New Challenges on Crossplatform Digital Contents

Jesús Iglesias Feijoo, Guillermo Amat Gomariz

Zed World Wide

***Abstract* — When we speak about devices and platforms, generally we think about those of general use which are currently available (mainly smartphones and tablets). Surely, we would forget all those which are on the way (watches, glasses, cars) and those which are coming. The Internet of Things will transform the technological world in which we are into an amalgamation of devices and interfaces. This paper analyses the challenge for the coming years of getting all these new devices to communicate between them, regardless of their technology and the platforms they use, and it is based on the works done under the Visio Project, funded by the Spanish Ministry of Industry, Energy and Tourism. Finally, a truly universal platform to avoid market fragmentation and provide access to information and services is proposed.**

***Keywords* — Apps, cloud, crossplatform, internet of things**

I. Introduction

OVER the past 20 years we have been concerned about the compatibility of classic multimedia content (images, music, and video). It has never been as easy as it is now to play any media format on most popular devices (computers, smartphones, tablets...). This fact has been reached due to a slow process of technical improvements, format standardization processes, and also the adoption of these advanced and standardized formats by manufacturers and developers. This has led to a multi-device and multi-platform media scenario. The achievement has been to ensure the user is able to play media on any device, without needing to have the same content in multiple formats to display on different devices, and thus, avoiding the case of having to pay for every required format.

As an example of the opposite, in the late 90s, although HTML was a widely adopted standard, the emergence of various web browsers led to a real browser war where the most affected were web page developers, as they had to create an almost tailored version of their work for each of these browsers. Even worse, the end users themselves were affected as they lost some freedom regarding what browser to use, because some of them were unable to view certain web pages with certain browsers.

One more case to mention is the appearance of the Apple Store in July 2008 [1]. The iPhone application market revolutionized the market of content for mobile devices. It was the beginning of a revolution but also the beginning of the same old mistake. The rise of this concept and the spread of smartphones have achieved something particularly interesting, since nobody but the developers seem to be concerned about the total lack of cross-platform support for these new multimedia contents: applications.

This document is structured as follows: Section II defines the current problem of content incompatibility, section III explains the implications of security issues in the current scenario while section IV introduces the relation with cloud computing. In section V some existent partial solutions and approaches are explained. Finally, section VI proposes a number of possible solutions to achieve a cross platform of digital content environment.

II. Current problem

The next section shows that regardless of the type of Smart Object, the problem is common for all of them. It is observed that the strategy designed for smartphones has propagated the same problem to the new technological sectors, as they have all copied the same paradigm of proprietary, closed and completely isolated systems.

A. Smartphones

As these new "application markets” evolve, the various players in terms of operating systems are making their application platforms available for developers, thus consolidating two major companies (Apple and Google) over the rest of competitors:

1. iOS
2. Android
3. Windows Phone
4. Blackberry
5. Others

From the user’s point of view, this behavior has created a barrier against freedom of choice of the user device. This occurs, for example, when users buy applications for their smartphones. If, after a few months, they decide to switch to another device with a competing platform, e.g. iOS to Android, then what happens is that previously purchased

DOI: 10.9781/ijimai.2015.323

applications must be acquired again. When a user has to pay again for all his applications, the decision to change one platform for another is a deterrent like no other, as the cost after several years of use, paying for several applications, can be very high and unaffordable.

As a consequence, currently we do not buy applications but the right to use one in certain platform. A similar fact occurred years ago with the attempt to encode media formats using DRM technologies [2], trying to restrict music and videos to certain players. This, completely absurd from the point of view of the user, is however a big deal for the owners of the application stores.

Therefore, a problem for the development of multiplatform content arises. This is what usually is mentioned to explain the lack of such content. But on the other hand, this deficiency makes it difficult to find useful multiplatform content to end users for their consumption. However, the latter, as explained above, is not a serious problem for big players, as it is yielding great benefits for them.

B. Smart TVs

As for Smart TVs, they are undergoing a similar process. Many manufacturers started to launch their new devices based on Android OS but now major TV brands are introducing their own operative systems, aiming at differentiating them from the rest of their competitors:

- Samsung has recently announced Tizen OS for Smart TVs [3].
- Panasonic is going to be using Firefox OS [4]
- LG is already using Webos [5].

Currently, Sony is the only one which is still faithful to Android. The rest of manufacturers have realized that, by doing that, they are increasing Google's profits.

