology adopted in this report

There is a plethora of open source projects regarding IoT, from the Industry to just individuals. Contributions can be also found by many national or international funding efforts like the EU Horizon 2020. In Github for example the keyword IoT has around 39000 repositories, IoT-Edge around 200 repositories and Edge computing 145.

In this report we setup a few criteria to examine only those OSS solutions that fulfill them. These are:

- Active project: the project needs to be active, and have a clear roadmap (at least at the moment of writing this report)
- Community support: there should be a group of contributors or at least a company or an organization that actively supports it
- Open source code: it should be clear how to get the source code of the solution
- Clear licensing scheme: the solution should state how it can be used, even in commercial projects
- Be Interoperable and not bonded to specific close source solutions like Azure Iot Edge [4] or Amazon Greengrass which are tightly coupled to their IoT cloud solution

2 A reference architecture and processes

The topics of this section are extensively covered by IoT and Cloud literature. However, it will help to setup the problem area and later on to map the OSS solutions and their features. By examining a non-exhaustive list of various reports [5, p. 12], papers [6] and reference architectures [7] [8] [9, pp. 39, 53] [10] [11] [12] , we are able to define: i) a generic data/actions process from "things" and edge to the cloud and ii) a typical 3-tier architectural pattern of "things"-edge-cloud for IoT solutions with the essential elements of each tier.

In Fig 1 we provide a high level process involving the three tiers of "Things"-Edge-Cloud. This process is very generic and try to highlight the activities involved from when data are availble from a sensor up to a cloud application and vice-versa to sensors. Although, the defined activities are not all mandatory they need to be considered by architects deploying an IoT solution. Also the placing of the activities either on the edge or the cloud is based on the needs of the target vertical application. It is the authors opinion based on existing reports and technologies the architectural placement of these activities to the edge or to the cloud.

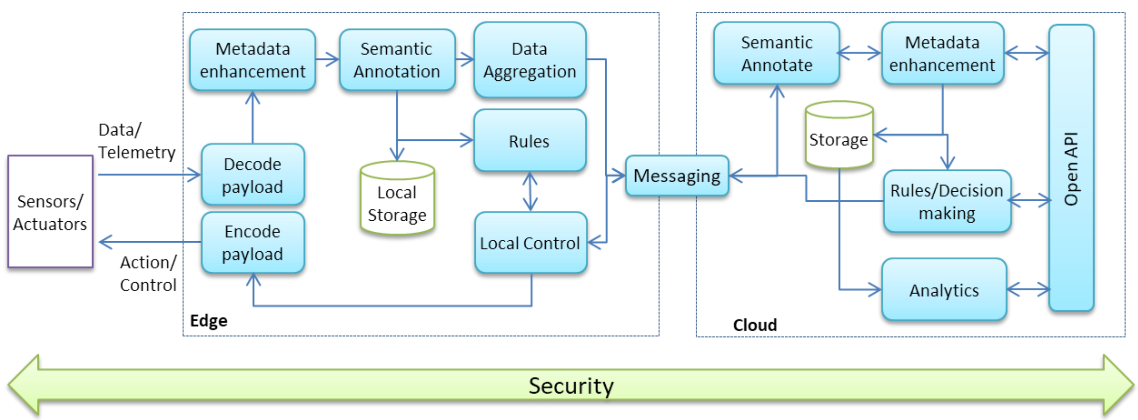

FIG 1 A GENERIC DATA/ACTIONS PROCESS FROM "THINGS", EDGE TO THE CLOUD

As Fig 1 displays, when telemetry data are available from a sensor towards the edge in most cases a needed to transform (see “Decode payload”) received data payload into typed values; e.g. temperature/humidity values from a sensor are transmitted in 2 bytes, the 1st byte need to be decoded into a Small Integer (representing a temperature value) and the 2$^{nd\ t}$ byte need to be decoded into a Small Integer (representing a humidity value); for actuators on the other side there is the need to encode for example (see “Encode payload”) a Boolean true or false value to open a valve to a bit. Of course, the activities “Decode payload” and “Encode payload” might involve security aspects, e.g. decrypt/encrypt values, MD5 signatures, etc.

The activity “Metadata enhancement”, might involve other metadata attached to the payload like gateway information, timestamp, or GPS location of the sensor. E.g. a simple outdoor temperature sensor does not have GPS location, so where it is located is hold as information in the metadata of the gateway related to that sensor. “Semantic annotation” might involve transforming data to an equivalent model following some ontology models and standards. Semantic annotated data might be also store to reduce latency and then pushed for further processing by the "Rules Activity" and "Local Control" for real-time control, processing and decisions. "Data aggregation" activity could be useful for offline-storage, local Analytics and for forwarding aggregated sensor data in a consistent manner to/from the Cloud.

