# Drivers, Standards and Platforms for the IoT: Towards a digital VICINITY

Aida Mynzhasova[1], Carna Radojicic[1], Christopher Heinz[1], Johannes Kölsch[1], Christoph Grimm[1], Juan Rico[2], Keith Dickerson[3], Raúl García-Castro[4], and Victor Oravec[5]

[1]Technical University of Kaiserslautern, Germany
[2]Atos Research and Innovation, Spain
[3]Climate Associates Limited, United Kingdom
[4]Universidad Politécnica de Madrid, Spain
[5]bAvenir, s.r.o., Slovakia
Emails: {mynzhasova, radojicic, heinz, koelsch, grimm}@cs.uni-kl.de, juan.rico@atos.net
keith.dickerson@mac.com, rgarcia@fi.upm.es, victor.oravec@bavenir.eu

***Abstract*—The Internet of Things is created by networking many different kind of things, enabling new services and business models. However, things from different manufacturers, various domains, and a number of standards have to interact. Unfortunately, the current situation is characterized by 'silos' or 'islands' that lack interoperability. This paper gives a survey and analysis of drivers, platforms, and standards for the Internet of Things (IoT) that provide a basis for interoperability. They are considered for requirements of the VICINITY project whose goal is to offer "Interoperability as a Service".**

***Keywords—Internet of Things; IoT; Interoperability; Standards***

## I. Introduction

The "Internet of Things (IoT)" has become a quickly growing topic that has the potential to change the way people live, work, and think. The vision of ubiquitous, networked devices was first formulated by Mark Weiser [1]. The concrete term "Internet of Things" was coined by Kevin Ashton [2], initially focussing on RFID tags. The International Telecommunication Union (ITU) defines the Internet of Things as *"a global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies"* [3], and *" (...) things include the surrounding environment, industrial robots, goods and electrical equipment"* [3].

Following the ITU's definition of the IoT, the creation of advanced services differentiates the IoT from well-known communication systems, e.g. in automation systems or wireless sensor networks. As data in the IoT stems from a variety of heterogeneous things from different domains, interoperability is a major challenge for the creation of advanced services. The objective of this paper is to define the basis for the creation of advanced concepts of interoperability, covering various layers, starting with business- and application layer, semantic interoperability, down to technical interoperability.

The rest of the paper is organized as follows. In the remainder of this section we describe a typical IoT application architecture. In Section II we summarize the outcomes of a survey identifying drivers and barriers from the point of view of various stakeholders. In Section III we give an overview of standards that strive to provide semantic interoperability. In Section IV we survey the software platforms that provide technical interoperability, e.g between ZigBee and Fieldbus protocols, and hardware platforms on which this can be implemented. The last section closes the paper and identifies the future work.

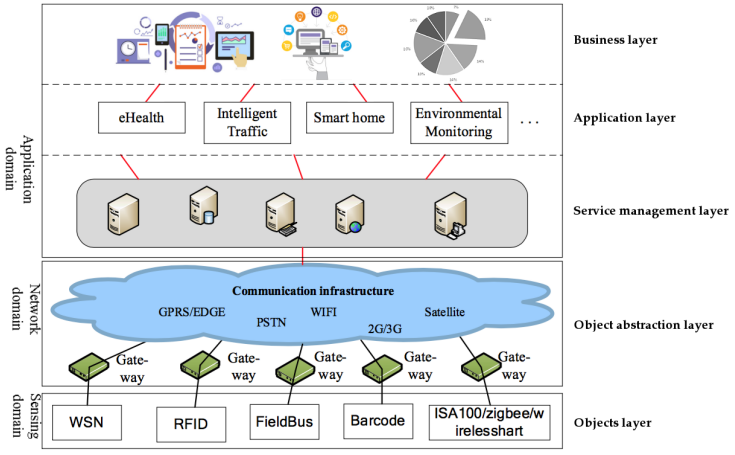


Fig. 1: A 5-layer IoT Architecture

### A. Layers of IoT Architectures

To simplify the complexity of IoT architectures, they are structured in layers that, stepwise, abstract data to semantic information as a foundation for applications and services. At least three layers are used as shown in Fig. 1: sensing (also perception or device or objects), network (or transmission), and application layer [4], [5]. In [6] the common 3-layer architecture is extended with two additional layers: service management and business layer.

