On the Integration of Heterogeneous Data Sources for the Collaborative Internet of Things

Federico Montori, Luca Bedogni, Luciano Bononi
Department of Computer Science and Engineering (DISI)
University of Bologna, Italy
Email: {federico.montori2, luca.bedogni4, luciano.bononi}@unibo.it

***Abstract*—The Internet of things is foreseen as one of the next imminent Internet revolutions, as many devices will seamlessly communicate together to provide new and exciting services to the end users. One of the challenges that the IoT has to face is about both the heterogeneity of the data available and the heterogeneity of the communication. In this paper we focus on the former, by presenting an architecture able to integrate data coming from different sources, including custom made deployments and government data. New services can be deployed directly by the end users, using reliable or unreliable data sources, and new processed data can be gathered by these services and used by others.**

I. Introduction

The Internet of Things (IoT) is one of the research and industrial fields that faced the most rapid growth in the recent years, mostly thanks to the proliferation of new technologies associated with the ease of installation and use. This development has created a wide variety of standards and solutions at each layer of the network and application stack, leading to an heterogeneous environment both in terms of communication technologies and data storage. This results in disconnected network islands, in which it is easy to build networks with homogeneous devices, however it is hard to integrate data provided by other sources. Since the beginning of its diffusion, its potential has been explored in various fields of application and its major usefulness has been claimed to be in service composition and interoperability [1]. The requirements when designing collaborative IoT-related automation systems are varying due to the heterogeneity of the platforms and the hardware components as well as the network interfaces. This resulted in a sparse set of technologies and terminologies used in several scenarios determining a lack of interoperability among systems. The common approach to the problem of unifying entities within an ecosystem is typically architectural and leads to a difficult reuse of the components among different solutions [2]. To face these issues the European Commission supported initiatives like IoT-A [3], which aimed to release an architectural reference model, and FI-WARE [4], which also helped architects in establishing a unified vision and nomenclature and now had become an implementation-driven open community. FI-WARE also provided a sandbox, in which partners could upload their open data, although it is not of broad use nowadays. Such solutions, unfortunately, did not solve the problem introduced by architectures, in fact different approaches still tend to create separate ecosystems which are hard to unify.

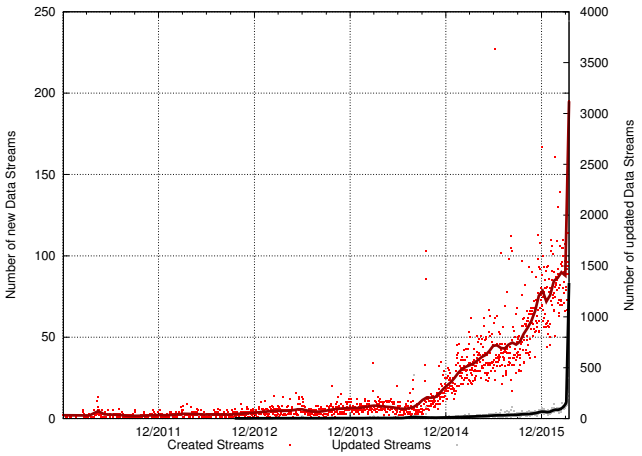

Fig. 1. Trend in creation and update of ThingSpeak channels.

At the same time, makers worldwide build their own IoT in-home networks, which provide a low cost and customized environment that suit their needs. Platforms like Arduino [5] and Raspberry Pi [6] have demonstrated their ease of use and they fit most of the needs of citizens willing to build their own network. However, certain types of sensor can be too expensive, or cannot be deployed due to restrictions or physical space limitations. For this reason, collaborative IoT is seen as a useful solution that facilitates the access to critical data.

