This is a pre-print version of the paper that will be published in the "Ad Hoc Networks" Journal , Elsevier.
Available online since 28 March 2013 at the Editor's website : http://www.sciencedirect.com/science/article/pii/S1570870513000590

Smart Things in the Social Loop: Paradigms, Technologies, and Potentials

Luigi Atzori, University of Cagliari, Italy – *l.atzori@diee.unica.it*
Davide Carboni, CRS4, Italy - *dcarboni@crs4.it*
Antonio Iera, Università Mediterranea di Reggio Calabria, Italy - *antonio.iera@unirc.it*

Abstract:

Information about human social activities and relationships are exploited by an ever increasing number of proposed applications and protocols in several scenarios, given the consequent increase in the system performance. Examples are data transmission over delay tolerant networks, content recommendation in search engines, and advertisement of products and services. An emerging field where social networks are being exploited is the Internet of Things, where smart objects connect to the network to bring the real world into the virtual dimension. Objects capable to communicate on social network sites are able to enter into their owners' social loop so as to automatically publish information of interest for selected communities of people and to perform some related automatic actions. In so doing, not only can objects be part of the human social networks but they can also build *their own* social network. As a consequence, interactions among them can be fostered towards the development of complex services for the direct benefit of people. Accordingly, objects mimic the human behavior towards a scalable and effective service discovery and composition as well as trustworthiness management.
On the basis of the importance achieved by this trend in the last couple of years, in this paper we intend to review the adopted approaches towards the exploitation of social network concepts by the Internet of Things, the technologies behind these, and the potentialities.

Keywords: Internet of Things, Social Networks, Smart objects

1. Introduction

The number of objects that are currently accessing the Internet, side-by-side to human beings to advertise, search for, and accessing enhanced services is growing exponentially. Among them are sensors, actuators, wireless and mobile devices, or simply every-day-life objects enhanced with capabilities to interact with the external world through the Internet. This is a clear signal that the much-vaunted (and sometime abused) Internet of Things paradigm is already turned into a reality on which there is a strong convergence of the interests of researchers, users, and industries. As a main effect, we have today a new approach available to build enhanced applications and services involving the communications among objects on the Internet to the service of the human beings.

Several studies have focused their attention to the definition of architectural models and solutions towards the use and the inter-connection of Web-enabled objects using open protocols and well-known architectural styles, REST and SOAP based Web services (such as, [1] [2]). As a consequence, the obvious evolutionary step of the IoT is the so called Web of Things (WoT) that envisages new scenarios and applications where Internet enabled objects become active actors and peers in the Web. Sample services, applicable to Smart Cities or Smart Homes, are given below:

- The car driver knows about the status of her car and of the roads on the path towards her destination. Such awareness is achieved by accessing, through her mobile phone (or through any communication technology in her car), web services that are fed by data collected from sensors scattered both in her car and in the areas of interest.
- The domestic appliances may be accessed by the owner through web services from remote sites and some actions can be performed on them to prepare comfortable conditions for a better welcome home.
- Eco-compatible houses may be equipped with controllers and sensors able to measure the local energy production and consumption and manageable through web services towards a reduction of the environmental impact.

Besides the obvious advantages of the depicted sample scenarios, one cannot hide the doubts on the ability of the proposed solutions to effectively harness the full potential of the new paradigm without colliding with the limitations of the current Web service platforms in the presence of trillions of additional actors (objects, precisely).

In our opinion, Web of Things is a paradigm which goes in the right direction but is not the solution to the cited issues. To foster resource visibility, service discovery, object reputation assessment, source crowding, and service composition in a Web populated by people and countless things there is the necessity to strongly push towards solutions that exploit concepts directly derived from the sole platforms that currently seem to be able to effectively allow peer-to-peer exchanges among huge numbers of actors, i.e., Social Networks.

Even if several aspects of the social networking among humans cannot be directly applied to the objects' world due to the specific distinctive characteristics (e. g., high heterogeneity and limited intelligence), such a need has brought to a substantial convergence of the "Internet of Things" and "Social Networks" domains. Interesting ideas have recently appeared in the IoT arena, which testify to the interests in Social Network oriented solutions for the Internet of Things.

People at the User Experience Lab at Ericsson Research started from the idea that the complexity of network solutions that underlie the Internet of Things are hardly understood (and mentally accepted) by all users. Thus, it is wise to make this complexity completely transparent during the user-thing interactions. Differently, the concept of "friendship" and 'social relations' are understood by virtually everyone, as they are intuitive concepts. As a consequence, they proposed a solution to both the practical scalability and understand-ability issues which is simply "dressing" a network of things as if it was a social network [3]. They have been the first to introduce the concept of "Social Web of Things" and also made some applications' prototypes.

Further studies and implementations of this concept have been carried out around the world. An example is given by the work in [4], where the authors propose a Social Web of Thing Framework based on the Restful Web Service and Social Networks, discuss the relevant key technologies and use cases, and introduce a case study named MagicHome. Furthermore, even a prototype of a scalable architecture for a large scale social Web of Things for smart objects and services, named Paraimpu, has been developed [5].

In line with this evolutionary path, but from a different perspective, the authors of [6] and [7] introduce the concept of Social Internet of Things. In analogy with the social networks of human beings, they (i) define of a notion of social relationship among objects, (ii) design a reference architectural model implementing a social Internet of Things based on codified inter-object relationships, (iii) analyze the social network structure, which derives from the objects interactions based on the defined social relationships.

The examples above make us realize that the time is ripe for a serious reflection on the possible ways of integrating *objects* into *social networks*, whether they are shared with those of their owners or they are independent and autonomous.

Aim of the present paper is to analyze the potentials of a synergic use of Social Networks and Internet of Things concepts towards the deployment of effective service platforms able to face the future challenges of a future world of trillions of interconnected objects. We will illustrate the main solutions that are appearing in the IoT arena to let things enter the so called "social loop" and compare their points of strength and their weaknesses by also highlighting their technological requirements and architectures.

This paper is organized as follows. In Section 2 we present the technologies behind the Web of Things as one of the prevailing approaches towards the integration of the objects into the Internet. In Section 3, we describe how this paradigm can be extended by providing the things with the capabilities to take part to the human social activities on relevant social network websites. In Section 4 we describe a complementary approach that allows objects to build their own social networks, so that interactions among them can be fostered towards the development of complex services. In Section 5 we present the ongoing projects that come out from the concepts described in the previous sections. Finally, in Section 6 we draw final conclusions.

2. Web of Things paradigm and technologies

The ongoing evolution of the Internet of Things towards the Web of Things (WoT), where Web-enabled smart objects connect and communicate with each other by using the Web, has raised several research issues ranging from the adoption of the right protocol and communication paradigms to the choice of the most suitable architectural styles. WoT was not born as a field in academic research but rather as the attempt to build an ecosystem from an heterogeneous variety of services and products, often not conceived in a way to interoperate.

Several efforts have focused their attention to the definition of architectural models and solutions, towards the use and the interconnection of Web-enabled objects, which exploit open protocols and well-known architectural styles, such as Representational State Transfer (REST)and Simple Object Access Protocol (SOAP) based Web services.