The *sensing (perception) layer* is the lowest level. Its main responsibility is to collect data, or to control actuators. It uses

technologies such as RFID, ZigBee, or wired field buses for communication. This is also the layer where protocol transformation is performed and middleware for sharing multiprotocol data is provided.

The responsibility of the *network layer* is to transfer data collected from the sensing layer to remote destinations via the Internet. For this purpose, technologies such as Ethernet and GPRS are used.

The *application layer* processes information and implements services. Depending on the needs of the user, this layer covers many kinds of application domains, including smart home, smart transportation, eHealthcare, etc. [7].

The heterogeneity and complexity of the IoT requires further means to structure data, to create (semantic) information out of it, and to manage services. This is done at the *service management layer*. It pairs service with its requesters, stores data from the network layer, enriches it with semantic data, makes decisions, processes information and passes it to the application layer [6].

Technically, the sensing layer is implemented by any kind of thing that provides data; the transition to the network layer including protocol transformation is usually implemented in (IoT) gateways, and the higher layers are implemented on the server side.

The basic or extended IoT architecture identifies two main levels where IoT interoperability is needed: gateway-based interoperability of devices at lower (device) level and interoperability at higher, semantic level that allows communication of "things" on the service management layer. IoT gateways are universal devices that allow translation of different communication protocols to a common protocol that is required for communication with the cloud on the server side. The key to enabling communication between "things" is interoperability achieved through standardisation. The aim should always be to re-use existing standards wherever possible and to propose extensions to these if they cannot be used as they are.

### *B. VICINITY: Interoperability as a service*

The H2020 project VICINITY, started in January 2016, aims at providing a decentralized bottom-up ecosystem that offers "Interoperability as a service" [8]. In VICINITY, the users share – like in social networks – access to their devices and the data these devices gather and produce. On a local gateway device a VICINITY Agent is running, together with adapter modules, taking care of the necessary translation between encrypting and enriching shared objects with metadata, in order to enable interoperability as a service to existing IoT deployments. VICINITY does not aim to introduce yet another standard for the IoT. Instead, a strong emphasis is put on monitoring, adapting and contributing to existing standards.

## II. Drivers and Barriers for the IoT

IoT systems generate a large amount of data that can be collected for the generation of novel services. By sharing thoughts and concerns with 150 stakeholders, the VICINITY project identified the key aspects that can boost or delay the adoption of IoT based solutions and services. The drivers and barriers collected are quite heterogeneous but there are two main groups. The first group covers the technical foundations of the solution while the second covers non-technical aspects such as the social perception, the regulatory framework and the potential impact on existing and emerging business. According to this classification we summarise the technical and nontechnical barriers and drivers identified in the project [9], which are shown in Fig. 2.

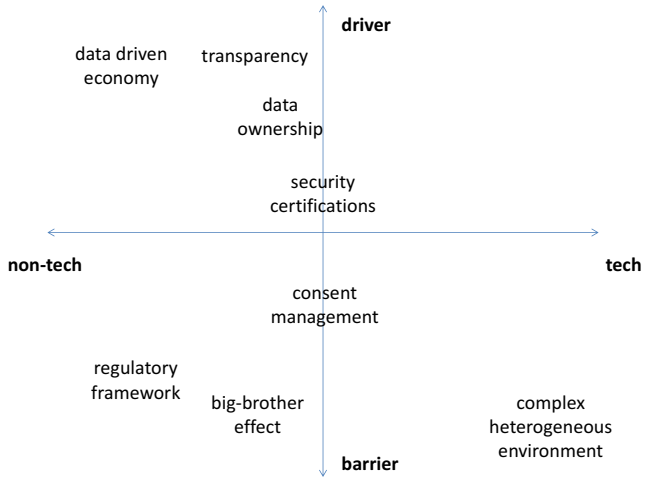


Fig. 2: Barriers and drivers identified in the VICINITY Project

The technical drivers identified in VICINITY are:

- Enabling simple access to a heterogeneous, already complex environment. To simplify and facilitate how this happens today is a major challenge whose solution will boost the adoption of Interoperability as a Service.
- End-to-End encryption which will increase the perception of creating and participating in a secure environment where everything is under control of the users.
- Cloud technologies act as a relevant driver since the cost derived from the deployment of the solution will decrease without affecting functionality and performance.

