Future Internet: the Connected Device Interface Generic Enabler

P. GARINO[1], L. ZUCCARO[2], G. ODDI[2], A. PALO[2], A. SIMEONI[2]
[1]Telecom Italia,
Via G. Reiss Romoli 274, Torino, 10148
ITALY
pierangelo.garino@telecomitalia.it
[2] Dip. di Ingegneria Informatica, Automatica e Gestionale (DIAG),
Università degli Studi di Roma La Sapienza
Via Ariosto 25, Rome, 00185,
ITALY
{zuccaro;oddi,andi,asimeoni}@dis.uniroma1.it

Abstract: - The panacea of application development is the ability to create a single application and deploy it on a range of mobile and traditional computing platforms. Web technology has offered the possibility of common programming environment for application developers across a range of platforms, however this common environment has not been realised. Rather various web runtime platforms have emerged, Tizen, Webinos, PhoneGap, and Firefox OS, each one offering a web technology based developer environment but with different APIs. This still requires applications to be ported between web runtime environments. Several solutions to face the Internet heterogeneity exist, but none of them proposes a holistic and standardizable approach to interface the connected devices to the network. In this work we present a Future Internet enabler called Connected Device Interface (CDI) proposing a unified and extended set of APIs counting on both best available standard solutions as well as completely novel functionalities. The CDI architecture is analysed in detail, its API set is described and an instance implementation is presented.

Key-Words: - Future Internet, Generic Enabler, Fragmentation Problem, Connected Device Interfaces, Cross-Domain and Cross-Device Development, Web runtime, Application platform

1 Introduction

It is a fact that the Internet revolution began in nineties and is still leading the technology evolution, indeed most of the current strategic trends rely on it ([1], [2]). It is a tangible fact that the amount of available services and contents grows as well as the ways they can be accessed. Internet has become in few years an open business arena where device manufacturers, content, service and network providers fight to gain the market share. But the freedom of the Internet-based business has its own drawbacks. Driven by innovation and offer differentiation, device vendors and manufacturers are continuously introducing new features and capabilities in their products. Such vertical businesses introduce severe interoperability and portability limitations, both due to the lack of standards and to the existence of heterogeneous technologies (e.g., native programming, scripting, codecs, platforms, app markets). This is known as the “fragmentation problem” because it poses high barriers to the development of innovative future network applications, to the remote management of device connectivity functionalities, but it also decreases network applications ecosystem revenues and increases expenses and network and mobile energy consumption (e,g, see [3]-[6]). The need for standard solutions in the future Internet is exploding, for instance, in wireless sensor networking scenarios, where a common architecture (such as ZigBee) may allow the design of energy-efficient signal processing facilities (see, e.g., [7], [8]). The only constraint in a similar scenario, is that users are willing to access their favourite services and contents by means of devices connected to the Internet through access technologies managed by a limited set of network operators. Such consideration inspired some edge research works [9] as well as standardization bodies and industrial initiatives worldwide to redesign the Future Internet fundamentals with the aim to address the fragmentation problem.

1.1 Future Internet

The raise to the Future Internet is worldwide challenge. Several initiatives in China, Japan and USA have already started with the aim of developing new infrastructures for the Internet. In January 2010 the most representative European and international companies in the field of ICT presented a white paper defining the role of the Future Internet Public Private Partnership (FI-PPP). Although the interest of the European authorities and industry also encompasses the need of an improvement of the network infrastructure. the enhancement of the overall capacity, and the security aspects, the priorities of the European approach are different in that they embrace the full system dimension of the Future Internet. The Future Internet is articulated around the Internet by and for People, the Internet of Contents, the Internet of Services and the Internet of Things supported by the Network Infrastructure foundation. The European Future Internet Initiative ambitions include the need (i) to provide the European citizen and industry with better and smarter services and applications, (ii) of fostering the creation of a new extended economy environment over the net, accessible by stakeholders in all the EU member states (iii) of leveraging the enlargement of service offering over the net, allowing a wider range of better quality-enabled services to all economy stakeholders, from SMEs to large corporations, from academic centres to public agencies.

1.2 FI-WARE and FI-CORE projects

To cope with these challenging objectives a big research project, FI-WARE, created the foundation of the European Future Internet Core Platform. In the last four year (from 2011 to 2014), as a part of the ambitious European FI PPP program, FI-WARE has started to materialize as a powerful foundation for the Future Internet. FI-WARE is an innovative, open cloud-based infrastructure for cost-effective creation and delivery of services, at a unprecedented scale. FI-WARE is now well underway to successfully achieve its goals of boosting the effectiveness of creating new services of high economic and societal value, reinforcing EU competiveness and bringing opportunities for high-growth entrepreneurs and SME players.