In [8], an architecture is defined for the development of composite applications, to interconnect physical devices, on top of the open and simple standards that made the success of the Web (REST, XML, HTTP, or Atom). The layered architecture is composed as follows:

- Device accessibility layer: a layer that from the application point of view enables consistent access to all kinds of connected objects.
- Find-ability layer: a layer that, given an ecosystem of billions of smart things, allows for finding their services.
- Sharing layer: although device accessibility and find-ability can technically allow for sharing data too, this layer is specifically designed to manage a social circle authentication, based on accounting and authorization procedures.
- Composition layer: a layer that enables users to create composite applications on top of smart things.

The overall goal of this layered architecture is to facilitate the integration of smart things with existing services on the Web and to facilitate the creation of Web applications by using smart things. The layers above are not directly mapped onto the ISO OSI model, but are rather useful to understand the foundations of WoT as an ecosystem.

The next subsections analyze the main aspects relevant to the design choices for architecting the WoT. These include: the architectural style (SOAP/REST), the degree of centralization, the device/thing degree of accessibility to the network, and the ways in which data, services and objects can be composed together.

2.1. SOA(P) vs. REST

The choice between the SOAP and the REST software architecture styles deserves a deeper investigation.

In [9] the authors go towards the definition of an architecture where devices are viewed as services, in order to integrate a wide range of physical devices into distributed IT enterprise systems adopting a Service-Oriented Architecture (SOA). In a similar way, the projects WS4D7 and SOCRADES [10] apply a SOA approach to the context of embedded networks. Moreover, existing standards for Web Services (WS), focused on embedded devices, such as Device Profile for Web Services (DPWS) [11], confirm a real consensus and effort towards a SOAP-based WoT.

On the other side, some recent research works (i.e. [12][13][14]) adopt REST for IoT/WoT architectures. According to the experience documented in [13], the programmatic complexity of SOAP based services is not well-suited for the end-user to create ad-hoc applications, while the authors of [14] state that, in many cases, the SOAP complexity becomes superfluous whereas RESTful services can support “a la Mash-up” integrations.

In a RESTful architecture, the main resources, such as entities, collections, or anything else that is worth being represented in the application domain, are uniquely identified by its own URI. The reference methods - in this case, the HTTP verbs - are mapped onto application semantics in a very simple and straightforward way.

Many new Internet companies (see the trends in Fig. 1) that face the market, often prefer to provide their services through a RESTful API rather than a solution based on SOAP. This suggests that, probably, REST is more immediate and less expensive for rapid prototyping.

The main advantages of REST web services can be summarized as it follows:

- Lightweight – not a lot of extra XML markup.
- Human Readable Results - in some cases (i.e., a GET request of a resource representation), RESTful services can be directly invoked by typing the URL in a common web browser.
- Easy to build – no toolkits required.

SOAP also has some advantages mainly related to the formal definition of service interfaces:

- Easy to consume - clients can be automatically generated on the basis of the exact specification provided in the WSDL document.
- Rigid – type checking adheres to a programming contract and XML schemas are provided for data exchanged to filter unacceptable inputs/outputs.

- Development tools and business process definition (e.g., BPEL language).

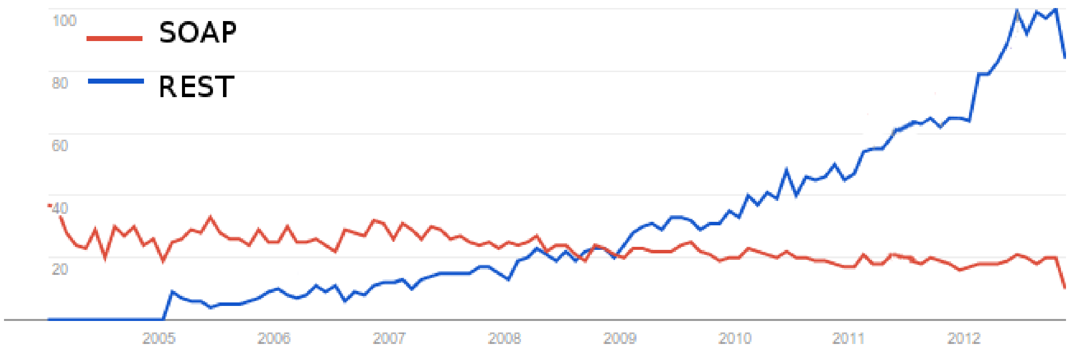

Fig. 1. *REST API Vs SOAP API in Google Trends.*

Even if the locution “Web Service” was often associated to the SOAP stack, this is not the way the Web works. Whereas SOAP is aimed at the next phase of Internet development defining a set of brand new specifications, REST is more the realization that the existing principles and protocols of the Web are already there to create robust Web services.

Summarizing, even if the strong type checking of SOAP based Web services is a plus, in the context of the IoT, the RESTful Web Services have some advantages over SOAP such as less overhead, less parsing complexity, statelessness, and tighter integration with existing HTTP. Moreover, the RESTful protocol called CoAP [15] is similar to HTTP but re-designed especially for devices with small footprint and constrained computing environments. According to the recent studies mentioned in this paper it seems that the use of standards like CoAP and EXI and web paradigms are the key factors for extending the Internet making the vision of the IoT become reality.

2.2. Device Accessibilty

It is useful to explore how things can be classified according to their computation and communication capabilities. The following list provides a coarse classification:

- *Virtual Things*, like web sites, e-mail boxes and social networks, just to mention some. These “objects” can be easily wrapped and then referenced in a HTTP addressing space like resources (REST) or like services (WSDL) or they already provide such abstractions and interfaces.
- *HTTP-enabled Smart Appliances*, like wireless printers, networked screens, and smartphones. These are already equipped with a network connection and a complete HTTP stack but usually do not provide a WS stack; thus, it is necessary to deploy a proxy or to install a minimal WS stack in the device, where possible. Usually, HTTP-enabled objects are not able to receive incoming requests from outside. In many cases, they are deployed under firewall restrictions

and only can act as clients. One possibility is to deploy in the middle a relay like YALER [16], which enables secure Web access to embedded systems behind a firewall/NAT/network gateway. A simple HTTP handshake makes a Web service running on the hidden device accessible from any HTTP client. Another option is WebSocket, which provides full-duplex communication channels over a single TCP connection. The WebSocket API is being standardized by the W3C, and the WebSocket protocol has been standardized by the IETF as RFC 6455.

- *Internet-enabled Things that are not equipped with a complete HTTP stack but can still communicate at the TCP/IP or UDP/IP level*. For those objects it is straightforward to build a HTTP wrapper and a WS stack as a proxy. A viable protocol alternative to HTTP is CoAP [15]. CoAP could be viewed as a compression or redesign of HTTP by taking power, memory, and computation constraints into account. Just like HTTP, which is designed as a transfer protocol for traditional web media content, CoAP is redesigned as a transfer protocol for devices to implement interoperations.
- *Network-enabled Things that cannot communicate over IP networks, but still can communicate with different protocols like ZigBee, Bluetooth or X10*. For those objects a proxy can be deployed to present them in the HTTP addressing space, by also using WS technology standards. A viable solution to extend IP also to small devices that usually are not equipped with IP-stack is 6LoWPAN [17].
- *Things not digitally enabled, bare physical objects*. For these objects a digital counterpart must be built and published online. RFID or barcode sticks can be used to interface these objects with devices and networks.