On the other hand, there are also technical barriers that need to be taken into account:

- Dealing with the consent of users in a systematic way is a two-fold barrier. From the technical perspective, it is necessary to create a general framework that can be applied to multiple scenarios, while being simple enough to allow anyone to take part in it and clearly understand the impact of their choices.
- In addition to the complex standards environment, it is also a challenge to select the front-end interfaces that are used for a similar purpose. This multiplicity is a barrier since the lack of a common approach requires education and training.
- The different languages that are used in the different sectors for naming similar things requires the definition of common ontologies and their mapping to real

life scenarios. The health domain is a clear example of how different zones deal in a different way with the same problem.

Regarding the non-technical aspects, the main drivers identified are:

- Use of incentives for promotion plays a large role in user adoption of technology and in ease of use. Since the concept presented is something new that aims at empower citizens, it is important to offer them the right incentives for a first contact with the novel solution.
- Security certifications and labelling will increase the reliability of the system and the perception of its users inviting them to take part in a trusted environment.
- Transparency, the impact of active monitoring in the different application areas can be perceived as an intrusion in the users' life. Dealing with this new scenario with transparency will facilitate the adoption of the technology.
- IoT and the provision of a common framework for accessing information in a simple manner will accelerate the development of new business as part of the data driven economy.
- The ownership of data by users and not storing such data into centralized solutions represents a big advantage.

Non-technical barriers include:

- The regulatory framework nowadays is quite vast and heterogeneous, thus it is mandatory to be compliant with all existing regulations even when these aspects are different in the different countries.
- Some applications look promising. However, there are security aspects that need to be considered, for example, sharing indoor parking implies granting access to private properties which represents a risk.
- Access to health data is really crucial in emergencies, nevertheless the identification of those scenarios must be really clear for allowing access to users' sensitive data only to entities covered by a consent agreement.

## III. IoT Standards

The interaction with standards in a research project is always challenging and, in the case of IoT, interoperability is totally influenced by the standards landscape. Nevertheless, the influence that standards have over project activities and the return that the activities developed in VICINITY could have requires a detailed action plan as shown in Fig. 3 and documented in [10].

Standards recommendations are divided into two main groups. The first covers standards in which VICINITY partners are involved and will continue to be in the future. The following are priorities for VICINITY to contribute:

- AIOTI – Alliance for Internet of Things Innovation. AIOTI was set up by the EC in early 2015 in an attempt to generate a consensus on the standards needed to deploy the IoT globally. The following working groups are relevant to VICINITY: WG3 (Standardisation), WG4 (Policy), WG5 (Smart Living for Ageing Well), WG7 (Wearables), WG8 (Smart Cities), WG9 (Smart Mobility), WG12 (Smart Energy), and WG13 (Smart Buildings and Architecture).
- ETSI – European Telecommunications Standards Institute. Two different groups are relevant to VICINITY. The first one is the SmartM2M Technical Committee, which was set up to develop specifications for M2M services and applications focussing on IoT and Smart Cities. SmartM2M is in charge of maintaining the Smart Appliances Reference Ontology (SAREF) and its extensions, as well as of aligning SAREF to the oneM2M[1] Base Ontology. Currently, the second version of the SAREF ontology has been developed[2] as well as extensions in three domains: energy[3], environment[4], and building[5]. The second one is the recently-founded Industry Specification Group relating to cross-cutting Context Information Management (ISG-CIM), which has the goal of developing technical specifications and reports to enable multiple organizations to develop interoperable software implementations of a cross-cutting Context Information Management layer.
- World Wide Web Consortium – W3C. The World Wide Web Consortium (W3C) is the main standardization body that develops standards for the Web. Two of its main activities are relevant to VICINITY: the Web of Data and the Web of Things (WoT). From the Data activity (and its predecessor the Semantic Web one), VICINITY will use the set of standards