The FI-Core project (started on the 1st of Septmber 2014) aims to complete the FI-PPP vision and support a truly open innovation ecosystem around a working instance of FI-WARE (called FI-Lab) that is distributed across multiple datacenters in Europe and is effectively operated using a proper the suite of Future Internet tools (called FI-Ops). This way, FI-WARE, FI-Ops and FI-Lab will be a globally competitive foundation for Europe's economy.

1.3 Generic Enablers

In FI-WARE a Generic Enabler (GE) is defined as a functional building block of FI-WARE ([9]-[11]). Any implementation of a GE is made up of a set of components which together supports a concrete set of Functions and provides a concrete set of APIs and interoperable interfaces that are in compliance with open specifications published for that GE. By now the GE can be easily found on the FI-WARE catalogue, their software can be downloaded and tested and their documentation has open access as well. The GEs can be composed and instantiated autonomously, so that each stakeholder of the Future Internet can personalize its infrastructure and take part to the game.

It is worth noticing that GEs are built on top of existing communication network technologies: the control commands generated by the GEs, in fact, are then enforced by the underlying networks by means of their specific resource management procedure (e.g., admission control [12]-[15], routing [16] [36] and [37], load balancing [17]-[20], congestion control [21], [22], scheduling [23], resource discovery [24], [25] and allocation [26] [38], quality of service [35] [39], quality of experience [27], [28] and security [29], [30]).

1.4 Connected Device Interface

In compliance with the Future Internet architecture described in [9]-[11], in the present work we address the definition of a GE implementing a common set of interfaces to the connected devices aiming to extend and influence the existing standards and initiatives (e.g., W3C, GSMA, Appcelerator) active in the field. The GE is called Connected Device Interface (CDI) and it is part of a more comprehensive Future Internet Core Platform developed within the pan European FI-WARE project. In section 2 we analyse the state of the art solutions available to overcome the fragmentation problem, highlighting their limitations that call for a novel solution. In section 3 we describe the CDI architecture from a functional point of view, together with details about the offered APIs. An instance implementation of the CDI GE is presented in section 4. A brief analysis of the potential business benefits is done in section 5. Some considerations and future works are reported in the conclusive section.

2 State of the art

It is recognised that the fragmentation problem is caused by the coexistence of dissimilar software and hardware platforms (e.g. Grid [31]) adopted for connected devices, and by the variety of incompatible operating systems and programming languages. This heterogeneity is introducing several troubles to develop once for all applications and to make them run easily on all such platforms. Moreover, the lack of standard interfaces to control and manage broad ranges of different devices poses high barriers for applications innovation and interoperability.

The early tentative to fill the gap has been the adoption of middleware based technologies (Java, Flash, Shockwave, just to mention the most known), given their ability to abstract native platforms, boosted the portability of applications. However the adoption of the very same middleware has not been standardized, and thus cannot be imposed to such a broad and fragmented market. A recent survey [32] conducted by Kendo UI among developers and IT decision makers revealed the preference of web-based technology over native SDKs. In particular, survey results showed HTML5 beating out native SDKs as the preferred choice for developers across the board, regardless of company size. Taking this principle in mind, we decided to exploit the natural convergence of native applications toward web based technologies. Indeed most of the existing platforms already support, or at least are going to support, web based environments (browsers, web runtimes) and languages (HTML-HTML5, CSS and JavaScript). This broad and implicitly agreed support for web technologies, makes it the least common denominator on top of which any convergent solution to the fragmentation problem can rely on. We strongly believe that Future Internet applications will be based on web standards. This consideration is supported by the fact that standardization bodies and industrial initiatives work to define common sets of interoperable web based APIs, giving to Future Internet applications full access to native device capabilities and enhanced communication means with the external world.

For what concerns the interaction between the application and the device capabilities the main standardization effort is carried on by the W3C, aimed to produce detailed WebIDL API specifications, to be implemented and supported by most of the web browsers and web runtimes. For what concerns the remote device configuration the main initiative is carried out by the Open Mobile Alliance Device Management working group. It specifies protocols and mechanisms to perform management of devices by the exposure of RESTful APIs.