In this paper we aim to show the usefulness and the potential of open data exploitation for global cooperative IoT automation scenarios. As a preliminary approach, we focused our analyses on data coming from two of the main open data platforms for sensors, ThingSpeak [7] and SparkFun [8], which provide open data uploaded by privates and coming from different types of sensors, along with temporal and, possibly, positional information. Such parameters can tell us much about the general trend in the usage of these platforms throughout a time window of few years. Each data stream in both platforms comes with a creation date, which we reported, after a data extraction, in the diagram in Figure 1.

Starting from such analysis it results a substantial growth

in created channels starting from the end of 2014. A possible intuition behind this phenomenon is the parallel innovation in simple hardware modules, for instance August 2014 corresponds to the launch of the first version of ESP8266 [9] on the market and in October 2014 was possible to flash its firmware though an SDK [10]. In the same figure we also reported the time of the last update for all the found streams. The oldest updates we found fall back to 2012, therefore we do not have any information on streams that were updated for the last time before that date. The steepness of the curve in the last days reveals that a significant amount of the data streams are still in use and updated daily or even hourly, and can therefore provide fresh information. Such analyses shed some light on how rapidly the world of open data is growing and people are gaining interest in using a platform that takes away the burden of creating a local ecosystem. Thus, our work creates the basis for a solid global ecosystem with a strong impact on cooperative services, integrating data coming from custom made solutions with official data provided either by government agencies or other reliable data sources.

The rest of this paper is structured as follows: In Section II we present similar works from literature; Section III describes the data stream examples we considered for this analysis, and Section IV is devoted to the description on how it is possible to integrate heterogeneous data and the challenges on the topic; Section V describes our proposed architecture, and Section VI concludes this paper.

II. Related Work

The IoT is nowadays growing exponentially together with the number of solutions and architectures proposed to handle it. The total number of connected devices is expected to be 27 billions by 2024, while the total revenue opportunity is predicted to be up to $1.6 trillions [11] . Regardless of the different perspectives, it is clear that the number of devices is growing, the data is becoming more and more heterogeneous, and one of the main challenges is how to handle such an amount of data and how to give a meaning to it. In the recent years there has been a huge number of attempts which, most of the times, are either self-contained, since they require compliance to a specific framework, or commercial solutions.

Commercial solutions aim to constitute a living ecosystem in which entities are "plugged" and interoperable, participating for the benefit of the whole system and fully compliant with the other actors within the environment. Most of the times such frameworks, some of which are depicted in [12], provide efficient software adapters for legacy systems. Such types of frameworks are often self contained and tend to create a cluster of devices which need to be framework-compatible in order to interoperate. An example is Cumulocity [13], a platform providing an unified service oriented HTTP REST interface to devices. Another project attracting interest in recent years is AllJoyn [14], developed by the Allseen Alliance. Such a framework again forces devices to either implement an attachment to a software bus between applications, which is indeed the AllJoyn core, or connect to an AllJoyn router using a thin library. Either way, the communication introduces very low overhead and grants integration to even constrained devices. However, the protocol used is highly customized and makes AllJoyn a quite isolated ecosystem. Another example is Xively [15], which, again, allows devices to obtain interoperability even among different application protocols (CoAP, MQTT, HTTP, XMPP and others) offering an API that implements a custom message bus. Finally, another ecosystem to mention is the one implemented by Open Mobile Alliance, named OMA-LwM2M [16], which defines a custom layer over CoAP focused on exchanging instances called "objects" and operating upon them via the custom interfaces.

III. Open data as a source

As stated in the introduction, open data are the most powerful source of information when such data are not producible by the utilizers themselves. In this section we outline two well-known sources that we considered in order to achieve an homogeneous data store. From both sources we extracted open data in the form of data streams (or, similarly, data channels), locations relative to one user each in which such user uploads measurements produced by its sensors.

A. ThingSpeak

ThingSpeak [7], originally launched in 2010 by ioBridge, is an open source data platform and API for the IoT that allows the user to collect, store and analyze data. In more detail, it provides a personal cloud that users can deploy over their Local Area Network and easily display the data produced by sensors using ThingSpeak's straightforward API. Data analysis and visualization have been made possible due to the close relationship between ThingSpeak and Mathworks, Inc. since such functionalities are driven by the integrated MatLab support.