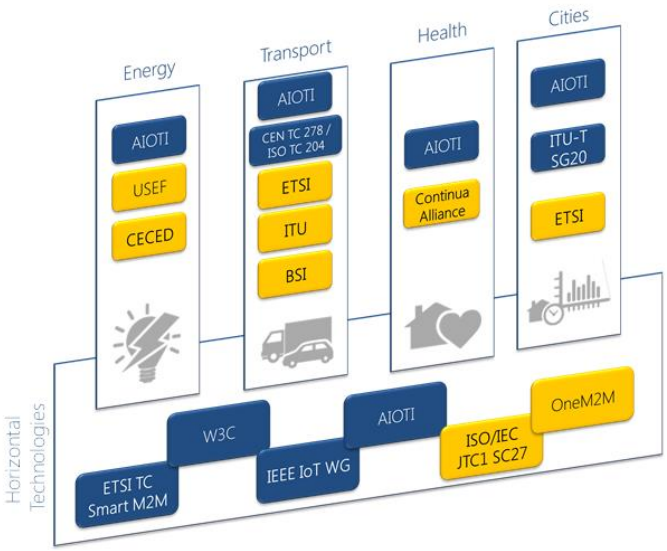


Fig. 3: Landscape of IoT standards considered for VICINITY

[1]OneM2M: online available: http://www.onem2m.org/technical/latest-drafts
[2]SAREF online available: https://w3id.org/saref
[3]SAREF4ENER online available: https://w3id.org/saref4ener
[4]SAREF4ENVI online available: https://w3id.org/def/saref4envi
[5]SAREF4BLDG online available: https://w3id.org/def/saref4bldg

defined for representing data on the Web (Resource Description Framework, RDF), describing ontologies that give meaning to such data (Web Ontology Language, OWL), querying such data (SPARQL Protocol and RDF Query Language), and providing REST interfaces to access them (Linked Data Platform, LDP). One of the current groups of relevance to VICINITY is the Spatial Data on the Web Working Group, jointly chartered between the W3C and the Open Geospatial Consortium (OGC), that aims to guide on how to represent and use different types of data that are relevant in the IoT domain, i.e., spatial, temporal, sensor, and coverage data. The Web of Things Interest Group, and its continuation in the Web of Things Working Group, are also key to VICINITY since they aim to enable interoperability across IoT platforms by standardizing data models, interfaces, bindings, and security and privacy policies and mechanisms.

- ITU-T SG20 – IoT and applications including smart cities and communities. SG20[6] has drafted an IoT Standards Roadmap. The services offered and the objectives of the VICINITY trials could be contributed to SG20 with a view to developing new ITU Recommendations or Supplements.
- IEEE IoT WG – P2413 Architectural Framework for the IoT.
- CEN TC 278 / ISO TC 204 – ITS Standards. CEN TC 278 is heavily integrated with ISO TC 204 and covers Transport Telematics and Traffic, which may be relevant to the Smart Grid and Parking VICINITY use case [10].

The second group includes the standards that will be monitored during the whole duration of the project; no active contribution is needed:

- ISO/IEC JTC1 SC27 (Information Security) and WG10 (Internet of Things). ISO/IEC SC27 covers development of standards for the protection of information and ICT. This includes generic methods, techniques and guidelines to address both security and privacy aspects summarized in [10].
- oneM2M Partnership Project – base ontology. The structure of oneM2M can be found in [11]. The work on semantics/ontologies is carried out in WG MAS (Management Abstraction and Semantics). This group is developing a Base Ontology based on the requirements of specific ontologies such as SAREF.
- Continua Health Alliance. The Continua Health Alliance standard addresses the fundamentals of data exchange between medical devices [12]. The Continua Health Alliance addresses Personal Connected Health and is required for the VICINITY eHealth at Home pilot.
- USEF – Universal Smart Energy Framework. The USEF Foundation develops, maintains and audits the USEF framework. USEF partners are working together to deliver the foundations of one integrated system which benefits all players - new and traditional energy companies and consumers.
- CECED – European Committee of Domestic Equipment Manufacturers.
- ITS (Information Technology Services) groups in ETSI, ITU (International Telecommunication Union), BSI (British Standards Institution).
- ETSI Board and SSCC-CG (Smart and Sustainable Cities and Communities - Coordination Group).