3 CDI Architecture

The Connected Devices Interface Generic Enabler provides the means to detect and to optimally exploit capabilities and aspects about the status of connected devices, through the implementation of interfaces and Application Programming Interfaces (APIs) towards device features. The CDI aims to select the best set of APIs, extend them to fill the gap and to fuse them into a unique, homogeneous and standardizable GE representing a milestone in the field of interfacing the connected device to the network.

The term “connected devices” refers to a broad range of networked electronic components, including Handsets (such as cellular phones, smart phones), Tablets, Media phones, Smart TVs, Set-Top-Boxes, In-Vehicle Infotainment, Information kiosks, each being able to connect to a communication network.

The CDI is located in the connected device, as shown in Figure 1, with the aim to provide common APIs for exploiting device capabilities, resources, contextual information and contents to network services and developer’s applications.

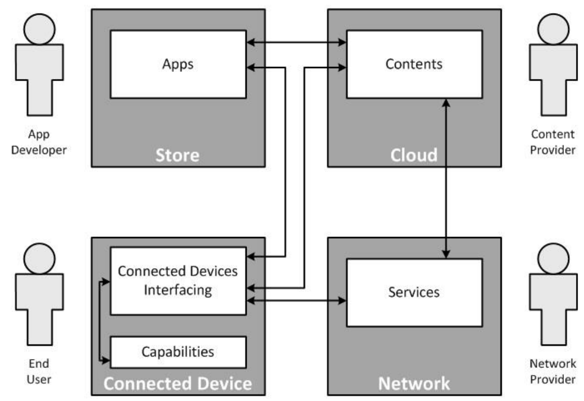

Figure 1 – CDI target usage

The APIs provided by the CDI GE consist of four main macro groups, identified by the *On-Device*, *Distributed Compute*, *Remote Management* and *Mobility Manager* interfaces.

The *On-Device* APIs offer functionalities to local running applications, and are aimed to bridge the portability gap between many different platforms, by exposing a common set of APIs to exploit native devices' capabilities. This subsystem contains also models and primitives for the assessment of the current Quality of Experience level of the user.

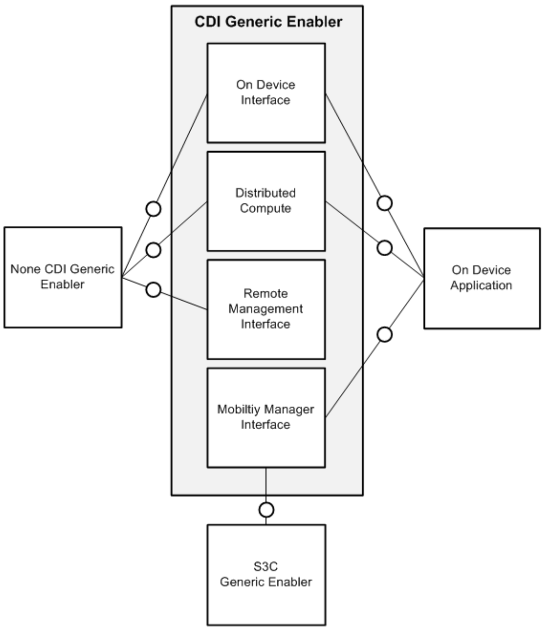

Figure 2 - High level description of the CDI's architecture in FMC (Fundamental Modeling Concepts)

The *Distributed Compute* interface provides an easy and convenient method for CDI based applications to communicate between cloud and connected device. It allows an application written for CDI to execute both on the connected device and on a cloud hosted device.

The *Remote Management* interface presents an externally accessible interface which supports other Generic Enablers. It allows to configure device and network settings providing a push notification framework where external GEs can signal connected devices with information and data. The managed information includes e.g. configuration settings, operating parameters, software and parameters, application settings, user preferences.

The *Mobility Manager* interface is aimed to connect the FI-WARE S3C (Service Capability Connectivity & Control) system, and exploits its exposed APIs toward the EPC (Evolved Packet Core). Mobility Manager interactions with S3C are aimed to configure network policies, routing rules, and flexibly allocating bandwidth on the access network, to fulfill applications' QoS requirements.

3.1 Technical requirements

Even though the set of functionalities exposed by the different blocks are quite different, all the described subsystems share the following technical requirements.

3.1.1 API discovery and linkage to application/service

Most of the identified functionalities are already provided at native level, but some of them (i.e., QoS/QoE) are not standard, and require an external native implementation. Moreover also standard functionalities could not be supported by a given device, entailing a strong necessity for an integrated API discovery service. That enables applications (local or remote) and external services to discover all APIs they depend on, for a given target platform. If APIs are successfully discovered the application/service should be able to bind and finally use the required functionalities.