Furthermore, such a platform provides a global cloud hosting millions of open data records, called channels, which is useful both to users who cannot deploy their own cloud and to consumers who need to infer information coming from the stored data. Data is stored with an absolute freedom of expression, meaning that any data channel can have any name and does not need to stick to any format constraint. Data channels can be both private and public and provide also raw measurements encoded in XML, JSON or CSV and can be updated with new measurements every 1015 seconds.

In the recent years, ThingSpeak had become very popular due to the rise of easily programmable IoT platforms such as Arduino [5], BeagleBone Black [17], ESP8266 [9] and many others. Such devices are becoming cheaper and cheaper and, on the other hand, it is easier to get started with them. Nowadays, for instance, an ESP8266 is able to manage a sensor, get connected through WiFi, be programmed through the simple, C-like Arduino SDK and still cost less than 5$ while its battery, if the duty cycle is light enough, is estimated

to have a duration of around 7 years [18]. With a WiFi connection and an open platform such as ThingSpeak, a first home sensor network is very easy to bootstrap, since the device owner does not need to have the control on the cloud and, furthermore, the data produced by the sensor is easily displayable on the end consumer's personal device, such as a Smartphone or similar.

B. Sparkfun

SparkFun Electronics, Inc. [8], founded in 2003 in Colorado, is a microcontroller seller and manufacturer, known for releasing all the circuits and products as open-source hardware. Along with the latter, it also providestutorials, examples and classes.

For the purpose of the present paper, SparkFun also hosts its own open source cloud of open data[1], on which the customers can test and upload the data collected by the embedded sensors. Users can push for free their data on such cloud in streams of 50 MB maximum size and with a maximum frequency of 100 pushes every 15 minutes. Unlike ThingSpeak, the location where the data comes from is specified at a coarse granularity since the name of the city is often obtainable, however the real GPS coordinates are never given. On the other hand, data coming from SparkFun cannot be private and consumers can download stream contents encoded in JSON, XML, CSV, MySQL, Atom and PostgreSQL.

IV. Data Unification

In this section we point out the data streams' characteristics obtainable from our two sample open data clouds and how we aim to unify them onto a single data cloud. We extracted the whole repositories and parsed the JSON files in order to give a first structure to such data. Since the data structure does not force strong constraints data is often incomplete in such a way that, in some cases, it is not usable. This happens when no location information is given, the stream name and the description is not understandable, the stream has not been recently updated and so on.

Hereby are briefly presented some of the metadata that can be extracted from data streams:

- **Stream ID**: it is the data stream's unique ID. In ThingSpeak it is represented by an incremental number, which is assigned when the stream is created. At the time of writing there are 28806 active and public streams with IDs spanning from 0 to 100172. In SparkFun the unique ID is given by a string of 20 random ASCII characters. At the time of writing we counted 3575 different SparkFun streams.
- **Stream name**: it present in both platforms and it is determined by the user with no constraint. It might carry or not useful information about the stream.
- **Geolocalization**: it is present in both platforms. In ThingSpeak not all the streams come with GPS data. Similarly, in SparkFun not all the streams are geolocalized, however, when they are, only the name of the city, or sometimes just the state or even just the country, is given. When extracting data in JSON from SparkFun, GPS coordinates are given, however we observed that such coordinates are probably obtained through some API converting the name of the city, since streams coming from the same city have the same GPS coordinates.
- **Tags**: are included in both platforms and represent the keywords that users assign to streams. They often help to infer useful information about the data.
- **Creation Timestamp**: it is included in all ThingSpeak and SparkFun streams as a metadata. It usually does not correspond to the timestamp relative to the first registered update, since each stream has a limited number of updates staying registered, then the platform erases the first updates in excess. In SparkFun the limit is 50 MB, while in ThingSpeak is 100 updates.
- **Last Update Timestamp**: it is included in all ThingSpeak streams as a metadata. In SparkFun is simply deducible from the timestamp of the last update in the stream, since the timestamp is implicitly included for each update.
- **Description**: it is a ThingSpeak metadata and its characterization is fully assigned to the user (who can also decide not to include it).
- **Elevation**: it is a ThingSpeak metadata and not always indicated.
- **Last Entry ID**: it is a ThingSpeak metadata, which points to the last update record in the data, ordered using an incremental ID for each update.