In addition, the standards groups are classified according to the nature of the standards under development. The first group is related to the requirements of the underlying technology and foundations of the IoT ecosystem, grouped as horizontal technologies. A second group covers the standards that support vertical use cases (see Fig. 3).

## IV. Hardware and Software Platforms for the IoT Gateway

The IoT gateway implements the interface between the sensing layer and the higher layers. The challenge is to merge a variety of non-IP protocols into the higher-level IP protocols, and to abstract from the heterogeneity of the sensing layer, providing a homogeneous view to the higher levels (object abstraction layer). The IoT gateway can also implement parts of the service layer. The focus of the IoT gateway is technical interoperability at the object abstraction layer.

### *A. Software platforms for object abstraction*

In the following we describe and discuss software platforms that provide, to some extent, the above-mentioned functionality of the IoT gateway. OpenHAB, AllJoyn, OpenRemote, DeviceHive, and IoTivity were selected as they are the most popular platforms providing object abstraction amongst other features:

1) **OpenHAB SmartHome Framework**[7]. This open source project is one of the most widely known solutions for IoT gateways. It was developed to simplify home automation, but because of its modular OSGi architecture and the variety of supported protocols it can be also used as an IoT gateway. The openHAB runtime environment is a set of OSGi bundles deployed on the OSGi framework Equinox. It is therefore a pure Java solution and needs only a JVM to run[8]. Its modular structure allows adding and removing functionality during runtime without the need to stop the service. Functionality includes keeping track of the status of items, communication between services, sending commands to items. Everything communicates through an openHAB Event Bus so the connection between openHAB instances is kept as low as possible.
OpenHAB is vendor-neutral and protocol agnostic. It can be easily accessed with mobile phone or web applications.

[6]Online available: http://www.itu.int/en/ITU-T/jca/iot/Pages/default.aspx

[7]Openhab online available: http://www.openhab.org/features/introduction.html

[8]More info on: https://github.com/openhab/openhab/wiki

2) **AllJoyn**. Alljoyn is a project supported by the AllSeen Alliance and is strong enough to evolve to a worldwide standard. It can be used on all modern operating systems, since it offers an abstraction layer for Android, IOS, Linux and Windows. AllJoyn offers features such as easy discovery and group formation, the ability to share control among devices and applications and easy extendibility to integrate new protocols.
3) **DeviceHive**[9] **IoT Framework**. DeviceHive was mainly designed to enable message exchange between smart devices and client applications. DeviceHive contains cloud services, open source server and client libraries, protocol adapters, examples, documentation, management system. DeviceHive is an AllSeen Alliance member, so is compatible with AllJoyn, providing cloud connectivity for AllJoyn devices and also bridges third party protocols into AllJoyn expanding the ecosystem of supported devices. Thus, AllJoyn support comes out of the box and can be customized for various integration scenarios. DeviceHive is well suited for usage in enterprise solutions and works in public and private clouds such as OpenStack, Microsoft Azure, and own data centers.
4) **OpenRemote**[10]. As an open source project, OpenRemote (started 2009) addresses the problems caused by attempts at integration between different protocols and existing M2M communication solutions. Its main parts are an online designer, controllers and panel or Android/iOS custom application. Online Designer provides help to the Building Modeller to configure devices and internet services, define macros and write rules to automate the IoT system. Another part of online Designer, UI Designer, provides a way to design a user interface for a Web Browser, iOS or Android. The OpenRemote Controller actually represents the gateway - it connects devices and services and runs the designed automation scripts. For controller purposes a variety of hardware can be used, for example, Raspberry Pi or BeagleBone. The idea of the panel or Android/iOS custom application is to connect to the controller and display designed in online Designer interface. Users can see the status of connected devices or services via these Apps or a browser connection, and control the system using buttons, sliders, etc. High scalability potential makes possible the usage in Industry, Health Care, Smart Cities.
5) **Iotivity**[11]. The Iotivity project was created in 2015. Although younger than Alljoyn, it has the potential to become complementary to the Alljoyn framework. Iotivity has advanced discovery mechanisms and provides transparent data and device management possibilities. It can be installed on Android, Linux, IOS and Windows. Compared to Alljoyn, Iotivity offers a more simplified API to create Iotivity compliant Servers and Clients.