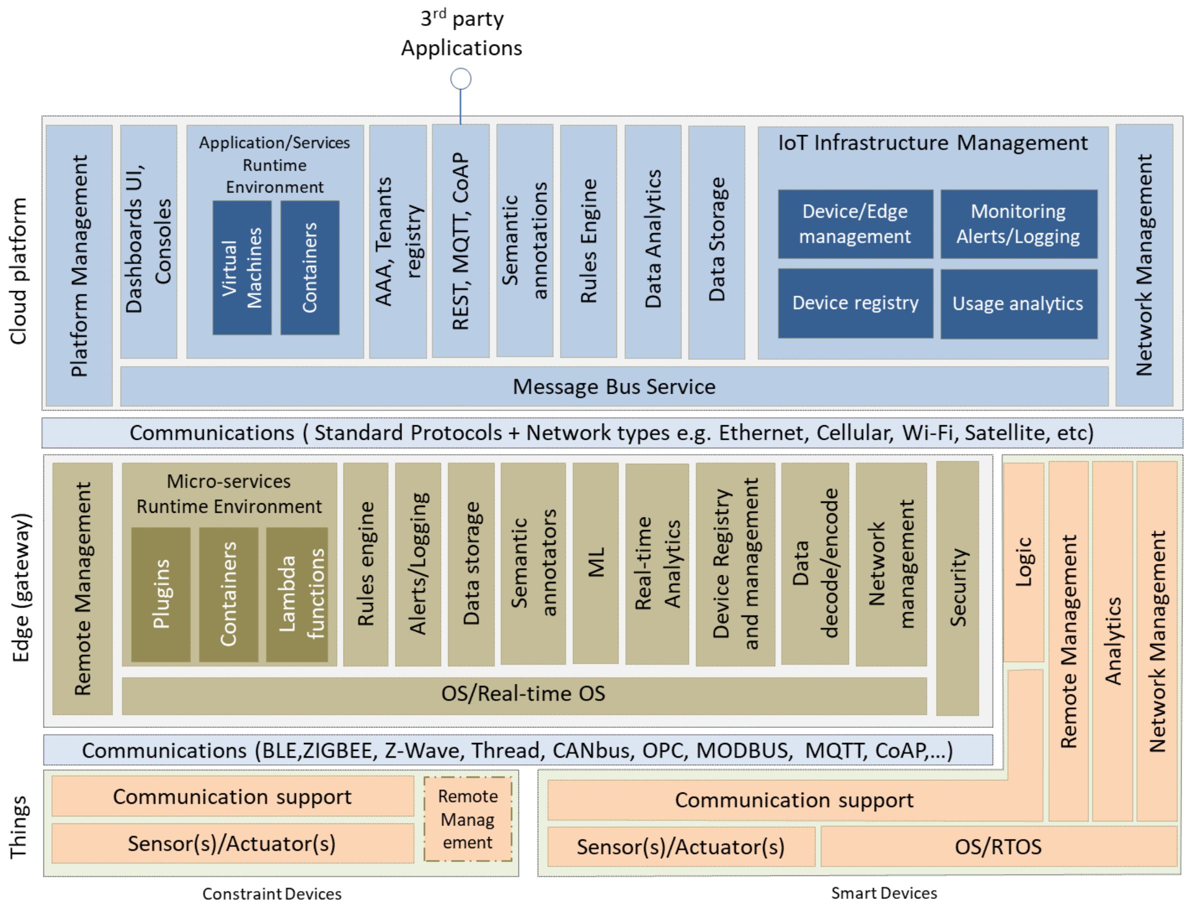

FIG 2 AN IOT 3-TIER ARCHITECTURAL PATTERN INVOLVING THINGS-EDGE-CLOUD

Messaging and connectivity is used to exchange data between the Edge devices and the Cloud. Cloud maintains all the IoT infrastructure management and hosts several services that can process the data received from a large number of Edge devices and "Things". Again "Semantic Annotation" and Metadata Enhancements might perform on received data. All data of the IoT infrastructure are stored to provide global Analytics and perform the necessary Decision making. All data are available to Applications via an Open API.

Finally, security is an aspect that should cover the data processing end-to-end. Not only the APIs and the connectivity should be secure, but the processing of the data inside the service must be done in a secure way. Multi-tenancy in the edge should be an aspect to be studied in future. Especially combined with the concepts of Fog computing. There could be use-cases, for example, that it will not be possible for the Edge to encrypt/decrypt and know the value and the type of the payload. This would be the case if an Edge is used by multiple tenants. E.g. an Edge in a street road needs to process and send to the cloud data to two different companies.

The described basic process that needs to be supported by an IoT solution can be implemented by a 3-tier architectural pattern of "things"-edge-cloud. This is depicted in Fig 2 together with the essential components of each tier which will help as later to identify OSS that implements them. Next sections describe this 3-tier architectural pattern

2.1 "Things"

"Things" or devices measure and enable the interaction with the physical world. We distinguish two categories, as Fig 2 displays:

- **Constraint devices:** Usually based on a micro-controller, contain some sensors or actuators and implement usually a single wired or wireless protocol. Rarely can offer a remote management system for limited configuration or firmware upgrade.
- **Smart Devices:** Based on micro-controllers or micro-processors typically more powerful than a constraint device, usually have an embedded or real time OS. May include sensors and/or actuators. May include remote management features; allow some minimal logic, device analytics and network management and configuration especially if needed to be connected to network types such as a cellular network.

2.2 IoT Edge Computing

Edge computing has gained a lot of attention lately due to the need of local processing or data aggregation by many IoT data streams, or data filtering, since processing at the Edge can respond quickly to local events, operate with intermittent connections, process locally a huge amount of data, provide compute power and storage in the space between the device and the cloud. and transmit only data that are worth of back to the cloud thus minimize the cost of transmitting IoT data to the cloud. Edge computing is needed for local loop scenarios (e.g. home gateway, managing the lights of a street) and real time processing (e.g. image processing). A forecast, cited by Huawei, reckons more than 50 billion terminals and devices will be connected by 2020. In addition, over 50% of data is expected to be analyzed, processed, and stored at the network edge in the future [13] . Edge Computing uses processing power of gateway-devices to filter, pre-process and aggregate IoT data. It might be able to run complex analytics on those data and, in a feedback loop, support decisions and actions about and on the physical world. Edge compute devices include IoT gateways, routers, and micro data centers in mobile network base stations, on the shop floor and in vehicles, among other places. Most applications in the areas like Industry 4.0, Virtual Reality and Smart Cities are data intensive or time sensitive and depend on a lot of data from sensors and devices being processed almost in real time. Furthermore, depending on the application, information must be extracted from the data and transmitted securely. Moreover, the concept of "Edge Intelligence" emerged in [9], a combination of Edge Computing, sophisticated Application Function and Network Infrastructure management powered by Machine Learning. Having solutions coupled with the new 5G network infrastructure, which provides the right software-based network for each application, in future, a central cloud will be obsolete for verticals with the need for robust and secure communication and low latency. In terms of hardware there are many devices that can be used as IoT gateways supporting many Operating Systems [14], from high-end costly solutions like the Dell Edge Gateway 5000 Series until cost-effective solutions like Raspbery-PI.

2.2.1 The essential components of an IoT Edge

Referring to Fig 2 the following components and services are identified at the Edge:

- Embedded OS/Realtime OS: the OS of the Edge device
- Remote Management: Services that allow to remotely manage the host services of the Edge, as well as monitor the Edge, perform remote software deployment and updates
  - An Identity registry of allowed tenants should be present
- Micro-services Runtime environment: an environment to run user/application code, provide functionality isolation in the form of services possibly written in any programming language, preferably updated in dynamic ways. Candidate implementations can be based on:
  - Plugins: e.g. implementations around OSGI the Java modular framework [15], Service Registries like consul [16]
  - Containers: virtualization on the OS level with little overhead. Examples are Docker [17], Linux Containers LXD [18] and rkt of CoreOS [19]
  - Lambda functions: executable code when needed (term found e.g. in AWS Greengrass [20]). Can be thought of as instantaneous instantiation and termination of containers in a scalable manner
- A rules engine: needed for action management and performing local logic at the edge
- Alerts/Logging and Monitoring components responsible for:
  - Continuous background Information monitoring/logging/debugging
  - Custom alarms either locally or escalated to the cloud
- Data storage: a service for data storage for the generated data gathered by the Edge, given any resource constraints (storage) of the Edge device
- Semantic annotators: implementing the activity of “Semantic annotation” as previously discussed
- Machine Learning: an optional component making the Edge intelligent by holding models that can support decision support and local control. For example, an Edge gateway that controls 100 lamps of a street can decide about the dimming of the light of the whole street
- Real-time analytics: Services that provide real-time analytics on the data gathered by the Edge device; given the constraint processing power of the device this can be limited for a short period of data
- Device Registry and management: enable the Edge device to manage attached constraint/smart devices. For example, an Edge LoRa gateway that get values from hundreds of nodes, has a registry of their GPS location that is propagated later on to the cloud. Mobile nodes that move in the city need to register or unregister themselves automatically from the Edge that is nearest to them
- Data Decoding/Encoding services: as discussed in previous section implementing the “Decode/Encode payload” activities

- Network management and communications: Edge device needs to support many connectivity protocols especially in the South interface-towards devices (like BLE, ZigBee, OPC, etc) but also wired and wireless network types (Ethernet, Cellular, Wi-Fi, etc)especially in the North interface-towards the Cloud
- Security is a vertical aspect of the Edge device for the Data and the APIs. Lately there are efforts to involve Blockchain for data like the IOTA protocol

2.3 IoT Cloud Platform

Managing an IoT Cloud platform can be a difficult process. Considerations about always-on connectivity and latency need to be addressed, and while device to cloud data tends to be encrypted, having the data residing outside enterprise walls might prove a problem for regulated industries. Also many organizations deploy their private cloud or there are cases that you cannot afford any more to go with a Public IoT solution and you just want to install your own IoT platform even on top of a cloud provider as the shirt study in [3] explains.