2.3. Centralized Vs Decentralized

Decentralized architectures have a number of theoretical advantages, such as single-point-of-failure robustness and privacy/anonymity enforcing capability. When analyzing the Web, one discovers that (i) the degree of decentralization is quite low, (ii) the prominent topology is client-server, and (iii) even if P2P networks have gained popularity in some application areas, the idea to move computation at the boundaries of the Internet is more a niche than a main stream.

The distinction between client-server and P2P architectures is no longer a technological distinction, but a matter of governance. The so-called “clouds” often adopt their own solutions of P2P load-balancing and fault tolerance, but the control is centralized in the hands of one provider. Cloud computing and Software-as-a-Service demonstrate that large IT companies are concentrating computation inside large globally distributed infrastructures; thus, transforming personal computing in a mere user interface to remote computing services.

Also, in the new field of Internet-enabled objects, we foresee that P2P connections between objects are unlikely to occur unless hard real-time and multimedia communications are involved.

A centralized online service can speak multiple protocols and data formats on top of HTTP and can act as a broker to let services meet in a logical space; it can be a proxy of data coming from sources and can be a relay of data to consumer services; moreover, it can make data format adaptation where required. In other words, it can operate at all the layers described above, from device accessibility to composition.

In a centralized WoT tool the workload is put at the extreme. Some applications need to have samples on every second, and even only one thousand sensors connected to a central server would produce 1000*60*60*24 = 86400000 events per day.

The load is structured in many small HTTP POST messages with a keep-alive connection. The requirements for *a cloud-based Web of things architecture* (see Fig. 2) can be summarized as it follows:

- C10K+ [18] capable web servers (C10K+ stands for 10,000 or more HTTP connections handled simultaneously, i.e. non-blocking servers that do not map every connection to a system thread);
- database engine able to be horizontally partitioned over a grid of machines in a transparent way (sharding);
- event handling and data processing delegated to a pool of worker processes distributed among multiple processors and even different machines.

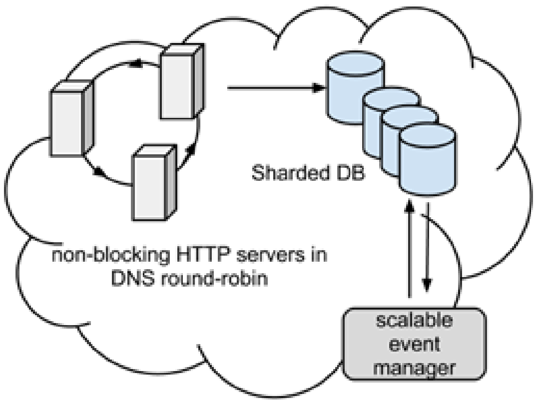

Fig. 2. *The architecture of a WoT-cloud based service.*

In [19], a little but insightful benchmark between a non-blocking web server, like *Nginx*, versus a thread-based server, like *Apache*, is presented. The plots show (Figs. 3 and 4) how the throughput of

Nginx are still high even with a large number of connections and with a very low memory usage. Nevertheless, the WoT system designer must consider that to fully gain an advantage from a non-blocking Web server, all the components in the backend must be designed to be non-blocking as well.

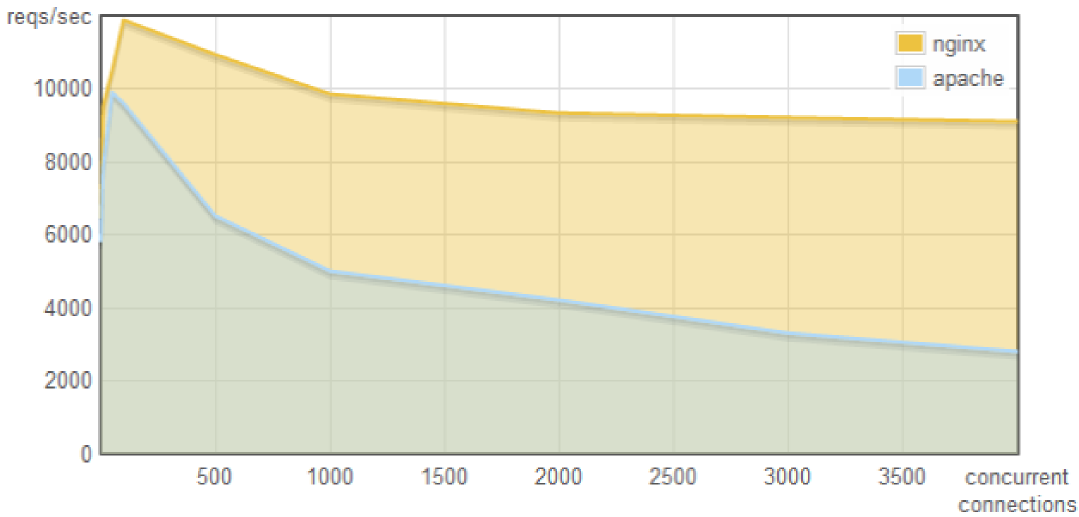

Fig 3. *Throughput against concurrent connection for nginx (non-blocking) and Apache (threaded) web servers [19].*

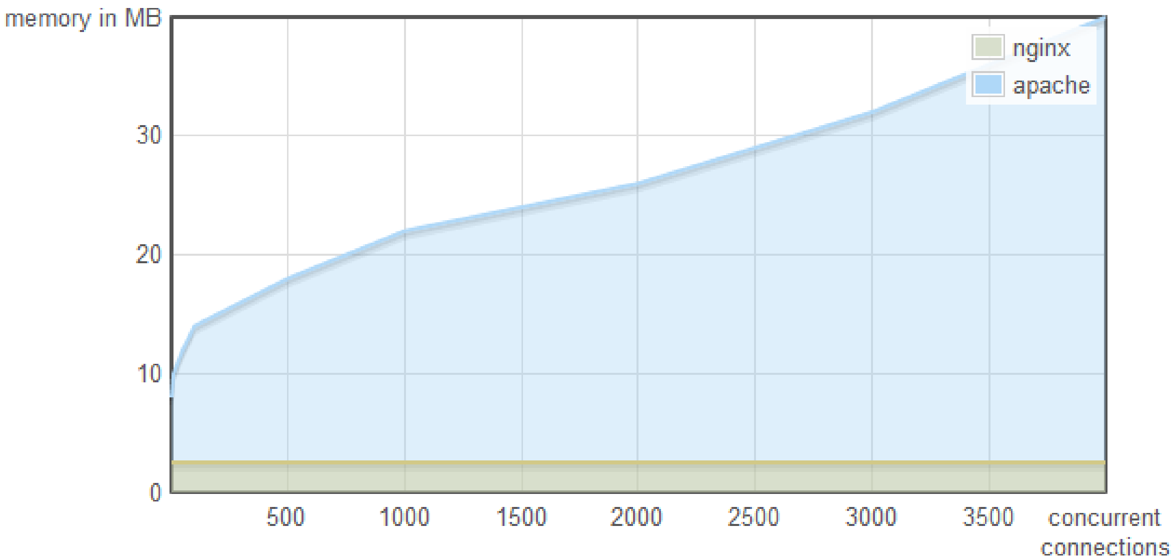

Fig 4. *Memory usage against concurrent connection for nginx (non-blocking) and Apache (threaded) web servers [19]*