3.1.2 API call delivery at native level

API invocations generated at API level must be conveyed down at native level. The same holds for API return values (i.e., JavaScript callbacks) that must be replied back from the native level to the API level.

In order to address these needs, CDI team studied a variety of existing web based platforms and selected the best one to fulfill the CDI requirements. This paper focuses on the implementation done adopting the Webinos framework as a platform to embed the aforementioned APIs inside the device. Webinos [34] is a service platform project under the European Union's FP7 ICT Programme. It provides a web runtime system offering a common set of APIs that runs over multiple devices (phone, PC, tablet, TV, vehicle etc.) and operating systems (Linux, Windows, OSX, Android).

Webinos permits an overlay network of web applications that use APIs offered by local and remote devices. It is based on the concept of Personal Zone (PZ), as a set of Webinos enabled devices owned by a Webinos user, and federated under a Personal Zone Hub (PZH) hosted in the cloud, or on top of one selected device in the PZ. Local and remote (intra/inter PZ) service invocation is supported by JSON-RPC 2.0 calls, and routed by Personal Zone Proxies (PZP) running on each device.

3.2 CDI APIs

CDI implements a subset of the W3C API specifications extending them with new and innovative functionalities. The CDI APIs allows to access many information of a connected device, to read data from its sensors, to access media and the personal information in the device, to detect and use voice or messaging communication functionalities, and to distribute code execution between multiple clients and servers. Moreover, the network operator can exploit the CDI APIs to remotely manage the parameters used by the device for establishing the connection to the network, controlling the Quality of Service, and measuring a user feedback to improve the perceived Quality of Experience. The result is that end users can enjoy their favorite contents anywhere and anytime, with the best experience, using their preferred apps across heterogeneous networks and different connected devices. Application developers, at the same time, may adopt the CDI APIs to develop applications and services that operate efficiently and consistently over dissimilar connected devices and network accesses.

The APIs, organized into the groups detailed above, are further subdivided into the following functional blocks:

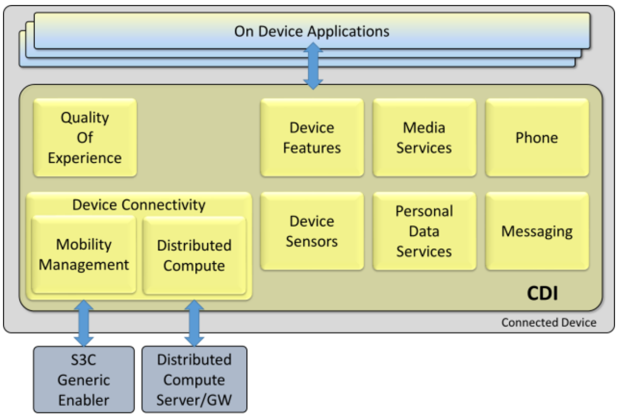

Figure 3 – CDI functional structure in FMC (Fundamental Modeling Concepts)

- *Quality of Experience (QoE)*: CDI provides to application developers useful QoE control functionalities. The QoE control functionalities are embedded are able to combine explicit QoE feedback coming from the user, with the QoS level provided by the network [27]. Both inputs are matched against a target QoE level to be achieved, and network resources are consequently requested to the Mobility Manager in order to reach the target level.
- *Device Sensors*: All connected devices targeted in FI-WARE contain a range of sensors which can provide useful functionality for application developers. CDI supports the following four sensor types: Camera, Microphone, Geo-Location, Device Orientation & Accelerometer.
- *Device Feature*: This API group enables on device applications and remote services to tailor their behaviour in function of the specific device features (device form factor, screen size, CPU, disk space, current connectivity etc.) to be accessed by local and remote applications.
- *Media Services*: Set of APIs to discover media types and codecs supported by the device, for media consumption and production.
- *Phone*: Allows to discover if the device is a phone, and in such case to access basic phone information (Engaged, Ringing, Dialing, Calling or Idle)and operations (make call, end call, answer call, reject call).
- *Personal Data Services*: Aimed at providing functionality to access (Read/Write) personal data on the device, opening up a wide range of possible applications. Personal data includes: Contacts, Calendar, Gallery, File System access and Personal Data Service Discovery (to verify native functionality support).
- *Messaging*: More and more devices are equipped with communication technology. It makes sense to offer that functionality to application developers where possible. This group of functionality deals with Email, SMS and MMS messaging.
- *Device Connectivity*: This functional group enables developers to access device connectivity capabilities, detect connectivity technologies (WiFi, 3G, Bluetooth, NFC), switch connection between available communication interfaces, release connections, access connectivity features,

enumerate access networks, connect to a network. This group also includes the QoS support provided by the Mobility Manager subsystem and the Distributed Compute functionalities.