2.3.1 The essential components of an IoT Cloud platform

Referring to Fig 2 the following components and services are identified at the IoT Cloud. The reader will notice that there are similar components that also reside in the Edge. The difference here is that the functionality is performed on top of an aggregation of Edge devices connected to the Cloud:

- Message Bus service, Connectivity, Network management: The Cloud platform needs to support multiple connectivity protocols as well as a Message bus service that allows cloud services to exchange messages
- Platform Management: Allows to manage all deployed IoT Cloud services
- Dashboards, UI: UI to Visualize and control the platform
    - Data Visualization: Specific UI for visualizing data from the “things”
- Application/Services runtime environment: this is offered by almost any Cloud platform and allows to install new Virtual Machines or Containers with new services and applications
    - A useful feature here will be the presence of Lambda functions as in an Edge computing device
- AAA of User/ Tenants registry and management and Access Control: needed for registering access of users but also to allow IoT 3rd party applications to use the IoT Cloud API via API key management, or define policies that allow access restrictions to resources
- Support for common APIs for 3rd part applications: External Interfaces [APIs, SDKs and gateways that act as interfaces for 3rd party systems (e.g., ERP, CRM)]. For example , Through this interface can be attached: billing and reporting, SDK application engine, Data export, Custom UIs, etc
- Semantic Annotations/Normalization: implementing the activity of “Semantic annotation” (e.g. types based on ontologies and standards, geolocation, etc) as previously discussed in Fig 1
- Rules engine: for decision making

- Data Analytics: services for performing operations on top of data
- Data Storage: provide (big and scalable) data storage
- IoT Infrastructure Management: needed for managing the IoT infrastructure that is connected to the IoT Cloud platform. It may offer:
    - Device/Edge Management: Backend services for the management of Edge/devices, perform remote software deployment & updates
    - Device registry: keeps all the registered Edges and devices
    - Monitoring: offers services for:
        - Continuous background Information monitoring
        - Creating and managing alarms
    - Usage Analytics: Contains information about the usage and utilization of the infrastructure
- Secure Communications: a common aspect for the whoel platform
- Device shadowing: used to store and retrieve current state information for a thing regardless of whether the thing is connected to the Internet. Seen in commercial IoT platforms like Amazon AWS IoT [21] a useful feature to find also in OSS.

3 Open Source platforms for the three tiers of "Things"-Edge-Cloud

In this section, we will examine OSS for each of the three tiers pf “Things”-Edge-Cloud. Although there is a plethora of solutions found by just browsing the Internet, as explained in Section 1.2, this work will focus on solutions that are Open sourced, with a clear licensing model, a vibrant community, large installed base and a clear roadmap. We will present also technologies that may perform a single function (e.g. Data Analytics) but can be easily integrated with the rest of the services existing in the Edge or the IoT Cloud, in order to provide a more complete implementation of the reference architecture of Fig 2 .

3.1 OSS for “things”

OS/RTOS on Embedded and Smart Devices exist for years, and many open source solutions are available. There are useful comparisons and lists at [22] [23] Notable efforts being actively developed are: RIOT, Zephyr from Linux foundation, Contiki and Apache Mynewt (See Table 1)

For the communication support (machine to machine) the most notable efforts are:

LWM2M client libraries:

- Eclipse Wakaama
- Leshan Client

MQTT clients:

- Paho one of the most known MQTT clients with many target programming languages, devices like Arduino, Android and a very active community
- Mosquitto client: mainly for C,C++ clients again with an active community

Developers should be also aware of ROS. Robot Operating System (ROS) [24] is the most popular robotics middleware and not an operating system. (Implementors call it, *meta-operating system*). The support for Real-time systems is being addressed in the creation of ROS 2.0.

Finally Eclipse Edje defines a standard high-level Java API called Hardware Abstraction Layer (HAL) for accessing hardware features delivered by microcontrollers such as GPIO, DAC, ADC, PWM, MEMS, UART, CAN, Network, LCD, etc. that can directly connect to native libraries, drivers and board support packages provided by silicon vendors with their evaluation kits.

TABLE 1 WELL DEVELOPED OPERATING SYSTEMS FOR CONSTRAINT DEVICES

	RIOT-OS	Zephyr	Apache Mynewt	Contiki
Multi-threading	yes	Partial support	yes	Partial support
Real-time	yes	yes	yes	partial
Source code URL	https://github.com/RIOT-OS/	https://github.com/zephyrproject-rtos/zephyr	https://github.com/apache/mynewt-core	https://github.com/contiki-os/contiki
Support cite	http://www.riot-os.org/	www.zephyrproject.org	https://mynewt.apache.org/	http://www.contiki-os.org/
License	LGBL	Apache License, version 2.0	Apache License, version 2.0	BSD
No of Committers	181	268	70	161
Version	RIOT-2018.01	1.11.0	1.3.0	3.0
Last code update	5/5/2018	5/5/2018	5/5/2018	8/3/2018
Supported Companies/Organizations	Freie Universität Berlin, INRIA, Hamburg University	Linux Foundation	Apache foundation	Texas Instruments, Atmel, Cisco, ENEA, ETH Zurich, Redwire, RWTH Aachen University, Oxford University, SAP, Sensinode, and many others
Programming languages	C, C++	C	C	C
Supported Standards/ Communication protocols	Partial POSIX compliance	Supports standards like 6Lowpan, CoAP, IPv4, IPv6, and NFC. Supports Bluetooth®, Bluetooth® Low Energy, Wi-Fi*, 802.15.4	CoAP and 6LoWPAN . BLE, LoRa PHY and LoRaWAN	Contiki supports fully standard IPv6 and IPv4, along with the recent low-power wireless standards: 6lowpan, RPL, CoAP.
Date seen	5/5/2018	5/5/2018	5/5/2018	5/5/2018

3.2 OSS for the Edge

This section presents OSS solutions for the edge, according to the described methodology and the degree that these solutions fulfill the presented edge architecture of Fig 2. Moreover, we present efforts by other organizations especially from the areas of fog computing and 5G. Since most presented edge solutions do not fully implement the proposed reference architecture we discuss complementary OSS solutions that can be easily integrated with the target edge solutions that the underlying constraint hardware can seamlessly execute.

3.2.1 Edge platforms

3.2.1.1 EdgeX Foundry

Hosted by The Linux Foundation, EdgeX Foundry [25] develops an open source common framework for IoT edge computing. The framework consists of open source microservices plug-and-play components and provides a reference software platform that claims to be hardware and OS agnostic. Dell Technologies contributed the initial EdgeX code and now it is supported by over 60 members, Industry and SMEs from both Europe and US. The project is at its infancy but there is already a second release called California scheduled for June 2018. EdgeX Foundry executive director claims that *"EdgeX wants to do for IoT on the industrial side what Apache did for websites"* [26]. The source code is located at [27] mostly is written in Java and uses extensively Docker containers. Table 2 displays the architectural features that are defined but still not all of them are implemented in the current release (in parentheses the corresponding EdgeX Foundry architectural element)

3.2.1.2 Eclipse KURA

KURA [28] is an IoT gateway supported project by the Eclipse foundation. KURA has a lot of the characteristic that an IoT solution needs at the Edge. It is platform independent (based on Java VM) and contains both South Communication and Connectivity APIs and services to interface it with many device protocols as well as a North interface for network management of various interfaces. Apart from standard messaging solutions (MQTT) and routing it offers a remote management solution for monitoring and controlling KURA services. Applications can be deployed remotely as OSGI bundles in similar way as other KURA services are deployed (e.g. embedded database, GPS service etc). KURA is a quite mature platform, initially contributed in 2014 to Eclipse by Eurotech [29]. Moreover, there is a benefit for developers that there are already development tools around KURA . Source code is available at [30] with 31 contributors at the day of writing and supported companies Eurotech, Redhat, IBM, Hitachi and Rapicorp

3.2.1.3 Little IoT Agent (liota)

Little IoT Agent (liota) is the answer of VMware for IoT solutions. Liota runs on any gateway that supports Python, and has a small footprint of dependencies thus running in platforms like Raspberry Pi. Liota offers an open source SDK for building IoT gateway applications for managing, monitoring and orchestrating data in an IoT gateway. Thus, it does not offer a complete solution but rather the framework (liot calls them abstractions) to develop and edge gateway solution. Liota source code is available at [31] with 11 contributors at the day of writing. In their future plans are to support more language bindings (C, C++, Java and Lua), include CoAP transport and a blockchain mechanism for creating unique identifiers.