2.4. Smart Objects in Service Orchestrations

A big issue in composing objects/services to form an application logic, is how to ensure that data coming from a data source can be properly read and processed by a recipient data sink. The intuitive, but not widely practicable solution, is to adopt a rigorous set of data schemas shared by all participants. In this way connections could be handled with no pains because every object that consumes data is able to receive and to decide what to do with the data. Unfortunately, having a common set of data types defined and shared for all interconnected objects is far to be realistic. Objects are built from different manufacturers for vertical applications, often inside walled gardens or with small capabilities to interoperate with external entities.

A more realistic assumption is to consider objects able to produce data in a format among those commonly encapsulated into HTTP messages. So we can expect that a data source can push a string containing either numbers or alphanumeric values, or more structured data like JSON objects or XML instances.

As an example, Paraimpu [20] implements the concept of mapping as a couple of expressions in the form of (*cond*, *repl*), where *cond* is a boolean expression while the *repl* expression is a valid instance of the data type expected by the sink (the actuator).

This approach can be subsumed in the more general IF-THEN paradigm applied to events, data and actions in the IoT. The online service called IFTTT (IF This Than That), although initially conceived for social networks and not for physical mashups, is the immediate proof-of-concept of how this mechanism can work. Also other sensor data collectors like COSM (former Pachube) implements the concept of trigger (IF condition on datum THEN post datum on URL).

A more versatile approach is to use business process definition to declare a more complex composition logic. In the case of SOAP-based WoT mashups, interactions can be described in two ways: executable processes and abstract processes. Both can be modeled by BPEL. Executable processes model the actual behavior of a participant as interactions, while abstract processes describe the observable behavior and/or process template. BPEL extends the WS-* interaction model to enable business transactions. BPEL defines an interoperable composition model that enables the extension of automated process integration both within and between businesses.

Pintus et al. [21] also propose a SOA framework where smart things are described by using the WSDL standard, while logical connections between smart things are modeled as web services orchestrations by using the BPEL language.

The availability of an API to be consumed by clients, of course, opens also the possibility to write third-party programs or scripts in arbitrary programming languages and build applications as a result of the composition of data coming from source objects, processed and transformed into actions for other smart objects.

3. Adding the Social Aspect to WoT

To date, IoT has been conceived as a top-down technology that silently takes the control of objects in a machine-to-machine internetworking with large benefit for logistics, manufacturing, and business productivity. On the other hand, the point we stress here is that for IoT/WoT to acquire some socialization value, end-users should be enabled to be active and creative in the definition of new relationships where smart things become either an instrument or a toy to share or to play with collaboratively. In this respect, it is important to provide users with enabling platforms, based on known and understandable metaphors, to manage, control and personalize the composition between sensor data and actions in the real life.

In order to enable a wide adoption of WoT in all sectors of the society, the WoT architecture and its protocols should motivate every citizens to contribute to a growing number of devices and smart objects in order to build new streams of information available to the community.

We can identify at least two generic use cases. The first one is the participatory sensing and the second one is the “device in a circle”.

Participatory sensing applications use data of mobile sensor nodes collected in collaboration with the device owner. With the right tools, community groups may participate in campaigns to collect data on highly-focused topics, such as traffic patterns, pollution-safe routes for school buses, without waiting for an institutional body or private agency to perform the detection. For example, the NoiseTube project [22], started in 2008 at the Sony Computer Science Lab in Paris and currently hosted by the BrusSense Team at the Vrije Universiteit Brussel, proposes a participative approach for monitoring noise pollution by involving the general public.

For users to become active in the bottom-up construction of a participatory sensing scenario, aspects like trust, reputation, control of distributed devices and tracking of information flows must be appropriately adjusted to control what is happening with the information and devices contributed. The identities of users are not harvested and should be kept anonymous while on the other hands the management of reputation of the data produced must be deployed to allow the participatory sensing to become effective and discard malicious or irrelevant data.

In the “device in the social circle” use case the social-ability is a key element to share and collaboratively use the smart things in the Web. Device accessibility layer and findability layer can technically allow sharing as well, but to manage physical objects in a social circle, features like authentication, accounting and authorization need to be properly designed and implemented.

Integrating smart objects and main-stream social networks is another key aspect. Every WoT system who aims to become popular and adopted by users, should integrate and communicate with social media in some way. In some extent, existing social media are the precursor of the future social IoT because they have already faced a number of issues like privacy concerns, content moderation, insights visualization and so forth.

The integration of Social media and Web of Things can occur in many ways:

- User Login/Authentication: users can access a system by using existing credentials without building a brand new profile. Thanks to the Oauth protocol, the user is redirected to her/his preferred social tool to perform the login. This leads to the immediate benefit to have contacts/friends imported in the new system. In this way, it would be possible for a generic WoT site to connect its users with their Facebook or Twitter counterparts.
- Social network as monitoring tool, for instance the IoT portal [23]: it allows users to quickly connect sensors and actuators to the system, and then create dashboard visualizations as Facebook applications. These applications allow users to share and monitor sensed data and control actuators in the real world.
- Sensors/Actuators: a social entity, such as user's Facebook wall and the Twitter flow associated to an hashtag, can be considered a sensor or an actuator (or both) just like physical things in the implementation of virtual-to-physical mashups. For instance, the "device in the circle" could be a set of parking sensors and the social circle could be the list containing the customers' contacts of a restaurant keeper. As an example, the restaurant manager can build a geographical selection of parking sensors in the neighborhood saving this selection as a datasource in her work space called "parkingSlots". Then she can create a link between the selected datasource and her customer list using for instance his Facebook profile. In so doing, her customers will receive the number of free slots directly on their mobile or PC and decide to go there by car or by other means. As dual example, the "device in the circle" can be a device shared and controlled by means of social media, like for example, a multimedia installation, even on a large scale (e.g., night-time lighting of a skyscraper). The social circle may be the set of users of a certain social network that post messages to a certain topic (for example, a topic with a hash tag on twitter). Analytics based on text recognition (or even mood recognition) on tweets may trigger different actions on the installation (for example, the ignition with certain colors or with certain choreography).
- The social dimension is useful for device/thing find-ability. The navigability of users' profiles enables to discover objects shared by friends, but also to find new friends and new devices/things.