4 CDI Implementation

The CDI development team is devoting a big effort in implementing W3C specifications, and investigating, validating and proposing the standardization of new APIs, that are expected to be valid enablers for proliferation of Future Internet services and applications. CDI's roadmap spans three years of activity, with three major delivery milestones as planned by FI-WARE. The first release was defined with Webinos as the underlying integration platform, as it provides a bidirectional channel between JavaScript and device's low level core. The CDI Javascript interface is broken down into a number of functional blocks, each one consisting of access to a particular native interface. This implementation was initially designed to support Android platform and is available to users and developers in the FI-WARE Catalogue [33] under the item "A-CDI".

All these functionalities are supported by the subsystem in Figure 4, hosted on top of the connected device, and perfectly integrated within the Webinos on device framework.

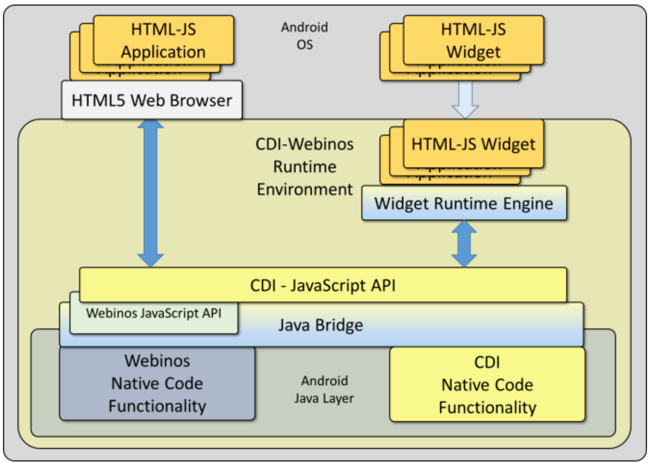

Figure 4 - High level description of the CDI's architecture in FMC (Fundamental Modeling Concepts)

4.1 Webinos integration

On device web applications hosted in the browser or in the Webinos WebRT, are able to discover APIs registered by a RPC (Remote Procedure Call) handler. For the sake of brevity, we will refer to WebRT as to any environment able to host and execute applications built on top of web languages. Available APIs are registered and discoverable through an URL that uniquely identifies the group of provided functionalities (i.e., http://fiware.cdi.dev-features.org/). Accordingly to the Webinos approach, for developers' use such URLs are all listed in a publicly available repository on the web. Once APIs have been discovered, Webinos offers functions to bind them to applications and finally invoke the provided methods. API Discovery/Bind/Usage invocations from applications are all packed into JSON RPC 2.0 messages, and sent by the JavaScript CDI API to a RPC handler waiting at the other end of a communication pipe. This is the first step to convey API calls out of the WebRT. The communication pipe is also engaged by JavaScript callbacks, when they are sent back to applications in response to API invocations. The whole system on the other side of the pipe is hosted by NodeJS, a highly modular JavaScript based environment, that allows multithreading, and enables access to native device libs. NodeJS is thought to provide server side functionalities and is the core platform for Webinos operations. The RPC handler is hosted as a NodeJS module, aimed to deliver RPCs to a backend CDI layer (again composed of NodeJS modules), whose purpose is to relay API calls at native level, using primitive NodeJS glue mechanisms. In the case of native Java code to be reached (i.e., in Android platforms) NodeJS does not support the mapping, but Webinos comes in help by providing a Java-bridge facility. At the lowest level in the picture we have the native platform along with all described groups of functionalities, that in some cases are not implemented by the OS and need to be separately ported into the system (i.e., QoE and QoS/Mobility Manager functionalities). For what concerns the Mobility Manager interface toward the S3C, this is implemented as a RESTful web service, and is completely independent by Webinos and its framework. CDI exposes RESTful APIs also to support machine-to-machine and to handle agent-like communications.