3.2.1.4 Thingsboard IoT Gateway

This gateway solution is provided by ThnigsBoard.io as open source with the limitation that it allows to integrate IoT devices connected to legacy and third-party systems with ThingsBoard cloud IoT. Since ThingsBoard cloud IoT is also open source as you will see in next section we include also this solution here. It is written in Java, but it is based on microservices not OSGI so developers can use more languages while integrating with the gateway API. The ThingsBoard IoT gateway offers extension mechanisms, and some tasks like device provisioning, local data persistence, message converters. They have in their roadmap remote configuration, edge analytics and monitoring. The source code is at [32] actively developed but with a limited number of contributors.

3.2.1.5 Flogo

Flogo [33] is a lightweight, based on GoLang, open source framework for edge devices due to small footprint, but can be used also to complement an IoT cloud platform. Offers connectivity to IoT technologies like MQTT, CoAP and offers a RESTful API. It offers an engine and a flow service to execute flows. It is quite minimal, but has a wide range of (paid) extensions and plugins to extend its capabilities. It is quite active project and mainly supported by TIBCO software.

3.2.1.6 SDC-Edge

StreamSets Data Collector Edge (SDC Edge) [34]is a light edge solution, written in Go programming language and it enables at-scale data ingestion and analytics for edge systems. It provides edge analytics, data normalization, and provides pipelines for both sending and receiving data. SDC Edge is supported by StreamSets Inc.,can be easily installed in various OSS but there is also a containerized solution based on Docker. It seems not a complete IoT edge solution since it is only oriented for Data streaming. The source code is available at [35] and since it is supported by a single company has a limited number of contributors

3.2.2 Efforts to be aware of

3.2.2.1 Telecom Infra Project – Edge Computing

The Telecom Infra Project (TIP) was founded in February 2016 and is an initiative by a large number of members to create open solutions and as many open hardware and software components as possible in the telecom infrastructure. TIP has a specific Edge Computing group [36] which plans to deliver open solutions for both 5G & IOT service deployments.

3.2.2.2 The Open Connectivity Foundation

The Open Connectivity Foundation [37] was founded on February 2016 and is an initiative of over 400 member organizations mainly from Industry (like Microsoft, Samsung, Cisco, etc) delivering a standard communications platform and specification, an open source implementation and a certification program for devices involved in the Internet of Things (IoT).

3.2.2.3 ETSI multi-access-edge-computing

The *European Telecommunications Standards Institute* (*ETSI*) has initiated the multi access edge computing effort [38]. They have launched a set of specification for APIs and services related to Edge computing

3.2.2.4 The Open Edge Computing Initiative

The Open Edge Computing Initiative [39] founded by Carnegie Mellon University, Intel, NOKIA, Vodafone and others delivers an open source mobile edge solution.

3.2.3 OSS Smart home gateways

Although not placed a IoT gateways, Smart home gateways could be used by IoT platforms that want to extend functionality until homes. Most of these projects want to bring solutions to their users without the need of a cloud, but still the possibility is there, if there can be a North interface towards an IoT Cloud solution.

3.2.3.1 openHAB (for the edge, see home gateway as edge)

OpenHAB [40] is an open source software platform written in Java, it offers SDK to construct frontends and integrates different home automation systems, devices and technologies. It has a vibrant community and a huge list of open source add-on plugins to control numerous home devices by different manufacturers. It has a REST interface that it can be potentially exploited by interested IoT Cloud solutions but also the openHAB Cloud Connector [41]

3.2.3.2 Home Assistant

Home Assistant [42] is an open-source home automation platform based on Python offering a mobile application, SDK and components for connecting home devices. It also offers a REST API.

3.2.3.3 MyController.org

MyController [43] is a Java OSS controller and a home automation server to monitor and control a home from anywhere. It's architecture includes an MQTT and REST API at the North side, as well as, it has multiple features we seek in Edge computing like remote monitoring and configuration, rule engine, dashboards, user roles, etc. It is actively maintained at [44] by 9 developers at the time of writting

3.2.4 Services and technologies for an Edge device

resinOS [45]: It's a minimal OSS platform for running containers on constraint devices,

Ubuntu IoT Gateway [46]: Canonical's offer for implementing IoT edge gateways based on **Ubuntu Core**

OpenFaas [47] , **fnproject** [48]**, Apache OpenWhisk** [49] : Serverless and Function as a Service (Faas) technologies that enable lambda functions in a constraint device

Openstack-edge computing [50]: Openstack, the well-known OSS cloud platform, can be used for Edge-computing solutions although does not include specific IoT edge services as the one described by our reference architecture. At [51] for example, there are solutions to operate Openstack in the Edge

IOTA technology [52]: offers a marketplace allows connected devices to securely transfer, buy and sell fine-granular and diverse data across the global to any buyer

Apache Edgent [53]: is a programming model and micro-kernel style runtime that can be embedded in gateways and small footprint edge devices enabling local, real-time, analytics on the continuous streams of data coming from equipment, vehicles, systems, appliances, devices and sensors of all kinds.

See also on next section **Node-RED** and **Apache NiFi** that could also be used in Edge devices for executing dataflows.

TABLE 2 EDGE SOLUTIONS AND ARCHITECTURAL FEATURES

	EdgeXFoundry	KURA	LIOTA	ThingsBoard IoT Edge	Flogo	SDC Edge
Environment	Mainly Java(Python, Go, C)	Java	Python	Java	Go	Go
Remote Management	yes(System Management)	yes (Administration GUI)	no	planned	yes	yes
Monitoring	no/planned	yes(Link Monitors)	no	planned	via Extensions	yes
Remote software deployment and updates	yes(System Management)	yes	no	no	via Extensions	no
Tenants Identity registry	yes(Client registration)		no	no	via Extensions	no
Micro-services Runtime environment	yes(Docker containers)		no	yes	via Extensions	no
Rules engine	yes	yes(Apache Camel)	no	no	via Extensions	no
Alerts/Logging	Yes		no	planned	via Extensions	no
Data storage	Yes(Core Data)	yes(Data Services)	no	yes	via Extensions	yes
Semantic annotators	no (perhaps Heystack)		no	no	via Extensions	yes
Machine Learning	no		no	no	via Extensions	no
Real-time analytics	no/planned (Supporting Services (SS) Layer)		no	planned		yes
Device Registry and management	yes(Device Profile)	yes	Entities, EdgeSystems and Devices, Device Discovery	yes	via Extensions	
Data Decoding/Encoding services	provided SDK for developers (Device Services (DS))		Metrics, SI Units	yes	via Extensions	yes
Network management and communications	yes(Device/Export Services)	yes		yes	yesy	no
North Interface	yes(Export Services)	Cloud Services	DCCComms	MQTT	MQTT,CoAP	http
South Interface	yes(Device Services (DS))	Field Protocols/Device Abstraction	DeviceComms	Sigfox,OPC-UA	MQTT,CoAP	mqtt, http
Security	yes	yes(SSL, Firewall service)		no	via Extensions	no
Scheduling	yes			no	no	no
Committers	~30	35	12	4	17	5
Supported Companies	Linux foundation,Industry consortium 50+ members	Eclipse foundation (Eurotech, Redhat, IBM, Hitachi and Rapicorp)	VMWARE	ThingsBoard.io	TIBCO	StreamSets inc.
Licensing	Apache 2	Eclipse Public License	BSD 2	Apache 2	BSD	Apache 2.0