4. Evolution of Social Internet of Things

4.1 Objects that handle social relationships

In the IoT, everything real becomes virtual, which means that each person and thing has a locatable, addressable, and readable counterpart on the Internet. These virtual entities can produce and consume services and collaborate toward a common goal. The car driver might know about the status of her car and of the roads towards her destination, thanks to the autonomous communications of the sensors and actuators in her car with those installed in other vehicles and along the road. The embedded system in a swimming pool could share its state with other virtual entities. All these scenarios are possible with an intense interaction between objects and related services, enabling the most powerful and fascinating applications. For instance, in [24] the authors introduce the idea of objects able to participate in conversations that were previously only available to humans. Those envisioned are objects aware of dynamic community structures, thus being able to develop a spontaneous networking infrastructure based on the information to be disseminated other than information on the objects themselves. Analogously, the research activities reported in [25] consider that, being things involved into the network together with people, social networks can be built based on the Internet of Things and are meaningful to investigate the relations and evolution of objects in IoT.

Again, IoT and social network technologies and concepts are jointly exploited towards the development of tools that can make people’s lives easier. This time the idea is to use social networking elements in the Internet of Things to allow objects to autonomously establish social relationships. The driving motivation is that a social-oriented approach is expected to put forward the discovery, selection and composition of services and information provided by distributed objects and networks that have access to the physical world. The proposed social-oriented approach is characterized by the capabilities of the objects to autonomously establish social relationships of different kinds ([24], [26], [7]).

Within the resulting object social network, a key objective will be to publish information and services, find them, and discover novel resources to support the implementation of complex services and applications. This can be achieved in a trusty and efficient way by navigating a social network of “friend” objects, instead of relying on typical Internet discovery tools that cannot scale to billions of future devices. Indeed, social networking concepts have proven to be of great importance for handling the relationships among humans and, in the same way, these are expected to have a great impact on the management of services in the IoT.

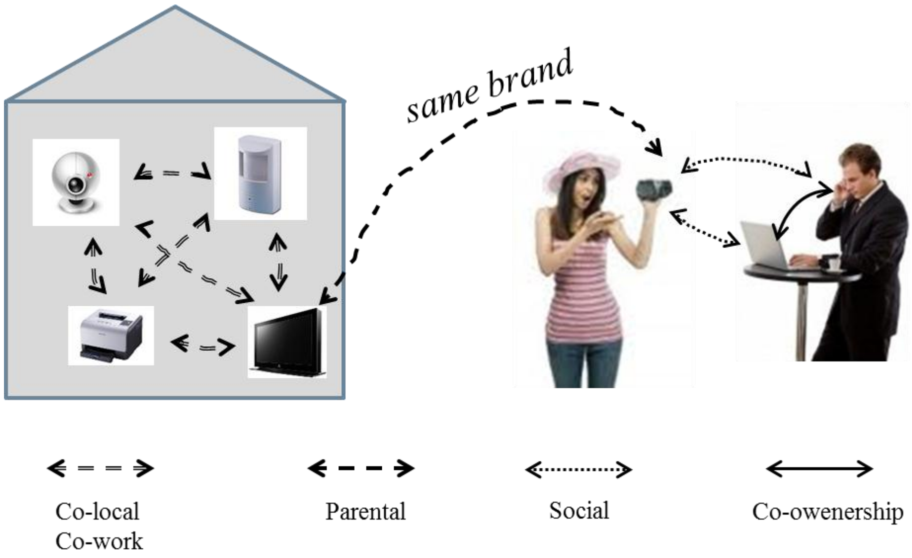

Fig. 5. *Sketch of the five types of relationships defined in the SIoT paradigm.*

Specifically, in [7] the Social Internet of Things (SIoT) has been proposed. According to this model, a set of forms of socialization among objects are foreseen as shown in Figure 5. The *parental object relationship* is defined among similar objects, built in the same period by the same manufacturer (the role of family is played by the production batch). Moreover, objects can establish *co-location object relationship* and *co-work object relationship*, like humans do when they share personal (e.g., cohabitation) or public (e.g., work) experiences. A further type of relationship is defined for objects owned by the same user (mobile phones, game consoles, etc.), which is named *ownership object relationship*. The last relationship is established when objects come into contact, sporadically or continuously, for reasons purely related to relations among their owners (e.g., devices/sensors belonging to friends); it is named *social object relationship*. These relationships are created and updated on the basis of the objects features (such as: object type, computational power, mobility capabilities, brand) and activity (frequency in meeting the other objects, mainly). The parental and ownership relationships are determined by just the static characteristics of the object (or slowly varying characteristics): type, brand, ownership. The others are determined by the movement of the object and by the other nodes it encounters. The relationships' management is implemented in the cloud, in the object gateways, and in the objects themselves, if capable of supporting the relevant logic. Clearly, the configuration of these functionalities is controlled by the object owner; accordingly, the resulting links are asymmetrical.

To manage the resulting network and relationships, the foreseen SIoT architecture is made of four major components (among others). The *Relationship management* introduces into the SIoT the intelligence that allows objects to start, update, and terminate relationships. The selection of which friendship to accept is based on human control settings. *Service discovery* is finalized to find which objects can provide the required service in the same way humans seek for friendships and information. Indeed, to discover the service, the object queries its social relationship network. *Service composition* enables the interaction among objects. The service discovery exploits the object relationships to find the desired service, which is then activated by this component. Both a reactive and a proactive approach to service composition are envisaged. This component will also include the functionality of crowd information processing, to process the information obtained from different objects and obtain the most reliable answer to a query on the basis of different visions. *Trustworthiness management* is aimed at understanding how the information provided by other members has to be processed.

4.2 The SIoT architecture

In [6], a possible SIoT architecture has been proposed, which follows the simple three-layer architectural model for IoT. In this model, the sensing layer is devoted to data acquisition and node collaboration in short-range and local networks. The network layer is aimed at transferring data across different networks. Finally, the application layer is where the IoT applications are deployed together with the middleware functionalities. The application layer is the key part in the SIoT, making this architectural proposal distinctive with respect to alternative solutions. It consists of three sub-layers, with the base sub-layer devoted to the database for the storage and management of the data and relevant descriptors, which record the social member profiles and their relationships, as well as the activities carried out by the objects in the real and virtual worlds. The component sub-layer works on top of the base sub-layer and implements the functionalities described in the previous subsection, i.e., relationship management, service discovery, service composition, trustworthiness management. The application layer is where the application are located, relying on the social-oriented behavior of the objects.

With the intent to highlight the distinctive features of the SIoT architecture, in Fig. 6 we provide a mapping of the major SIoT components into the reference model provided in [27] by the IoT-A European research project, which represents a major effort in defining a reference architecture for the existing and future IoT solutions. In this figure, we show the five functional layers with the main functionalities in grey-colored boxes. In the upper layer are located the applications that are built on top of an implementation of the IoT-A architecture. This represents instances of the *process execution* and *service orchestration*, which indeed is used to combine different services (also provided by different system implementations) to implement complex services. The relevant APIs are also a key part of this layer. The Virtual entity (VE) and information is the layer that maintains and organizes information related to physical entities, enabling search for services exposing resources associated to physical entities. Accordingly, it is intended to response to queries about a particular physical entity by providing with addresses of the service related to the physical entity. The lower layer, i.e., the IoT service & resource, links specific services to the related resources. It also notifies application software and services about events related to resources and corresponding physical entities. The Device connectivity and communication layer provides the set of methods and primitives for device connectivity and communication.