4.2 Business benefits

The CDI represents a valid solution to the fragmentation problem. Thus it is a breakthrough milestone in the Future Internet convergence path. The main effect of such convergence is measurable in the context of interoperability among the different actors of the Internet business: users, device manufacturers, network operators, content providers, application and service developers. The business impact is obvious for all the above mentioned stakeholders. Users of such devices can enjoy any available content or service through the accessible networks. Device manufacturers can use standard APIs to open their platforms to a variety of contents and services guaranteeing a continuity in the user experience and expectations. The portability of contents and services enables the "create once, run anywhere" paradigm that allows developers of applications and services as well as producers of contents to reach larger user communities. Another important aspect for the CDI acceptance is an explicit support to remote management and control of device configuration, allowing the network operator to play a key role in the cooperation between users, devices, services and contents with the Internet infrastructure. In this context, a hypothetical scenario may involve a cloud-hosted media rich service accessible by many consumers across fixed or mobile terminal devices. The service provider may want to ensure that all consumers get the best possible experience, as a function of their connectivity, of the display features and of the device processing power, or even considering remaining battery power of a mobile equipment. So, by programmatically enabling the service with the ability to detect relevant details of the client device and its connectivity, decisions may be made at run-time in terms of selecting different levels of media richness, including sizing, resolution, quality and format, such as 2D or 3D, for individual users.

5 Conclusions and Future Works

The work described aims to be a solution to the problem of platforms fragmentation that affects application portability, interoperability among heterogeneous systems and impacts heavily on remote devices configuration. The adoption of the CDI concept dramatically pushes connected devices to become flexible, ready to use, multi-purpose and vendor independent open boxes. It is worth to note that any CDI implementation would necessarily rely on existing technologies and thus needs to be updated and maintained as new features and functionalities become available. But we believe that the CDI is the only practical solution to feed the related ecosystems with a breeze of innovation, breaking barriers for new services, applications and then revenues.

The CDI team has reached the 3rd final release of the FI-WARE project by integrating an important set of the identified functionalities on top of Webinos. The architecture design has been improved to support extended use cases, and interfaces with other GEs. Such interactions and dependencies are planned to be supported and addressed in the FI-Core project together with enhanced On Device, Remote Management and Mobility Manager functionalities.

Acknowledgements

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 285248 also referred as FI-WARE. It is worth to mention the valuable contribution of Chris Woods (Intel), Gianmario Bollano (Telecom Italia), Frank Schulze (Fraunhofer FOKUS), Prof. Francesco Delli Priscoli, Prof. Stefano Battilotti, Prof. Salvatore Monaco, Prof. Roberto Cusani, Dr. Antonio Pietrabissa and Dr. Vincenzo Suraci ("La Sapienza" University of Rome).

References:

[1] C. Pettey, "Top 10 Strategic Technology Trends for 2013", Gartner Symposium/ITxpo, Orlando, Fla., October 23, 2012

[2] J.P. Javaudin, M. Bellec, D. Varoutas, V. Suraci (2008). Towards convergent Gigabit Home Networks. In: *IEEE PIMRC 2008 - Gigabit Home Access Special Session*. Cannes, France, Sept 15-18, p. 1-5, ISBN: 978-1-4244-2643-0

[3] M. Migliardi, A. Merlo. "Improving energy efficiency in distributed intrusion detection systems" (2013) *Journal of High Speed Networks*, 19 (3), pp. 251-264.

[4] M. Curti, "Towards energy-aware intrusion detection systems on mobile devices" in *Proc. of the 2013 International Conference on High Performance Computing and Simulation, HPCS 2013*, art. no. 6641428, pp. 289-296.

[5] V. Suraci, A. Marucci, R. Bedini, L. Zuccaro, A. Palo (2013). Energy-aware control of home networks. In: *Lecture Notes in Computer Science*. vol. 7972, p. 299-311, 2013, doi: 10.1007/978-3-642-39643-4_23

[6] V. Suraci, A. Palo (2011). Green networking: Integration of Wireless Sensor Network with Home Gigabit Access. In: *Proceedings of Future Network & Mobile Summit 2011*. Warsaw, 15-17 June 2011, ISBN: 978-1-905824-16-8

[7] M. Martalò, C. Buratti, G. Ferrari, and R. Verdone, "Decentralized detection in IEEE

802.15.4 wireless sensor networks," *EURASIP J. Wirel. Commun. Netw,* January 2010, 10 pages. DOI=10.1155/2010/174063.

[8] M. Martalò, C. Buratti, G. Ferrari, and R. Verdone, "Clustered IEEE 802.15.4 Sensor Networks with Data Aggregation: Energy Consumption and Probability of Error," *IEEE Wireless Communications Letters,* vol.2, no.1, pp.70-73, February 2013. doi: 10.1109/WCL.2012.112012.120644.