Each data stream can contain different data fields. An example is given by the streams making use of the popular DHT11 or DHT22 sensors, which are devoted to sense temperature and humidity and, therefore, such channels have two fields. Data fields, both in ThingSpeak and SparkFun, also have names, which represent the only way to discriminate which field registers which measurement. In both platforms each measurement comes together with an integrated timestamp.

From each data stream we extracted in particular the GPS position for a location analysis, finding that such position is indicated, with different degree of precision, in 6665 data streams out of 32381. 32% of such cases belong to SparkFun, thus, as stated before, the GPS position indicates the center of the entity (the city, or the region) where the source is located. The results of the analysis are outlined in figure 2.

Given such results, the importance of information fusion from different sources is clear, since merging such sources not only increments the sampling number of the sensing infrastructure, but also its coverage. Indeed, ThingSpeak appears to have much more utilization in the European region, whereas SparkFun seems to be more popular in North America. Furthermore, this consideration might be extended to different macro topic areas, meaning that some open data sources are specialized

[1] https://data.sparkfun.com/

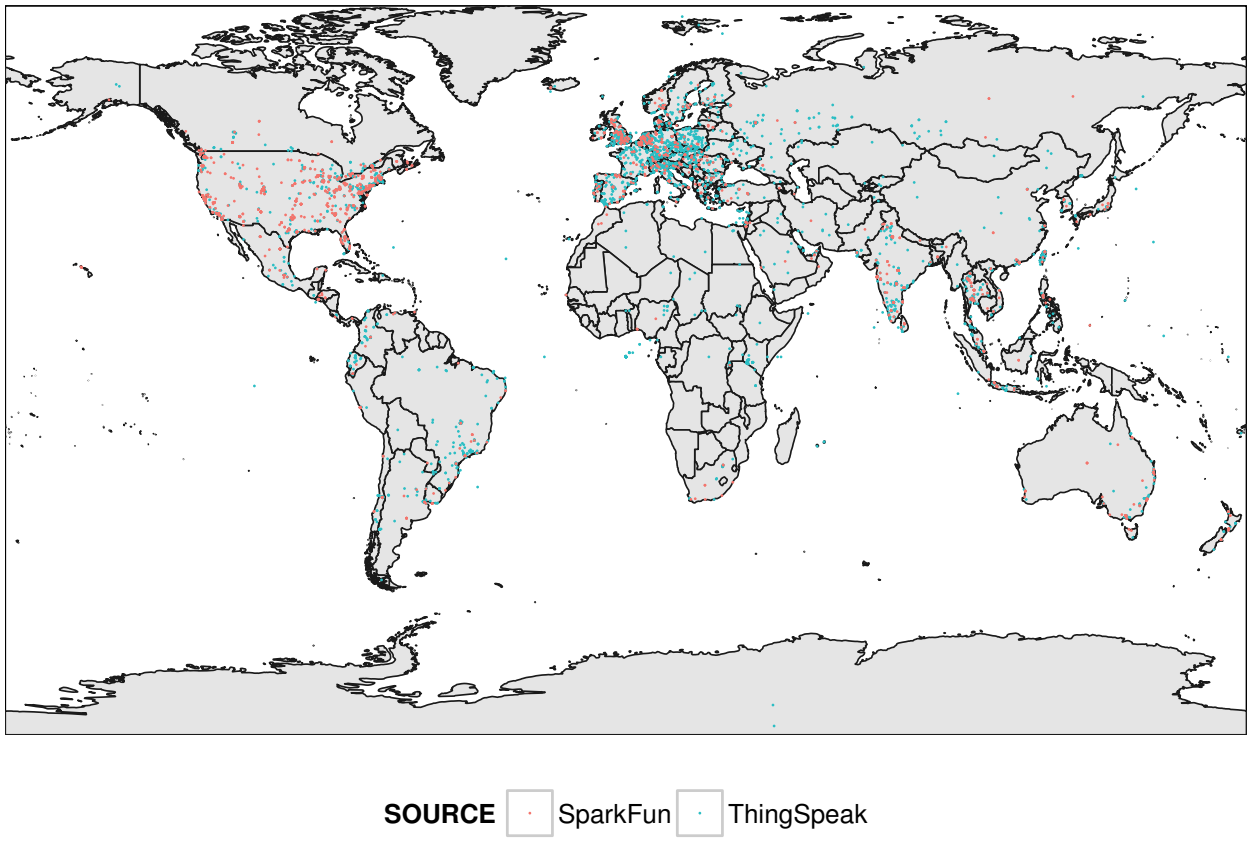

Fig. 2. Location of all ThingSpeak and SparkFun sensing sources.

on a specific field of measuring. For instance, governmental sources providing open data such as EPA (United States Environmental Protection Agency) [19] are primarily focused on environmental data, whilst crowdsensing sources such as OpenSignal [20] regard measurements on cellular network signal strength and coverage.