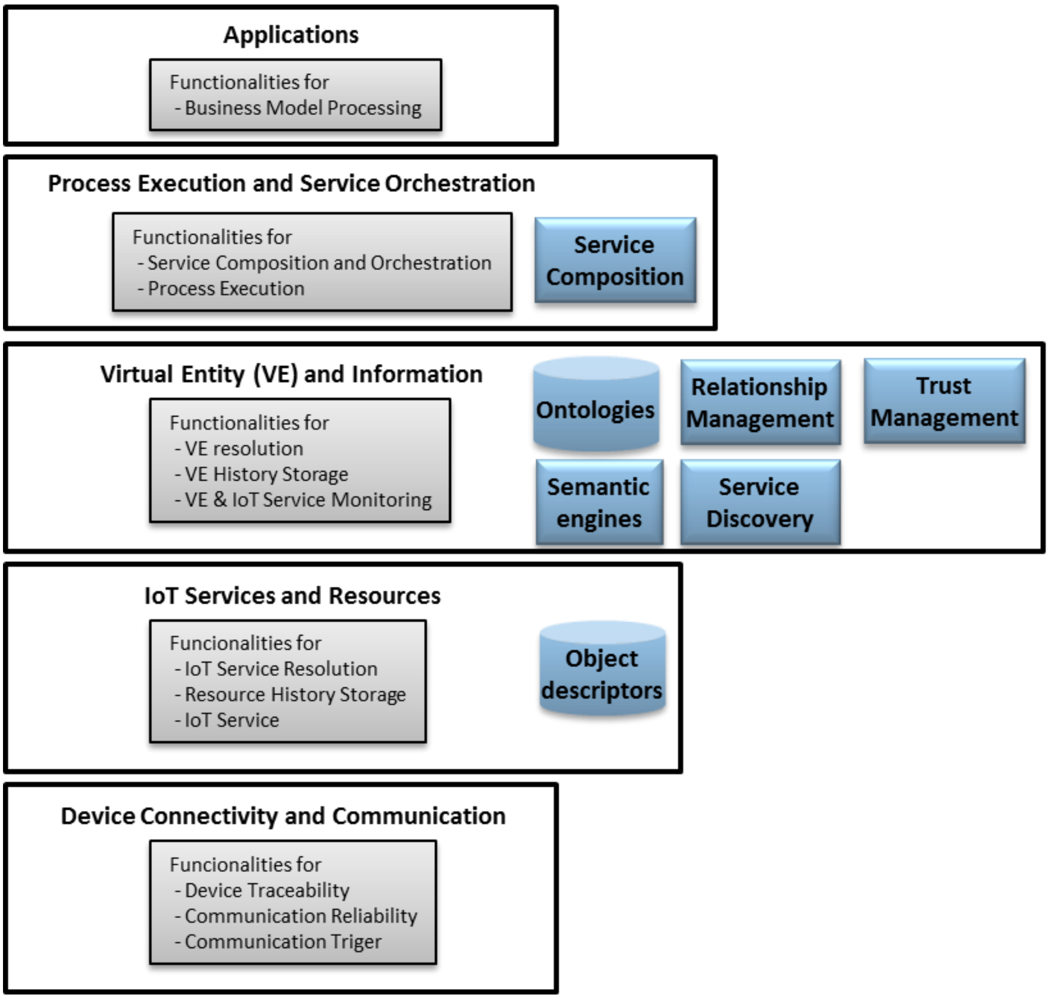

Fig 6. *Major functionalities of the SIoT system (in blue color) in the IoT-A reference architecture.*

In Fig. 6, we have highlighted the main features of the SIoT solution. At the second layer, the SIoT system may require the extension of the object descriptions to support the creation and management of the social-oriented behavior and relationships, such as: *owner ID*; *object position*, which can be changing over the time depending on the object mobility features; *power supply status*, that defines whether the object is either battery-powered (and the battery power level is provided), socket-connected (and whether is currently connected or not), or it harvests power from the environment; *amount of traffic generated* in terms of number of connections and overall bit-rate. The objects could also be grouped into different classes, depending on their main characteristics, such as mobility, computational and communication capabilities, interfaces, sensing capabilities, power supply.

At the third layer, the major part of the SIoT functionalities needs to be introduced. These also call for the definition of specific ontologies and semantic engines. The ontologies are used to represent a semantic view of the social activities, which is extracted through appropriate semantic engines. The objective is to provide a machine interpretable environment for representing attributes and operations of the IoT devices. Many works have already been conducted in this area, which could be partially used in the SIoT scenario, e.g., Ontology Web Language for Services (OWL-S) model, which has already been used as the basis of a semantic service modeling framework for the IoT [28][29]. Accordingly, services are used as an interface that represents the IoT resources (i.e., the physical world devices) and provide an access to the functions and capabilities of these resources. Ontologies to manage and control heterogeneous systems have also been investigated in [27]. This highlights that an automatic discovery will be impossible without ontological classification and semantic annotation processes. In [30], the importance of the ontology has been analyzed from a social network perspective as a format to represent the object information which is relevant to end users. Other approaches for creating semantic service descriptions could be used as described in [31]. These include: Semantic Annotations for WSDL (SAWSDL), Unified Service Description Language (USDL), Web Service Modelling Language (WSML), Web Service Modelling Ontology (WSMO), and Semantic Annotations for Representational State Transfer SA-REST [32].

The relationship management module is the one responsible to establish, update, and terminate the relationships among the objects in their virtual representation. It also elaborates the social network to facilitate the discovery of the services by identifying different clusters of resource relationships. Possible approaches are to make use of similarity measures and multi-scale renormalization and synergetic - self organization techniques. These should be embodied into the SIoT learners and adaptors of virtual resources following the cognitive networks paradigm. Consequently, an object social graph is built with objects that are linked by edges representing the established relationships, each one weighted by its computed degrees. This module is also responsible for calculating several topological features of the resource social graph such as: between-ness, closeness, degree/eigenvector hubs and authorities centrality measures. This allows for determining the most “central” nodes/resources, which is a key activity because the identified “central” nodes/resources will be those that control the data flows in SIoT and influence the rest of nodes/resources. By doing so, the “central” nodes will be marked as landmarks, representative for their local neighborhood.