A direct comparison of these software platforms is shown in Table I.

### *B. Hardware Platforms*

Nowadays the electronics market offers all kinds of hardware devices, starting from simple microcontroller boards to complex hardware development kits. They mainly follow the requirements of low power and small size. To a large extent these requirements are fulfilled by communication technologies and standards such as radio-frequency identification (RFID), quick response (QR) codes, Bluetooth low energy (BLE) technology, WiFi direct, IEEE 802.11ah (HaLow) and in addition new technologies such as Z-Wave, ZigBee or HomePlug, which pursue the same challenges. As a result, the world of embedded devices is full of different kinds of smart devices, which use a wide range of communication protocols and standards. The problem of their integration is supposed to be solved by so called IoT gateways. For use as an IoT gateway, a universal device is needed that is fast, reliable, with a huge number of interfaces (or easily extendable), energy-saving and cheap. On top of this hardware, a software framework/platform is required in order to enable: device discovery, plug-and-play identification of (device) services, identification-based connectivity, technical Interoperability, security, privacy, manageability and location-based capabilities.

In the following we give an overview of hardware platforms that have been considered as candidates for VICINITY requirements [10]. Table II summarizes the hardware platforms considered, comparing their specifications.

Five possible candidates are being taken into account:

1) **Banana Pro**[12]. An open-source single-board computer, developed by LeMaker (China) and released in October 2014. It is based on AllWinner A20 System on a Chip (SoC), has 1GB of DDR3 memory and can run a large number of different operating systems as Lubuntu, Android, Debian, Bananian, etc.
2) **Cubieboard 3 (Cubietrack**[13]). An open-source single-board computer released in October 2013 by CubieTeam (China). Like Banana Pro it is also based on AllWinner A20 SoC, but models with 2GB DDR3 memory also exist. The supported operating systems are: Android, Cubieez, Lubuntu and Fedora.
3) **Raspberry PI 3 Model B**. Nowadays this platform is the most famous single-boarded computer developed in the United Kingdom by the Raspberry PI Foundation. It has 1GB of DDR2 memory, and in comparison with Banana Pro and Cubieboard 3, it is based on 64-bit Broadcom SoC of the next generation.
Raspberry PI is designed to run the Raspbian operating system, but there Ubuntu, Pidora, OpenELEC, RISC OS and Windows 10 are also possible OS. It is most likely to continue with potential future revisions of this series, making it ideal when it comes to upgrades.
4) **Pine A64**[14]. The young project of a single-board computer, developed by USA. It was released in

[9]DeviceHive online available: http://devicehive.com
[10]Openremote online available: http://www.openremote.com
[11]Iotivity online available: https://www.iotivity.org
[12]LEMAKER Banana ProSpecification: http://www.lemaker.org/product-bananapro-specification.html
[13]Cubietruck online available: http://www.cubietruck.com
[14]Pine A64 online available: https://www.kickstarter.com/projects/pine64/pine-a64-first-15-64-bit-single-board-super-comput/description

TABLE I: Comparison of Software Frameworks [10]

| Name | OpenHAB | Device Hive | OpenRemote | AllJoyn | Iotivity |
|---|---|---|---|---|---|
| Platform | JVM | POSIX compliant | JVM | Linux, Windows, Mac, Android | Linux, Android, IOS, Windows |
| Development language | Java | C/C++, Go, Java, Python, Bash | Java | C, C++, Java, objective-C | C/C++, Java, Android, JavaScript (in future) |
| Documentation clarity | 4/5 | 4.5/5 | 5/5 | 4.5/5 | 5/5 |
| Alliance members, partners, users | EnOcean Alliance, AllSeen Alliance, Eclipse Foundation | AllSeen Alliance | TU Eindhoven, NEEO, Philips, Trust, etc. | AllSeen Alliance | Open Internet Consortium |
| Licence | Eclipse Public Licence | MIT Licence | GNU AGPL | ISC Licence | Apache Licence 2.0 |