	EdgeXFoundry	KURA	LIOTA	ThingsBoard IoT Edge	Flogo	SDC Edge
Introduced	4/2017 Dell	6/2014 Eurotech	6/2016 VMWare	1/2017 ThingsBoard.io	TIBCO 10/2016	StreamSets 2017
Documentation	https://www.edgexfoundry.org/	https://www.eclipse.org/kura/	https://github.com/vmware/liota	https://thingsboard.io/docs/iot-gateway/	https://tibcosoftware.github.io/flogo/	https://streamsets.com/products/sdc-edge
Source code	https://github.com/edgexfoundry	https://github.com/eclipse/kura	https://github.com/vmware/liota	https://github.com/thingsboard/thingsboard	https://github.com/TIBCOSoftware/flogo	https://github.com/streamsets/datacollector-edge

3.3 OSS for the IoT Cloud

Similarly, as in previous section for the edge, this section presents IoT Cloud OSS solutions, according to the described methodology and the architecture of Fig 2.

3.3.1 Cloud IoT platform solutions

3.3.1.1 KAA project

Kaa [54] is an open-source IoT platform, mainly written in Java, supported by a company called KaaIoT [55] which actively maintains it and support is. The source code of the platform is available at [56] and seems to be maintained by an adequate number of contributors. It can be deployed as a single node or in a cluster deployment for high availability and has multi-tenancy support, that is a single Kaa instance can support multiple independent applications. Overall Kaa, offers a good solution, however in order for developers to connect a device to the platform, they need to embed Kaa SDK types. Still there is a RESTful API for accessing Kaa's services.

3.3.1.2 Eclipse Kapua

Kapua [57] is a platform for managing IoT edge gateways backed by Eclipse foundation. Kapua offers device, data and registry management as well as messaging services. It allows remote administration and easy integration with other applications via RESTful API. It also includes data storage for analytics as well as dashboards for visualizations. However, it seems that it is not quite generic but more oriented for specific solutions integrated with Eclipse KURA. The source code is available at [58], it is actively maintained with primary backer companies Red Hat and Eurotech.

3.3.1.3 TTN

The Things Network [59] was created in 2015 and it is based on LoRaWAN technology. It is a free and open IoT network around the world which consists of hundreds of gateways. It is included in this report since TTN offers as OSS all its software stack at [60] and one can either use it for a private deployment or contribute resource to the TTN IoT cloud. TTN IoT Cloud platform offers most of the envisaged services of an IoT Cloud stack: connectivity with edge LoRaWAN gateways, multitenancy, device management, and an MQTT as well as a RESTful API for end user applications.

3.3.1.4 FIWARE

FIWARE [61] is the outcome of the Future Internet Public-Private Partnership Programme (FI-PPP), the European Commission launched in 2011. Today FIWARE has formed an open community supporting from platform APIs to FIWARE labs and startups. FIWARE software components and services are deployed as FIWARE calls the Generic Enablers (GEs). In this context FIWARE offers an IoT software stack [62]. The core component is FIWARE flagship GE Orion Context Broker (also for northbound connectivity) while IoT Agents offer connectivity (i.e. MQTT, http, etc) in south interface to gateways and devices. Most of the FIWARE components are supported by Telefonica and the source code is available at [63]. Although there is adequate documentation for the offered components, there is no simple installation and integrating them all together is not a trivial process.

3.3.1.5 ThingsBoard
ThingsBoard [64] is an OSS IoT platform offering from core IoT cloud services to Dashboards via widgets. Standards like MQTT, CoAP and HTTP are supported as well as storage, device management, logging, etc. It claims to support seamless horizontal scalability and multitenancy, meaning that one can deploy it even for more complex scenarios of supporting multiple customers and applications. There are two ways to integrate devices, either by implementing within your device their MQTT, CoAP, HTTP APIs or by integrating the ThingsBoard Gateway as discussed in section 3.2.1.4. Source code is available at [65] and it is actively supported by an adequate number of contributors.

3.3.1.6 OpenRemote
OpenRemote [66] is an open source project offering three main elements around cloud configuration tools (OpenRemote Manager), a controller for integrating protocols acting as a gateway (OpenRemote Controller) and managing rules and UI control panel for designers (OpenRemote Designer). OpenRemote Manager is the IoT cloud part of the whole solution and offers APIs for 3rd party services. It is actively maintained by the Openremote Inc company and the source code is available at [67].

3.3.1.7 IoTivity
IoTivity [68] a Linux Foundation hosted project is actively developed and maintained and is highly sponsored by the Open Connectivity Foundation (OCF).

3.3.1.8 Devicehive
Devicehive [69] is another OSS IoT platform which distributed under Apache 2.0 license. It can be easily deployed via Docker and Kubernetes. It can connect to any device via REST API, WebSockets or MQTT.

3.3.1.9 SiteWhere
SiteWhere [70] is an OSS solution that provides the ingestion, storage, processing, and integration of device data. It runs on top of Apache Tomcat coupled with MongoDB and HBase. It delivers a complete device management solution, allows connecting devices with MQTT, AMQP, Stomp, and other protocols and adding devices through self-registration, REST services, or in batches

3.3.1.10 Thinger.io
Thinger.io [71] is an OSS platform that claims easy scalability. It offers admin consoles and REST API for Application integration and easy integration with Docker and IFTT [72]. It offers also libraries for Arduino and Android. The source code is not maintained the last year

3.3.1.11 ThingSpeak
ThingSpeak [73] is an OSS platform that is tightly coupled with MATLAB allowing users to analyze and visualize uploaded data using MATLAB without requiring the purchase of a MATLAB license. Although the commercial solution is active the source code is not maintained the last 3 years

3.3.1.12 Zetta
Zetta [74] is an open source platform built on Node.js for creating Internet of Things servers that run across geo-distributed computers and the cloud and build APIs for device interaction. Although Zetta has a small number of contributors is actively developed.

3.3.1.13 Distributed Services Architecture (DSA)

Distributed Services Architecture (DSA) [75] is an active OSS IoT platform that facilitates device inter-communication, logic and applications at every layer of the Internet of Things infrastructure.

3.3.1.14 WSo2

WSo2 [76] seems a mature solution and active OSS IoT platform, which also claims that can distribute and manage applications/firmware of devices as well as Share device operations/data with other users. It also has mechanisms to Deploy lightweight streaming analytics on the edge or a local gateway to optimize network traffic. Also support real-time analysis of IoT data streams with WSO2 streaming analytics, identifying complex patterns and kicking off alerts and initiating responsive actions.

Table 3 displays some of the core features of the most active solutions.