The service discovery functionality is key in the third layer, since it is needed to find which objects can provide a service requested in the target application. This functionality should be implemented into the SIoT system by following the rules the humans adopt to seek for friendships and for any information in the social networking services. The discovery will be guided by the object social graph as described above. The trust management module is the one responsible to address the inherent risks in transactions with no prior experience with regard to the reputation of every other node. In doing this, it exploits the resource social graph to build a reputation-based trust mechanism for the IoT that can effectively deal with certain types of malicious behavior that intend to mislead other nodes [33]. In such a scenario, two possible models for the implementation of the Trustworthiness management can be followed. One is the *subjective trustworthiness*, derived from a social point of view, where each node computes the trustworthiness of its friends on the basis of its own experience and on the basis of its friends' experiences. If two nodes are not friends, then the trustworthiness is calculated by word of mouth through a chain of friendships. The other is the *objective trustworthiness*, obtained from P2P scenarios, where the information about each node is distributed and stored by making use of a DHT (Distributed Hash Table) structure. This information is visible to every node but is only managed by special nodes that we call Pre-Trusted Objects (PTOs).

Finally, the service composition component enables the interaction between services provided by different objects to achieve complex services and applications. Most of the time, the interaction is related to an object that wishes either to retrieve an information about the real world or to find a specific service provided by another object. In fact, the main potential we see in deploying SIoT is its capability to foster such an information retrieval.

5. On-going Projects

The projects listed here bring together the *technological aspects*, related to the functionality offered by the so-called smart objects, with the *social aspects*. Projects in this list mainly fall into one of the three scenarios emerged from the previous sections: *participatory sensing*, *device-in-the-circle*, and *devices-that-socialize*.

The tools to compose and build personalized and social application that merges together data from different sources belong to a field in continuous evolution and such tools are often the basis for other projects: they are mainly *cloud-based* because the cloud is always there, up and running, it is reachable from everywhere, and a large amount of application complexity can be moved from the target devices to the cloud components. Among these tools are: Ninja Blocks [34], IFTTT [35] and Paraimpu [36].

Ninja Blocks is a project started in 2012 from the crowdfunding platform Kickstarter.com. Ninja Blocks are cloud-based devices that can sense their environment and can act by controlling lights, power sockets, and other actuators. The system provides a tool to compose actions and sensing with common social web sites like Twitter, Facebook, Instagram. Among the sensors there is also a camera.

IFTTT is a San Francisco based startup whose service enables customers to create and share within minutes very simple applications that fit the "*if this then that*" rule. An example of application can be IF "someone tags me on Facebook picture" then "put the picture on my Dropbox". IFTTT was initially conceived for Internet services and social media and only later, in June 2012, the service crossed over to the physical world by integrating with Belkin WeMo [37] devices allowing IFTTT rules to compose social media events with home automation.

Paraimpu is a cloud-based research prototype from CRS4, which provides the functionality to manage smart Things and to compose them with services already on the Web to create personalized applications. Common DIY (Do It Yourself) boards like Arduino, can be easily integrated as the system automatically generates the code for these boards to produce/receive data. Paraimpu allows people to share smart things and devices in their social circle enabling social physical-virtual Web mash-ups.

The availability of such tools is fostering the development of personalized and interesting installation. At the moment these are generated by a small niche of geeks and early adopters but, to some extent, pave the way for a future diffusion and adoption of smart things in everyday social life. Among the installations, here are reported three "device-in-the-circle" cases:

- Jardimpu [38] (based on Arduino and Paraimpu) is an automated irrigation system based on the sensing of temperature, humidity, soil moisture, light conditions, which is then used to control the level of water in some plant's saucers (eg Dionaea Muscipula). This system is not an absolute novelty in the field of sensors-based garden irrigation, but it is one of the first examples of "social" gardening: people in the social circle can activate the drippers, can see plants in a live streaming, and can monitor their parameters.
- TLight is a permanent installation created by the Quit group [39] and connected to the web thanks to Paraimpu, which allows everyone to change the color tones and the behavior of the lights placed on the top of the big glass tower of the T-Hotel in Cagliari. To interact with the lights, a twitter user can post a message on with hashtag:#thotel followed by any phrase containing one of the following words: red, blue, green, orange, yellow, white, cyan, purple, wave, different, couple, full, pulse e random.
- Natural Fuse [40] is a social IoT game based on COSM [41]. Participants get a Natural Fuse unit which consists of a houseplant and a power socket. The amount of power available to the socket is limited by the capacity of the plant to offset the carbon footprint of the energy expended. If people cooperate on energy expenditure then the plants thrive (and everyone

> may use more energy). But if people don't cooperate, then the network starts to randomly kill plants. The electricity depends on the plants just as the plants depend on the electricity.

The work carried on in the field of Social Web of Things by Ericsson also deserves a citation as "devices-that-socialize" use case. Researchers at Ericsson in fact are working on methods for connecting device into a social media platform for nodes that aren't people. Recently they have presented at the Connected House exhibit, a social network interface that linked all of the different connected nodes in a home as well as trusted points in the public sphere.

Another very popular project is Waze. Waze is a company based on Israel which developed a service called social-GPS. The service is aimed at avoid traffic, and it is based on a large community spread all over the world. It allows, through a smart phone equipped with GPS, to share real-time traffic information and help everyone to save time and fuel. Even if it does not make use of any exotic hardware but it totally relies on the smartphone functionalities, it is a perfect example of tool for participatory sensing. At the same time, it is also a case of "device-in-the-circle" because the connection with Facebook allows participants to see their friends on the map to coordinate arrival times to give or get rides or meet up with other participants.

In the field of Smart Cities, the project CityScripts [42] is an experiment built on top of the SmartSantander [43] platform (with a base of 12000 urban sensors deployed in the city of Santander). The CityScripts project is aimed at integrating and experimenting a Web of Things scenario in which sensors and actuators in the city have a digital counterpart and can be used by citizens to compose personal applications integrating sensor data with social networks and other online data sources.

6. Conclusions

In this paper, we have reviewed the main approaches followed during the last years to make objects part of the human social loop and grant them a role within the human social network sites. This objective has been achieved by extending the paradigm of Web of Things to give things the capabilities to automatically post information on the social network sites and to be reached through these as well. A complementary approach is the one according to which objects have their own social network, which is independent from those of the humans but still controlled by them (albeit progressively less, as technologies progresses). The two approaches can be combined so that part of the information and actions of relevance for one social network type, that is the one where humans rule, can be exported/imported to/into the other one, where objects rule. Certainly, the objective of these approaches is to address the complexity in the management of the trillions of objects that will be connected to the network in a couple of years and to exploit their major potentialities. Through the presented survey of ongoing projects, we learned that the implementations deployed so far are limited in their potentialities. These only focus on specific applications and do not converge on an interoperable platform where social networking concepts are adopted as the major principles that drive the object interactions.

7. References

[1] O. Akribopoulos, I. Chatzigiannakis, C. Koninis, and E. Theodoridis, A web services-oriented architecture for integrating small programmable objects in the web of things, in Proceedings of the International Conference on Developments in eSystems Engineering , London, UK September 2010.

[2] D. Guinard, V. Trifa, T. Pham, and O. Liechti, Towards physical mashups in the web of things, in Proceedings of INSS’09, Pittsburgh, US, June 17-19, 2009

[3] http://www.ericsson.com/uxblog/2012/04/a-social-web-of-things/

[4] Chunhong Zhang, Cheng Cheng, Jang Ji, Architecture design for social web of things, Proceedings of the 1st International Workshop on Context Discovery and Data Mining, ContextDD '12, 2012

[5] A. Pintus, D. Carboni, A. Piras, Paraimpu: a Platform for a Social Web of Things, WWW 2012 – Demos Track, April 16–20, 2012, Lyon, France.