Therefore, a basic unification counting on an essential set of metadata is crucial, composing the minimum skeleton to which a data stream should be linked. For such purpose we aim to design an unique ID assignment policy, a geolocalization (in GPS coordinates with a precision error), the freshness of the information (given by the last update timestamp), when it was created (given by the stream creation date), a friendly name and an inferred measurement category for each field (such as temperature, humidity and so on) together with an unit of measure. The latter is essential, since most applications need to use services providing a certain type of information, which will be given by the class assigned to the measurement field. Without such a semantic approach, each data stream will have no meaning.

V. Our Architectural Proposal

In this section we discuss our proposal and how we intend to carry out the architectural design. We also give a glimpse on the case studies for this scenario in order to show why would someone use open data integration for measurement tasks. Figure 3 shows an architecture depicting several use cases.

In our proposed architecture we assume to have a decision middleware, called Orchestrator, which is capable to return a service record, or a data stream, given a set of parameters determined by the user's choice. Such information is returned from one among the sources available, provided that the user specified his or her preference for "reliable" or "unreliable" data, which namely corresponds to official or user-defined respectively. We also aim to have our own data cloud for both data streams and services which are not intended to be published as open data onto one of the sources mentioned.

As a case study, a user can build and run a custom application making use of different measurements, e.g. outdoor temperature and the amount of fine dust or pollen in the air, in order to infer an environmental condition or to