[9] M. Castrucci, M. Cecchi, F. Delli Priscoli, L. Fogliati, P. Garino, V. Suraci (2011). Key Concepts for the Future Internet Architecture. *Future Network and MobileSummit 2011 Conference Proceedings*. Warsaw, Poland, 15 - 17 June 2011, ISBN: 9781905824236

[10] M. Castrucci, F. Delli Priscoli, A. Pietrabissa, V. Suraci (2011). A Cognitive Future Internet Architecture. In: *The Future Internet*. p. 91-102, SPRINGER, ISBN: 9783642208973, doi: 10.1007/978-3-642-20898-0_7

[11] A. Palo, L. Zuccaro, A. Simeoni, V. Suraci, L. Musto, P. Garino. A common open interface to programmatically control and supervise open networks in the future internet. In: *2013 Future Network and Mobile Summit*, 2013. 6633572, ISBN: 978-190582437-3, Lisbon, 3 July 2013 through 5 July 2013

[12] A. Pietrabissa, "An Alternative LP Formulation of the Admission Control Problem in Multi-Class Networks", *IEEE Transaction on Automatic Control*, Vol. 53, N. 3, pp. 839-845, April 2008, DOI: 10.1109/TAC.2008.919516.

[13] A. Pietrabissa, "Admission Control in UMTS Networks based on Approximate Dynamic Programming", *European Journal of Control*, Vol. 14, N. 1, pp. 62-75 , January 2008, DOI:10.3166/ejc.14.62-75.

[14] S. Mignanti, A. Di Giorgio, V. Suraci (2008). A Model Based RL Admission Control Algorithm for Next Generation Networks. In: *Proceedings - The 2nd International Conference on Next Generation Mobile Applications, Services, and Technologies, NGMAST 2008.* Cardiff, WALES, SEP 16-19, 2008, ISBN: 978-0-7695-3333-9, doi: 10.1109/NGMAST.2008.19

[15] V. Suraci, G. Oddi, N. Mattiacci, A. Angelucci, (2010). Admission Control and Drop Strategies in a UPnP-QoS Controlled Home Network. In: *IEEE PIMRC 2010 Services, Applications, and Business - 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communication*. Instanbul, Turkey, September 26-30, 2010, ISBN: 978-1-4244-8017-3.

[16] C. Bruni, F. Delli Priscoli, G. Koch, A. Pietrabissa, L. Pimpinella, "Network decomposition and multi-path routing optimal control", *Transactions on Emerging Telecommunications Technologies*, Vol. 24, Issue: 2, March 2013, pp. 154-165, doi: 10.1002/ett.2536.

[17] G. Oddi A. Pietrabissa, F. Delli Priscoli, V. Suraci, "A decentralized load balancing algorithm for heterogeneous wireless access networks", *World Telecommunication Congress (WTC)*, Berlin, June 2014.

[18] D. Macone, G. Oddi, A. Palo, V. Suraci (2013). A dynamic load balancing algorithm for Quality of Service and mobility management in next generation home networks. *Telecommunication Systems*, vol. 53, p. 265-283, ISSN: 1018-4864, doi: 10.1007/s11235-013-9697-y

[19] T. Meyer, V. Suraci, P. Langendörfer, S. Nowak, M. Bahr, R. Jennen (2009). An inter-MAC architecture for heterogeneous Gigabit home networks. In: *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, Tokyo; Japan, 13 September 2009 through 16 September 2009; , doi: 10.1109/PIMRC.2009.5449931

[20] M. Castrucci, G. Oddi, G. Tamea, V. Suraci (2010). Application QoS Management and Session Control in a Heterogeneous Home Network using Inter-MAC layer support. In: *Proceedings of Future Network & MobileSummit 2010*. Florence Italy, June 16-18, 2010, ISBN: 978-1-905824-16-8

[21] A. Pietrabissa, F. Delli Priscoli, A. Fiaschetti, F. Di Paolo, "A Robust Congestion Control for Communication Networks with Time-Varying Delays", in Proc. of IEEE CSS International Conference on Control Applications 2006, pp. 2093-2098, Munich, Germany, 4-6 October 2006, DOI: 10.1109/CACSD-CCA-ISIC.2006.4776963.

[22] F. Delli Priscoli, A. Isidori A. (2005). A control-engineering approach to integrated congestion control and scheduling in wireless local area networks. *Control Engineering Practice*, *13*(5), 541-558.

[23] R. Cusani, F. Delli Priscoli, G. Ferrari, M. Torregiani (2002). A novel MAC and Scheduling strategy to guarantee QoS for the new-generation WIND-FLEX wireless LAN. *IEEE wireless communications*, *9*(3), 46.