[6] L. Atzori, A. Iera, G. Morabito, Michele Nitti (2012) , The Social Internet of Things (SIoT) – When social networks meet the Internet of Things: Concept, architecture and network characterization, Elsevier Computer Networks, Volume 56, Issue 16.

[7] L. Atzori, A. Iera, G. Morabito (2011), “SIoT, Giving a Social Structure to the Internet of Things”, IEEE Communications Letters, vol. 15, p. 1193 -1195 , ISSN: 1089-7798,

[8] D. Guinard, A Web of Things Application Architecture – Integrating the Real-World into the Web, ETH Zurich, Zurich, Switzerland, 2011.

[9] S. de Deugd, R. Carroll, K. Kelly, B. Millett, e J. Ricker, SODA: Service Oriented Device Architecture, IEEE Pervasive Computing, vol. 5, n°. 3, pagg. 94–96, c3, 2006.

[10] L. M. . de Souza, P. Spiess, D. Guinard, M. Kohler, S. Karnouskos, e D. Savio, Socrades: A web service based shop floor integration infrastructure, Lecture Notes in Computer Science, vol. 4952, pag. 50, 2008.

[11] OASIS Devices Profile for Web Services (DPWS). [Online]. Available: http://docs.oasis-open.org/ws-dd/ns/dpws/2009/01. [Accessed: 08-Nov-2012].

[12] A. S. Shirazi, C. Winkler, e A. Schmidt, SENSE-SATION: An extensible platform for integration of phones into the Web, in Internet of Things (IOT), 2010, 2010, pagg. 1–8.

[13] D. Guinard, V. Trifa, T. Pham, e O. Liechti, Towards physical mashups in the web of things, in Proceedings of INSS, 2009.

[14] C. Pautasso, O. Zimmermann, e F. Leymann, Restful web services vs. big’web services: making the right architectural decision, in Proceeding of the 17th international conference on World Wide Web, 2008, pagg. 805–814.

[15] Z. Shelby, K. Hartke, C. Bormann, e B. Frank, Constrained application protocol (coap), draft-ietf-corecoap-07, 2011.

[16] Yaler - a simple, open and scalable relay infrastructure. [Online]. Available: http://www.yaler.org/. [Accessed: 08-Nov-2012].

[17] G. Mulligan, The 6LoWPAN architecture, in Proceedings of the 4th workshop on Embedded networked sensors, 2007, pagg. 78–82.
[18] D. Kegel, The C10K problem. 2006.
[19] A little holiday present: 10,000 reqs/sec with Nginx! | WebFaction Blog. [Online]. Available: http://blog.webfaction.com/2008/12/a-little-holiday-present-10000-reqssec-with-nginx-2/. [Accessed: 08-Nov-2012].
[20] A. Pintus, D. Carboni, e A. Piras, The anatomy of a large scale social web for internet enabled objects, in Proceedings of the Second International Workshop on Web of Things, 2011, pag. 6.
[21] A. Pintus, D. Carboni, A. Piras, e A. Giordano, Connecting Smart Things through Web Services Orchestrations, in Current Trends in Web Engineering, vol. 6385, F. Daniel e F. M. Facca, Cur. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pagg. 431–441.
[22] Nicolas Maisonneuve, Matthias Stevens, and Bartek Ochab. 2010. Participatory noise pollution monitoring using mobile phones. *Info. Pol.* 15, 1,2 (April 2010), 51-71
[23] M. Blackstock, R. Lea, e A. Friday, Uniting online social networks with places and things, in Proceedings of the Second International Workshop on Web of Things, 2011, pag. 5.
[24] P. Mendes, Social-driven internet of connected objects, in Proc. of the Interconn. Smart Objects with the Internet Workshop, 2011.
[25] L. Ding, P. Shi, and B. Liu, The clustering of internet, internet of things and social network, in Proc. of the 3rd International Symposium on Knowledge Acquisition and Modeling, 2010.
[26] E. A. K. amd N. D. Tselikas and A. C. Boucouvalas, Integrating rfids and smart objects into a unified internet of things architecture, Advances in Internet of Things, vol. 1, no. 1, pp. 5–12, 2011.
[27] Internet-of-Things Architecture IoT-A Project, Deliverable D1.2 – Initial Architectural Reference Model for IoT, 2011
[28] S. De, P. Barnaghi, M. Bauer, S. Meissner, Service modelling for the Internet of things, in: Proc. of the Federeted Conference on Computer Science and Information System, September 2011.
[29] A. Katasonov, O. Kaykova, O. Khriyenko, S. Nikitin, V. Terziyan, Smart semantic middleware for the Internet of things, in: Proc. of the 5th International Conference on Informatics in Control Automation and Robotics, May 2008.
[30] J. Breslin, S. Decker, The future of social networks on the Internet, IEEE Internet Computing 11 (6) (2007) 86–90.
[31] D. Guinard, M. Mueller, J. Pasquier, Giving RFID a REST: building a web-enabled EPCIS, in: Proc. of Internet of Things 2010 Conference, November 2010.
[32] R. Studer, S. Grimm, A. Abecker (Eds.), Semantic Web Services, Concepts, Technologies, and Applications, Springer Verlag, 2007.
[33] M. Nitti, R. Girau, L. Atzori, A. Iera, and G. Morabito, A Subjective Model for Trustworthiness Evaluation in the Social Internet of Things, International Workshop on Internet-of-Things Communications and Networking, IEEE PIMRC, Sidney, Australi, September 2012.
[34] Ninja Blocks - The API for Atoms. [Online]. Available: http://ninjablocks.com/. [Accessed: 24-Nov-2012].
[35] IFTTT / Put the internet to work for you. [Online]. Available: https://ifttt.com/. [Accessed:

24-Nov-2012].
[36] Paraimpu - The Web of Things is more than Things in the Web. [Online]. Available: http://paraimpu.crs4.it/. [Accessed: 24-Nov-2012].
[37] WeMo | Belkin USA Site. [Online]. Available: http://www.belkin.com/us/wemo. [Accessed: 24-Nov-2012].
[38] jardimpu. [Online]. Available: http://jardimpu.blogspot.it/. [Accessed: 24-Nov-2012].
[39] INSTALLATION: quit. [Online]. Available: http://www.quit-project.net/?page_id=27. [Accessed: 24-Nov-2012].
[40] Natural Fuse: home / map. [Online]. Available: http://www.naturalfuse.org/. [Accessed: 24-Nov-2012].
[41] Cosm - Internet of Things Platform Connecting Devices and Apps for Real-Time Control and Data Storage. [Online]. Available: https://cosm.com/. [Accessed: 24-Nov-2012].
[42] CityScripts. [Online]. Available: http://cityscripts.crs4.it/. [Accessed: 24-Nov-2012].
[43] SmartSantander. [Online]. Available: http://www.smartsantander.eu/. [Accessed: 24-Nov-2012].