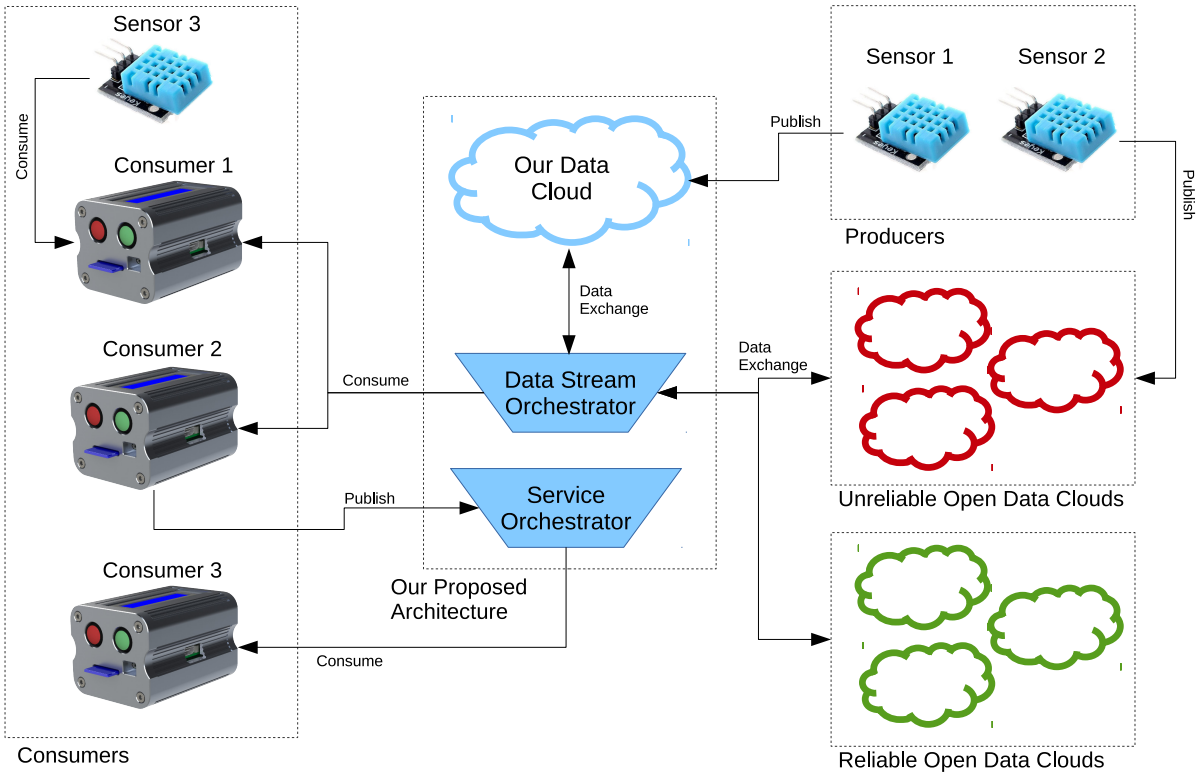

Fig. 3. Our proposed architecture.

trigger some action. For instance it would open automatically the window that is not facing the sun when it's too hot, but only if the pollen in the air is below a certain threshold, otherwise it turns on the air conditioning in order to avoid allergic reactions. This avoids pointless wastes of electrical energy while keeping the domestic environment safe. Such a case study can take place in different scenarios. A user, for example, might be the owner of the temperature sensor, since it is cheap and easily configurable, hence he will use it locally (in Figure 3 such case is represented by "Sensor 3"). However, other sensors such as pollen sensors or fine dust sensors might be too expensive or rare to get, or simply not owned by the end user and therefore other users' measurements are needed, provided they are nearby enough (this is also why geolocalization is meant to be crucial). In this case, the outcome of the application is possible only through integration of local resources with other measurements coming from other sources published in an open data platform (case represented by "Sensor 2" in Figure 3) or in our cloud (case represented by "Sensor 1" in Figure 3). Comment: in the architecture, sensor 1 and 2 are exactly the same. Either we change the figure, or we have to specifiy this better

Furthermore, data sources might have a different update rate and a different "reliability", since they may belong to providers that are recognized as trustworthy or not. For non-official data we also plan to use a feedback policy based on data streams' estimated precision and update rate as well as users' opinions, helping consumers and orchestrators to perform choices based on a trustworthiness value. Furthermore, in the integration presented in Section IV we used two user-driven platforms, which we aim to integrate with reliable and official data, such as EPA for the United States environmental measurements. Such measurements, since they are official, are considered to be reliable, however their slow update rate introduces a trade-off whenever a user must choose between a high reliability or a fast update rate. Some applications indeed might require information at a finer granularity over time, for example when they want to detect instantaneous condition changes. In such cases the user, choosing the update frequency at the expense of reliability, will necessarily use the data or the services provided by neighbors.