TABLE 3 IOT CLOUD OSS SOLUTIONS

	Kapua	KAA	TTN	FIWARE	ThingsBoard	OpenRemote	IoTivity	DeviceHive	SiteWhere	Thinger.io	ThingSpeak	Zetta	DSA	WSo2
Environment	Java	Java	Go/Python/ Javascript	~	Java	Java	C,C++	Java,Go	Java,Pythion, Javascript	C++, Java	Ruby	Node.js	Java, Dart,Python	Java,Javascr ipt
IoT Platform Management	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Monitoring	Yes	Yes	Yes	No	Yes	Yes	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Remote software deployment and updates	Yes	Yes	No	No	No	Yes(for controllers)	No	No	No	No	No	No	No	Yes
Tenants Identity registry	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	Yes	No	No	No	Yes
Micro-services Runtime environment	no	No	No	No	No	No	No	No	No	No	No	No	No	No
Rules engine	No	No	No	No	Yes	Yes	No	No	No	No		No	No	No
Alerts/Logging		Yes	Yes	Yes	Yes	Yes	No	Via plugins	Yes	No	Yes	No	No	Yes
Data storage	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	Yes	Yes	Yes	No	Yes	Yes
Semantic annotators		Yes	No	No	Yes via Attributes	No	No	via extensions	No	No	No	No	No	No
Machine Learning	no	No	No	No	No	No	No	via extensions	No	No	No	No	No	No
Real-time analytics	Yes	Yes	No	Yes	Yes	No	No	Yes	No(Free)	Yes	No	Yes (Stream)	Yes	Yes
Device Registry and management	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes
Data Decoding/Encodi ng services	no	via SDK	Yes	Yes	via extensions	Yes	Yes	via extensions	No	No	No	via extensions	No	No
Network management and communications	Yes	Yes	No	Yes	No	No	Yes	Yes	Yes	Yes	No	Via Scouts	Yes	No
North Interface	REST	REST	MQTT, REST	Yes	REST	REST API	REST API	REST API, MQTT	REST API	REST API	REST API, MQTT	REST API	REST API	REST API
South Interface	MQTT, CoAP	via SDK	LoRaWAN	MQTT,Co AP	MQTT,CoAP,H TTP	proprietary	CoAP,BLE, etc	WebSocket s or MQTT	MQTT, AMQP, Stomp, WebSockets	MQTT, CoAP	REST API	REST API	REST API	MQTT, WSO2
Security	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Committers	17	56	20-113	~40	~25	~8	~60	33	13	-	8	11	~12	~80

	Kapua	KAA	TTN	FIWARE	ThingsBoard	OpenRemote	IoTivity	DeviceHive	SiteWhere	Thinger.io	ThingSpeak	Zetta	DSA	WSo2
Supported Companies	RedHat,Eurotech,Bosch	KaaIoT	Thethingsindustries	Telefonica	ThingsBoard	OpenRemote Inc	Open Connectivity Foundation	DeviceHive	SiteWhere	Thinger.io	ThingSpeak	Apigee, Zetta Community	DGLogik	WSO2
Licensing	EPL 1.0	Apache 2.0	MIT	AGPL3.0, Apache2.0,…	Apache2.0	GNU AGPL 3,0	Apache 2.0	Apache 2.0	CPAL-1.0	MIT	GPL3	MIT	Apache 2.0	Apache 2.0
Introduced	6/2016 Eurotech	~2015	~2015	~2015	~2016	~2013	~12/2015	~2012	~2013	2015	2010	2014	2016	2015
Documentation	https://www.eclipse.org/kapua/	http://kaaproject.github.io/kaa/docs/v0.10.0/Welcome/	http://thethingsnetwork.org/	https://fiware-iot-stack.readthedocs.io/en/latest/	https://thingsboard.io/docs/	http://www.openremote.com/community/	https://www.iotivity.org/documentation	https://devicehive.com	http://www.sitewhere.org/	http://docs.thinger.io/console/	https://thingspeak.com/	http://www.zettajs.com/	https://github.com/IOT-DSA/docs/wiki	https://docs.wso2.com/display/IoTS310/
Source code	https://github.com/eclipse/kapua	https://github.com/kaaproject/kaa	https://github.com/TheThingsNetwork/ttn	https://github.com/Fiware	https://github.com/thingsboard/thingsboard	https://github.com/openremote	https://github.com/iotivity/iotivity	https://github.com/devicehive	https://github.com/sitewhere	https://github.com/thinger-io	https://github.com/iobridge/thingspeak	https://github.com/zettajs/zetta	https://github.com/IOT-DSA	https://github.com/wso2

3.3.2 Efforts to be aware of around IoT Cloud OSS solutions

3.3.2.1 IoT-EPI - European Initiative for IoT platform development

The IoT-European Platforms Initiative - IoT-EPI [77] is s a European Initiative addressing EU-funded H2020 programs in the domain of IoT. Currently under its umbrella are the following IoT research projects: Inter-IoT [78], BIG IoT [79], AGILE [80], symbIoTe [81], TagItSmart! [82], VICINITY [83] and bIoTope [84]. Most European funded projects are publishing their results as open source. From the above EU funded projects, related with this work are INTER-IoT, Big IoT and AGILE (especially for gateways).

3.3.3 List of IoT OSS Add-on solutions

Useful solutions for IoT Cloud developers to be aware of are the following:

3.3.3.1 IOTivity

IOTivity [85] is a Linux Foundation OSS project sponsored by the Open Connectivity Foundation (OCF) [86] with a single goal to provide a framework for device-to-device connectivity. It offers a cloud interface at the northbound as well as discovery and messaging services at the southbound interface. It is not an IoT Cloud solution, but by integrating IoTivity components with other 3rd part components one can have a good solution. The IoTivity model Things as resources while its offered cloud stack is composed of 4 subsystems, Interface, Account Server, Resource Directory and Message Queue. The project offers also software components for IoT device side for handshaking, resource registration/discovery, etc. There are some docker containers to ease the setup process. AllJoyn, another OCF project, is also merged as work now under IoTivity . IoTivity 's source code is available at [87] while it is a very active project.

3.3.3.2 Node-RED

Node-RED [88] is an IBM OSS contributed development tool for creating flows. The interesting part is the runtime engine in Node.js that can run your flows in the background, thus it can be very easily integrated with your IoT solution to ease the process of deploying flows and defining new logic for your IoT applications. The source code is available at [89] and is currently a very active project with more than 60 contributors to the project.

3.3.3.3 Apache NiFi

A similar solution to Node-RED, Apache NiFi [90] is a light-weight dataflow execution engine, to compose and execute generic IoT applications.

3.3.3.4 Analytics

There are no specific solution for IoT Analytics, but in most platforms well known Stream and Big data solutions such as ElasticSearch, Apache Spark, Cassandra and Kafka for real-time and batch processing can be easily integrated.

3.3.3.5 For Lambda functionality support

Serverless approaches are a feature of your cloud hosting. Apche OpenWhisk (as presented on edge section) is a serverless, open source cloud platform that executes functions, but still in incubation phase. Also if you have your own Cloud infrastructure for example based on OpenStack, there is no equivalent of e.g. Amazon Lambda, but there is something to expect from Openstack projects like Zun [91]

3.3.3.6 Eclipse Hono

Eclipse Hono [92] is an IoT connector which provides remote service interfaces for connecting large numbers of IoT devices to a back end and interacting with them in a uniform way regardless of the device communication protocol. Born out of the collaboration of large companies, including Red Hat and Bosch both members of the Eclipse Foundation, Hono aims to provide an open source framework that is completely open source, through which to build an end-to-end IoT solution that supports the main patterns: telemetry, events and command/control.