TABLE II: Comparison of Available Hardware Platforms [10]

| Name | Banana Pro | Cubieboard (Cubietruck) | Raspberry Pi 3 Model B | PINE64+2GB | Intel Edison (EDI2ARDUIN.A.L.K) |
|---|---|---|---|---|---|
| SoC | Allwinner A20 | Allwinner A20 | Broadcom | Allwinner A64 | Intel®Atom$^{TM}$ processor |
| CPU | 1 GHz ARM Cortex-A7 Dual Core | 1 GHz ARM Cortex-A7 Dual Core | 1 GHz ARM Cortex-A53 Quad Core | 64bit Quad Core ARM A53 1.2 GHz CPU | Dual Core processor at 500 MHz |
| Memory | 1 GB DDR3@432 MHz | 2 GB DDR3@480 MHz | 1 GB DDR2@450 MHz | 2GB DDR3 | 1 GB DDR3 RAM, 4 GB eMMC Flash |
| Graphic engine | Mali400MP2, compatible with OpenGL ES 2.0/1.1 (hardware acceleration support) | Mali400MP2 | Broadcom VideoCore IV | Mali400MP2 | - |
| Audio output | Yes | Yes | Yes | Yes | No |
| Audio input | Yes | Yes | No | Yes | No |
| HDMI | Yes | Yes | No | Yes | No |
| Camera Interface | Yes, 1 x Parallel 8-bit camera interface | No, USB-camera is possible | Yes | Yes | GPIO interface |
| Micro-SD slot | Yes | Yes | Yes | Yes | Yes |
| Expansion header | 40-pin header, 28xGPIO, can be used for UART,I2C, SPI, PWM, CAN, I2S, SPDIF | 54 pins including I2S, I2C, SPI, CVBS, LRADC x2,UART, PS2, PWM x2, TS/CSI, IRDA, LINEIN&FMIN&MICIN, TVIN x4 with 2.0 pitch connectors | 40-pin GPIO | Euler "e" bus, Raspberry Pi 2 Bus | 40 GPIO interface |
| External interface | 2 x USB Host 1 x USB OTG | 2 x USB Host 1 x USB OTG | 4 (from 5-port USB-hub) | 2 x USB 2.0 host port | 2x USB 2.0 |
| SATA | 2.0 | 2.0 | No | No | No |
| Ethernet | 10/100/1000 Mbps | 10/100/1000 Mbps | 100 Mbps | 1 Gbps | No |
| Wifi | Yes | Yes | Yes | Yes | Yes |
| Bluetooth | No | Yes | Bluetooth 4.1 BLE | Yes | Bluetooth 4.0 |
| ZigBee | No | No | No | No | No |
| Other | IR receiver | IR receiver, 8 GB NAND-Flash | | 3-pin connector for lithium battery, 2-pin connector for RTC clock, Touch panel connector, Display DSI connector, optionally Z-Wave | UART, I2C, I2S, GPIO Additional 12 (with 4 capable of PWM) |
| Price | 43 - 46 € | 93 - 104 € | 37.50 € | 25 € | 100 € |

February in 2016, and it uses AllWinner A64 SoC with the same architecture as the Broadcom SoC in Raspberry PI. It supports 1GB or 2GB of DDR3 memory. The main advantages of this platform are its low price, a full compatibility with Raspberry PI 2 extension boards and the capability to run Linux and Android as an operating system.

5) **Intel Edison**. This is the response of Intel to the IoT ecosystem of different mini-pc options in the market. It is a good alternative to Raspberry Pi with the advantage of using x86 chipsets which facilitate developments and deployment of software. It is covered with a wide community of support and developers. It provides even 40 GPIO interfaces that allow to prototype IoT devices and gateways.

The Raspberry Pi 3 has most likely the largest support in the open-source community and might be the best choice to attract most stakeholders to VICINITY. However, a final decision will be made after the requirements of the VICINITY pilot sites are identified.

## V. Conclusion

In the paper, we have summarized interviews with 150 stakeholders from the IoT ecosystem and highlighted the drivers and barriers that were identified. Major drivers are the management of resources, while data management including privacy and regularity compliance have been identified as major barriers besides the technical challenge of interoperability.