Our proposed architecture aims not only to unify raw

data streams and make them universally available, but also to make users able to share their service endpoint and to provide additional capabilities derived from both data aggregation and personal computational capabilities (represented by "Consumer 2" in the figure). As a simple example, a user receiving temperature and humidity data might calculate the heat index and expose it as a service interface. In such cases, the end consumer will not make use of the values onto the data streams, but it will query the orchestrator for a published and available service running on some private system. This reflects the concept of SOA. In conclusion, our proposal, given such a various set of use cases, provides the user with a wide variety of options regarding deployment and data retrieval. This is significantly straightforward since the user is not forced to stick to a particular approach and gives a great advantage in an era where heterogeneity affects not only data and protocols, but also solutions.

VI. Conclusions

In this paper we have studied the challenging topic of data integration between heterogeneous data sources for the Internet of Things. We have considered open data coming both from reliable sources like Governmental agencies as well as unreliable sources, made available through open clouds such as ThingSpeak and SparkFun. We analyzed the differences, and proposed a new architecture to integrate them together, along with the ability to deliver custom made services to the end users, using both reliable and unreliable data.

Future works on this topic go through the integration of additional data sources, which will eventually provide a wider set of data. We also plan to use Natural Language Processing (NLP) techniques, as much of the work will then be devoted to the study of how to integrate heterogeneous data, together, which can be described using natural language by the users providing them.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] S. Krco, B. Pokric, and F. Carrez, "Designing IoT architecture (s): A European perspective," in *Internet of Things (WF-IoT), 2014 IEEE World Forum on*. IEEE, 2014, pp. 79–84.

[3] "IoT-A European Project." [Online]. Available: http://www.iot-a.eu/public/front-page

[4] "FI-WARE Open Community." [Online]. Available: https://www.fiware.org

[5] "Arduino." [Online]. Available: http://www.arduino.cc/

[6] "Raspberry Pi." [Online]. Available: https://www.raspberrypi.org/

[7] "ThingSpeak, the open data platform for the Internet of Things." [Online]. Available: https://thingspeak.com/

[8] "SparkFun Electronics," \url{https://www.sparkfun.com/}. [Online]. Available: https://www.sparkfun.com/

[9] "ESP 8266." [Online]. Available: http://www.esp8266.com/

[10] "Espressif SDK Releases." [Online]. Available: http://bbs.espressif.com/viewforum.php?f=46

[11] Machina Research, "Global M2M market to grow to 27 billion devices, generating USD1.6 trillion revenue in 2024." [Online]. Available: https://machinaresearch.com/news/global-m2m-market-to-grow-to-27-billion-devices-generating-usd16-trillion-revenue-in-2024/

[12] H. Derhamy, J. Eliasson, J. Delsing, and P. Priller, "A survey of commercial frameworks for the Internet of Things," in *Emerging Technologies & Factory Automation (ETFA), 2015 IEEE 20th Conference on*. IEEE, 2015, pp. 1–8.

[13] "Cumulocity Framework." [Online]. Available: https://www.cumulocity.com/

[14] "AllJoyn Framework." [Online]. Available: https://allseenalliance.org/framework/documentation

[15] "Xively." [Online]. Available: https://xively.com/

[16] "Open Mobile Alliance lightweight Machine-To-Machine solution." [Online]. Available: http://openmobilealliance.org/about-oma/work-program/m2m-enablers/

[17] "BeagleBone Black." [Online]. Available: http://beagleboard.org/black

[18] A. Di Nisio, T. Di Noia, C. Carducci, and M. Spadavecchia, "Design of a low cost multipurpose wireless sensor network," in *Measurements & Networking (M&N), 2015 IEEE International Workshop on*. IEEE, 2015, pp. 1–6.

[19] "United States Environmental Protection Agency." [Online]. Available: https://www3.epa.gov/

[20] "Open Signal." [Online]. Available: http://opensignal.com/