3.4 Semantic modeling for IoT

IoT testbed providers have recently started to add semantics to their frameworks allowing the creation of the semantic sensor web (SSW), which is an extension of the current Web in which information is given well-defined meaning, enabling M2M communications and interactions between objects, devices and people [93]. There is academic literature around the proper semantic modeling in IoT data and applications, but here are some efforts that have wide adoption

3.4.1 Project Haystack

Project Haystack [94] is an open source initiative to develop naming conventions and taxonomies for building equipment and operational data by fostering the common association and interests of software and technology companies. It is focused on developing semantic modeling solutions for data related to smart devices including: building equipment systems, automation and control devices, sensors and sensing devices, promotion and education with respect to the semantic data modeling industry for building automation systems, and to engage in educational activities directed towards the improvement of business conditions of the semantic data modeling industry for smart device data, all on a not-for-profit basis. All work developed by the project-haystack.org community is provided for use as open source software under the Academic Free License 3.0.

3.4.2 The SSN Ontology

The SSN ontology [95] is one of the most significant and widespread models to describe sensors and IoT-related concepts. The SSN ontology provides concepts describing sensors, such as outputs, observation value, feature observed, observation time, accuracy, precision, deployment configuration, method of sensing, system structure, sensing platforms and feature of interest.

3.4.3 OneM2M

OneM2M has published a report for home automation and describes concepts and relationships [96]. It collects and studies the state -of-the-art technologies that may be leveraged by oneM2M to enable its abstraction & semantics capability. This includes a collection of terminology and use cases considered by other standardization or industrial for a working on ontologies, semantics and abstraction.

4 Bibliography

[1] "Top Trends in the Gartner Hype Cycle for Emerging Technologies, 2017," [Online]. Available: https://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017/. [Accessed 12 12 2017].

[2] H. C. e. al., "Bringing the Cloud to the Edge," in *Proc. IEEE INFOCOM Workshop*, Toronto, Ontario, Canada, May 2014.

[3] H. o. I. a. D. Igor Ilunin, "How to Choose Your IoT Platform – Should You Go Open-Source?," 10 10 2017. [Online]. Available: https://www.iotforall.com/iot-platform-open-source-vs-serverless. [Accessed 18 10 2017].

[4] "Azure iot-edge," [Online]. Available: https://github.com/Azure/iot-edge.

[5] SDxCentral, "2017 SDxCentral IoT Infrastructure Report - Online Edition," 2017. [Online]. Available: https://www.sdxcentral.com/reports/2017/internet-of-things/. [Accessed 20 10 2017].

[6] "The Art of Low - Cost IoT Solutions," June 2017. [Online]. Available: http://www.dataart.com/downloads/DataArt_White_Paper_Art_of_Low_Cost_IoT_Solution.pdf. [Accessed 18 10 2017].

[7] E. I. W. Group, "The Three Software Stacks Required for IoT Architectures," 2016. [Online]. Available: https://iot.eclipse.org/resources/white-papers/Eclipse%20IoT%20White%20Paper%20-%20The%20Three%20Software%20Stacks%20Required%20for%20IoT%20Architectures.pdf. [Accessed 20 10 2017].

[8] IoT-Analytics, "White paper – Guide to IoT Solution Development," 2016. [Online]. Available: https://iot-analytics.com/product/guide-to-iot-solution-development/. [Accessed 10 10 2017].

[9] IEC, "White Paper - Edge Intelligence," 2017. [Online]. Available: http://www.iec.ch/whitepaper/pdf/IEC_WP_Edge_Intelligence.pdf.

[10] I. I. Consortium, "Industrial Internet Reference Architecture," June 2015. [Online]. Available: https://www.iiconsortium.org/IIRA-1-7-ajs.pdf. [Accessed 20 10 2017].

[11] ETSI, "GS MEC 003 V1.1.1 Mobile Edge Computing (MEC); Framework and Reference Architecture," March 2016. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/003/01.01.01_60/gs_MEC003v010101p.pdf. [Accessed 20 10 2017].

[12] e. a. Yun Chao Hu, "ETSI White Paper No. 11 - Mobile Edge Computing A key technology towards 5G," September 2015. [Online]. Available: http://www.etsi.org/images/files/ETSIWhitePapers/etsi_wp11_mec_a_key_technology_towards_5g.pdf. [Accessed 20 10 2017].

[13] Huawei, "Huawei Launched Edge-Computing-IoT Solution, Enabling Industry Digital Transformation," 2017. [Online]. Available: http://www.huawei.com/en/events/mwc/2017/b2b/iot/huawei-edge-computing-iot-solution. [Accessed 2017].

[14] postscapes, "IoT Hardware Guide," [Online]. Available: https://www.postscapes.com/internet-of-things-hardware/. [Accessed 19 10 2017].

[15] O. Alliance, "OSGI," [Online]. Available: https://www.osgi.org/. [Accessed 20 10 2017].

[16] HashiCorp, "Consul," [Online]. Available: https://www.consul.io/. [Accessed 20 10 2017].

[17] Docker, "Docker," [Online]. Available: https://www.docker.com/. [Accessed 20 10 2017].

[18] L. Containers, "LXD," [Online]. Available: https://linuxcontainers.org/lxd/. [Accessed 20 10 2017].

[19] "CoreOS," [Online]. Available: https://coreos.com/rkt/. [Accessed 20 10 2017].

[20] "Greengrass," Amazon, 10 2017. [Online]. Available: https://aws.amazon.com/greengrass/. [Accessed 20 10 2017].

[21] A. A. IoT, "iot-thing-shadows," [Online]. Available: http://docs.aws.amazon.com/iot/latest/developerguide/iot-thing-shadows.html. [Accessed 22 10 2017].

[22] Wikipedia, "Comparison of real-time operating systems," [Online]. Available: https://en.wikipedia.org/wiki/Comparison_of_real-time_operating_systems . [Accessed 05 05 2018].

[23] "List of open source real-time operating systems," [Online]. Available: https://www.osrtos.com/ . [Accessed 05 05 2018].

[24] R. O. S. (ROS), "Robot Operating System (ROS)," [Online]. Available: http://www.ros.org/. [Accessed 5 5 2018].

[25] "EdgeX Foundry," [Online]. Available: https://www.edgexfoundry.org/.

[26] "Edgexfoundry move forward its push harmonize iot edge computing," [Online]. Available: http://www.ioti.com/infrastructure/edgex-foundry-moves-forward-its-push-harmonize-iot-edge-computing.

[27] "EdgeXFoundry source code," [Online]. Available: https://github.com/edgexfoundry.

[28] "Eclipse KURA," [Online]. Available: https://www.eclipse.org/kura/.

[29] Eurotech, https://www.eurotech.com/en/press+room/news/?672.

[30] "KURA source code," [Online]. Available: https://github.com/eclipse/kura. [Accessed 3 12 2017].

[31] VMWARE, "LIOTA," [Online]. Available: https://github.com/vmware/liota. [Accessed 28 11 2017].

[32] Thingsboard.io, "Thingsboard-gateway," [Online]. Available: https://github.com/thingsboard/thingsboard-gateway. [Accessed 3 12 2017].

[33] Flogo. [Online]. Available: http://www.flogo.io/. [Accessed 12 12 2012].

[34] "StreamSets Data Collector Edge," [Online]. Available: https://streamsets.com/products/sdc-edge. [Accessed 14 12 2017].

[35] "SDC Edge source code," [Online]. Available: https://github.com/streamsets/datacollector-edge. [Accessed 14 12 2017].

[36] T. I. Project. [Online]. Available: http://telecominfraproject.com/project/access-projects/edge-computing/. [Accessed 3 12 2017].

[37] O. C. F. (OCF), "OPEN CONNECTIVITY FOUNDATION (OCF)," [Online]. Available: https://openconnectivity.org/. [Accessed 5 5 2018].