For interoperability at the layer of communication protocols a number of hardware and software platforms is available. These enable interoperability between a number of nodes as long as heterogeneity is not increasing and the complexity of the network can be managed manually. However, once complexity and heterogeneity grow beyond that limit, interoperability at the semantic layer is required. This is provided by an increasing number of standards which is motivating solutions that provide "Interoperability as a Service" such as the VICINITY project.

## VI. Future Work

As this work has highlighted the motivation behind "Interoperability as a Service", in the near future, the VICINITY project has to come up with an architectural design to support this claim. A draft of the proposed architecture is shown in Fig. 4. As described in Sec. IV, IoT gateway need to provide object abstraction to higher layers. In the case of VICINITY, these higher layers should implement *VICINITY Nodes*, which grant a uniform access to IoT devices and their data "as a Service". This information is only securely transmitted to peers with whom it was shared, and never stored globally in a cloud. This grants full control to the users and ensures privacy by design. Only for special purposes like e.g. discovery, authentication/authorization etc. a global VICINITY cloud is required to interact with the gateway devices directly. To enable communication within the VICINITY, a gateway API is also defined and developed by the VICINITY project in the near future.

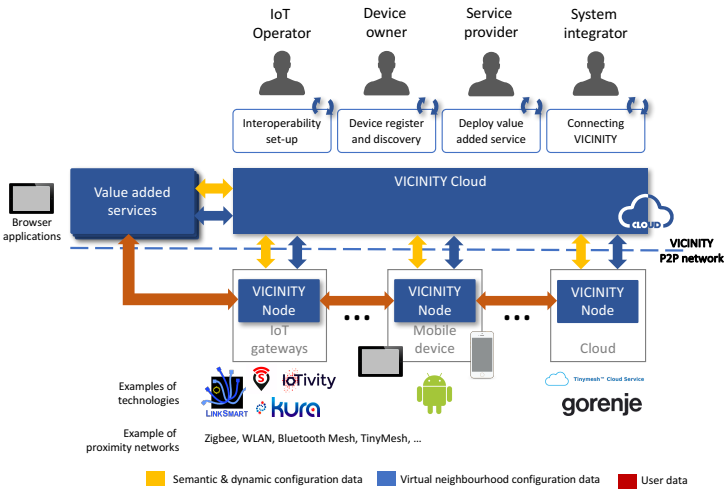


Fig. 4: Draft of VICINITY's architecture

## Acknowledgment

As a part of the VICINITY project, this work has been supported by EU (European Union) program Horizon 2020 under grant agreement number 688467.

## References

[1] M. Weiser, "The Computer for the 21st Century," *ACM SIGMOBILE Mobile Computing and Communications*, vol. 3, no. 3, pp. 3–11.

[2] K. Ashton, "That 'Internet of Things' Thing," *RFID Journal*, 2009.

[3] I. T. Union, "Overview of the Internet of Things," *Recommendation ITU-T Y.2060*, June 2012.

[4] Q. Zhu, R. Wang, Q. Chen, Y. Liu, and W. Qin, "IOT Gateway: Bridging Wireless Sensor Networks into Internet of Things," in *IEEE/IFIP International Conference on Embedded and Ubiquitous Computing*, 2010, pp. 347–352.

[5] H. Chen, X. Jia, and H. Li, "A brief introduction to IoT gateway," in *Communication Technology and Application (ICCTA 2011), IET International Conference on*, 2011, pp. 610 – 613.

[6] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A Survey of Enabling Technologies, Protocols and Applications," *IEEE COMMUNICATION SURVEYS & TUTORIALS*, vol. 17, no. 4, pp. 2347–2376, 2015.

[7] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things Architecture, Possible Applications and Key Challanges," in *Proccedings of the 10th International Conference on Frontiers of Information Technology (FIT)*, 2012.

[8] VICINITY Project Website. [Online]. Available: http://vicinity2020.eu/

[9] "VICINITY D1.2. Report on business drivers and barriers of IoT interoperability and value added services," September 2016.

[10] "VICINITY Deliverable D2.1. Analysis of Standardisation Context and Recommendations for Standards Involvement," September 2016.

[11] [Online]. Available: www.onem2m.org/about-onem2m/organisation-and-structure

[12] P. C. H. Alliance, "White paper on fundamentals of data exchange," September 2015.