[24] V. Suraci, S. Mignanti, A. Aiuto (2007). Context-aware Semantic Service Discovery. In: *Proocedings of Mobile and Wireless Communications Summit*. Budapest, July 2007, p. 1-5, ISBN: 963-8111-66-6, doi: 10.1109/ISTMWC.2007.4299110

[25] K. Frank, V. Suraci, J. Mitic (2007). Personalizable Service Discovery in Pervasive Systems. In: *Proceedings of Fourth International Conference on Networking and Services ICNS 2008*. Gosier, Guadeloupe/France, 16-21 March 2008, p. 182-187, doi: 10.1109/ICNS.2008.21

[26] G. Oddi, M. Panfili, A. Pietrabissa, L. Zuccaro, V. Suraci (2013). A resource allocation algorithm of multi-cloud resources based on Markov Decision Process. In: *IEEE 5th International Conference on Cloud Computing Technology and Science (CloudCom 2013)*, Bristol, UK, 2-5 December, 2013, pp. 130-135, doi: 10.1109/CloudCom.2013.24.

[27] F. Delli Priscoli, A. Pietrabissa, V. Suraci, M. Iannone (2012). Modelling Quality of Experience in Future Internet Networks. In: *Proceedings of Future Network & Mobile Summit*. p. 1-9, ISBN: 9781467303200, Berlin, 4 - 6 July 2012

[28] F. Delli Priscoli, L. Fogliati, A. Palo, A. Pietrabissa, "Dynamic Class of Service Mapping for Quality of Experience Control in Future Networks", *World Telecommunication Congress (WTC)*, Berlin, June 2014.

[29] F. Delli Priscoli, A. Fiaschetti, V. Suraci (2012). The SHIELD Framework: how to control Security, Privacy and Dependability in Complex Systems. In: *IEEE Workshop on Complexity in Engineering*. Aachen, June 11-13, 2012, doi: 10.1109/CompEng.2012.6242962

[30] A. Fiaschetti, F. Lavorato, V. Suraci, A. Palo, A. Taglialatela, A. Morgagni, A. Baldelli, F. Flammini (2011). On the use of semantic technologies to model and control Security, Privacy and Dependability in complex systems. In: *Proceedings Of 30th International Conference on. Computer Safety, Reliability and Security*. Naples, 19th to the 21st of September 2011, ISBN: 978-3-642-24269-4

[31] A. Merlo, "Secure cooperative access control on grid", (2013) Future Generation Computer Systems, 29 (2), pp. 497-508

[32] Global Developer Survey 2013. Available at: http://www.kendoui.com/docs/default-source/documents/global_developer_survey_kendo_ui_2013.aspx

[33] A-CDI Catalogue Entry. Available at: http://catalogue.fi-ware.org/enablers/cdi

[34] Webinos Introductory Whitepaper. Available at http://webinos.org/files/2014/06/webinos-foundation-whitepaper.pdf

[35] R. Pecori, L. Veltri, "A Statistical Blind Technique for Packet-level Recognition of IP Traffic with Dependence Enforcement", 10th IWCMC, 5th International Workshop on TRaffic Analysis and Characterization, TRAC 2014, Nicosia, Cyprus 4th-8th August, 2014

[36] R. Pecori, L.Veltri, " Trust-based Routing for Kademlia in a Sybil Scenario", XXII IEEE SoftCOM, International Conference on Software Telecommunications and Computer Networks, September 17th - 19th, 2014, Split, Croatia.

[37] Y. Honma, M. Aida, H. Shimonishi "A New Routing Methodology Focusing on the Hierarchical Structure of Control Time Scale" WSEAS Transactions on Communications, ISSN / E-ISSN: 1109-2742 / 2224-2864, Volume 13, 2014, Art. #56, pp. 505-512.

[38] X. Huang, S. Leng, F. Wu, Y. Mao h, "Energy-Efficient and QoS-Ware Resource Allocation for Relay-Enhanced OFDMA Wireless Cellular Networks", WSEAS Transactions on Communications, ISSN / E-ISSN: 1109-2742 / 2224-2864, Volume 13, 2014, Art. #44, pp. 394-401.

[39] H. Luo, D. I. Laurenson, A. Guo, J. Chen, P. Yang, H. Jiang, "A QoS Balancing Model for Mobile Ad hoc Networks", WSEAS Transactions on Communications, ISSN / E-ISSN: 1109-2742 / 2224-2864, Volume 13, 2014, Art. #41, pp. 372-377.