[38] ETSI, "multi access edge computing," [Online]. Available: http://www.etsi.org/technologies-clusters/technologies/multi-access-edge-computing. [Accessed 3 12 2017].

[39] o. e. computing. [Online]. Available: http://openedgecomputing.org.

[40] OpenHAB. [Online]. Available: http://www.openHAB.org. [Accessed 3 12 2017].

[41] "Connector, Openhab cloud," [Online]. Available: http://docs.openhab.org/addons/ios/openhabcloud/readme.html. [Accessed 3 12 2017].

[42] "Home-assistant," [Online]. Available: https://home-assistant.io/.

[43] M. Controller. [Online]. Available: https://mycontroller.org. [Accessed 3 12 2017].

[44] "Mycontroller-org sourcecode," [Online]. Available: https://github.com/mycontroller-org/mycontroller. [Accessed 3 12 2017].

[45] ResinOS. [Online]. Available: https://resinos.io/. [Accessed 6 12 2017].

[46] I. g. o. Ubuntu, "Ubuntu Core," [Online]. Available: https://www.ubuntu.com/internet-of-things/gateways. [Accessed 6 12 2017].

[47] "OpenFaaS," [Online]. Available: https://www.openfaas.com/. [Accessed 6 12 2017].

[48] fnproject. [Online]. Available: http://fnproject.io/. [Accessed 6 12 2017].

[49] A. Openwhisk. [Online]. Available: https://openwhisk.apache.org/. [Accessed 12 12 2012].

[50] O. edge-computing. [Online]. Available: https://www.openstack.org/edge-computing/. [Accessed 6 12 2017].

[51] A. L. e. al, "Revising OpenStack to Operate Fog/Edge Computing Infrastructures," in *IEEE International Conference on Cloud Engineering (IC2E)*, Vancouver, BC, Canada, 2017.

[52] IOTA, "The Next Generation of Distributed Ledger Technology | IOTA," [Online]. Available: https://www.iota.org/. [Accessed 5 5 2018].

[53] A. Edgent, "Apache Edgent," [Online]. Available: http://edgent.apache.org/. [Accessed 05 05 2015].

[54] K. project. [Online]. Available: https://www.kaaproject.org/. [Accessed 12 12 2017].

[55] KaaIoT. [Online]. Available: https://www.kaaiot.io/company/. [Accessed 12 12 2017].

[56] K. s. code. [Online]. Available: https://github.com/kaaproject/kaa. [Accessed 12 12 2017].

[57] E. Kapua, "Eclipse Kapua," [Online]. Available: Eclipse Kapua. [Accessed 6 12 2017].

[58] "Kapua source code," [Online]. Available: https://github.com/eclipse/kapua.

[59] "The Things Network," [Online]. Available: https://www.thethingsnetwork.org/. [Accessed 12 12 2017].

[60] "The Things Networks source code," [Online]. Available: https://github.com/thethingsnetwork. [Accessed 12 12 2012].

[61] FIWARE. [Online]. Available: https://www.fiware.org/. [Accessed 14 12 2017].

[62] "FIWARE IoT Stack," [Online]. Available: https://catalogue.fiware.org/iot_stack. [Accessed 14 12 2017].

[63] "FIWARE source code," [Online]. Available: https://github.com/telefonicaid. [Accessed 14 12 2017].

[64] "ThingsBoard," [Online]. Available: https://thingsboard.io. [Accessed 14 12 2017].

[65] "ThingsBoard source code," [Online]. Available: https://github.com/thingsboard/thingsboard.

[66] "Openremote Community edition," [Online]. Available: http://www.openremote.com/community/. [Accessed 14 12 2017].

[67] "Openremote source code," [Online]. Available: https://github.com/openremote/. [Accessed 14 12 2017].

[68] IoTivity, "IoTivity," [Online]. Available: https://iotivity.org/. [Accessed 5 5 2018].

[69] Devicehive, "Devicehive," [Online]. Available: https://devicehive.com. [Accessed 5 5 2018].

[70] sitewhere, "sitewhere," [Online]. Available: http://www.sitewhere.org/. [Accessed 5 5 2018].

[71] thinger.io, "thinger.io," [Online]. Available: https://thinger.io/. [Accessed 5 5 2018].

[72] I. T. T. That, "IFTT," [Online]. Available: http://ifttt.com/. [Accessed 5 5 2018].

[73] ThingSpeak, "ThingSpeak," [Online]. Available: https://thingspeak.com/. [Accessed 5 5 2018].

[74] zetta, "zetta," [Online]. Available: http://www.zettajs.com/. [Accessed 5 5 2018].

[75] D. S. A. (DSA), "Distributed Services Architecture (DSA)," [Online]. Available: http://iot-dsa.org/. [Accessed 05 05 2018].

[76] wso2, "wso2," [Online]. Available: https://wso2.com/iot. [Accessed 5 5 2018].

[77] "IoT-EPI - European Initiative for IoT platform development," [Online]. Available: http://iot-epi.eu/. [Accessed 14 12 2017].

[78] "INTER-IoT project," [Online]. Available: http://www.inter-iot-project.eu/. [Accessed 14 12 2017].

[79] "Big IoT," [Online]. Available: http://big-iot.eu/. [Accessed 14 12 2017].

[80] "AGILE," [Online]. Available: http://agile-iot.eu/. [Accessed 14 12 2017].

[81] "symbIoTe," [Online]. Available: https://www.symbiote-h2020.eu/. [Accessed 14 12 2017].

[82] "TagItSmart!," [Online]. Available: http://www.tagitsmart.eu/. [Accessed 14 12 2017].

[83] "VICINITY," [Online]. Available: http://vicinity2020.eu/. [Accessed 14 12 2017].

[84] "bIoTope," [Online]. Available: http://www.biotope-project.eu/. [Accessed 14 12 2017].

[85] "IoTivity," [Online]. Available: https://www.iotivity.org/. [Accessed 14 12 2017].

[86] "Open Connectivity Foundation (OCF)," [Online]. Available: http://www.openconnectivity.org/. [Accessed 14 12 2017].

[87] "IoTivity source code," [Online]. Available: https://github.com/iotivity. [Accessed 14 12 2017].

[88] "Node-RED," [Online]. Available: http://nodered.org/. [Accessed 20 12 2017].

[89] "Node-RED source code," [Online]. Available: https://github.com/node-red/node-red. [Accessed 20 12 2017].

[90] A. NiFi, "Apache NiFi," [Online]. Available: https://nifi.apache.org/. [Accessed 5 5 2018].

[91] O. Zun, "OpenStack Zun," [Online]. Available: https://wiki.openstack.org/wiki/Zun. [Accessed 5 5 2018].

[92] E. HONO, "Eclipse HONO," [Online]. Available: https://www.eclipse.org/hono/. [Accessed 5 5 2018].

[93] A. Sheth, C. Henson and S. Sahoo, "Semantic sensor web.," *IEEE Internet Computing,* vol. 12, no. 4, p. 78–83, 2008.

[94] P. Haystack, "Project Haystack," [Online]. Available: https://project-haystack.org/. [Accessed 5 5 2018].

[95] M. Compton, P. Barnaghi and P. Bermudez , "The SSN ontology of the W3C semantic sensor network incubator group," *Web Semantics: Science, Services and Agents on the World Wide Web,* vol. 17, pp. 25-32, 2012.

[96] OneM2M, "Study of abstraction and semantics enablement v.0.7.0. study of existing abstraction and semantic capability enablement technologies for consideration by OneM2M," Technical Report OneM2M (TR 0007), 2014.