The main conclusion is that we are witnessing a big fragmentation of operating systems for future TVs. This is not necessarily a negative thing; on the contrary, it increases competition and thus, the potential benefits for users. However, this fragmentation means, again, a lack of multiplatform content to allow a total availability of applications. In other words, each manufacturer will have to worry about increasing their applications catalogue on their own.

Smart TV content developers are in the same situation that occurs for the kind of devices that were previously discussed. Anyone who wants to make an application for Smart TVs will be forced to repeat the development as many times as platforms they wish to reach.

C. Connected Cars

Connected cars are one of the markets that most excitement and growth will experiment in the coming years. Traditionally car interfaces have been completely created by each car manufacturer. There have not been major horizontal suppliers of user interfaces. As the vehicles were implementing dashboard displays and infotainment systems, each manufacturer was developing their own UI to suit their needs. Therefore, there is no relationship between one car manufacturer and the rest. Now it is time for these interfaces to evolve providing connectivity, allowing the use of applications and so forth. The conclusion is that this sector has arrived to the same point we explained before: each manufacturer has their own proprietary and closed system and there is no way to get some multiplatform content.

In recent times, this situation has become increasingly complex due to new products being launched by two of the main mobile communication companies: Apple for Car Play [6] and Google for Android Auto [7]. MirrorLink should also be added to this list, as it is the solution proposed by most automotive-related manufacturers subscribed to the Car Connectivity Consortium, except for Apple. Regarding its functioning, these three systems show certain similarities, as they all project the Smartphone screen onto the infotainment system's and allow users to use certain applications, although the full catalogue of installed applications in such devices would not be accessible. All these applications would be adapted to be used in vehicles in accordance with the design, safety and usability standards of the car manufacturing sector.

Therefore, it must be stressed that these systems are not native to the actual vehicles. In fact, despite the fact that only a few commercial proposals including them have been pitched, the vehicles that have them will also count on the classic native system as devised by the manufacturer and also on a button that will grant the user access to these solutions. However, in principle, unless manufacturers design their own customization layers for these systems, users will not be able to control aspects such as the vehicle's air conditioning system or the radio, or to access the vehicle's setup panel. To sum up, users will always have two different systems in their vehicle, both with different interfaces.

It is widely thought that these systems may be the ultimate and universal solution concerning the use of applications in vehicles, but this might still be an unrealistic idea if we take into account what these products can actually achieve and the international safety standards to which they are subjected. Besides, it would also be necessary to bear in mind what the most common consequences to the use of these proprietary solutions might be. What would happen if a user decided to purchase a new telephone which they would not be able to use in their vehicle? Or if different members of the same family used the same vehicle but all of them had devices operating on different platforms? If, in a best case scenario, it were possible to install several systems in the same vehicle, what would the total cost be? The whole picture is becoming considerably more complex than it might have seemed in the first place.

D. Smart Objects

The Internet of Things and its Smart Objects, small devices that carry out specific functions within a wider network, are the ultimate development in the technological world. In order to understand their usability and future expansion, we can list a number of examples. For instance, there could be Small Objects that are able to measure the temperature or the relative humidity index in a certain environment, but there could also be other more elaborate ones that can detect a presence or, to a higher degree of sophistication, help control or prevent diseases. All these devices are connected within wider networks which receive the information gathered by the Small Objects and allow us not only to monitor their activity, but also to carry out actions on the devices so that they respond in real time to the measured conditions.

According to a study by Cisco[9], in 2015 there will be twenty billion connected devices, and this figure could be doubled by 2020. Therefore, Smart Objects are one of the subjects of interest within our industry, since they will allow nearly any everyday device to connect to the Internet and to be incorporated into the communication structures in our home or in our business. These objects would be useful both in an industrial or corporate environment and in a domestic one. In fact, DIY (do it yourself) objects will become one of the main factors contributing to the expansion of IoT networks, along with a decrease in the price of electronic components and the standardization of 3D printers.

Even though Smart Objects could still be considered a brand new element in the grand scheme of things, there is a series of mistakes that are already being made. There are not any standards in place regarding communication and interconnection protocols. There are not any free platforms that allow object networks to be developed universally in a controlled and safe environment. The market is currently monopolized by private solutions that do not interact among themselves, without the existence of open protocols or APIs that make it possible to develop a more sophisticated product.

Let us take the example of a simple “smart bulb” to help us illustrate this situation. The supplier would provide us with an application for our Smartphone from which we would be able to set up a timer for the bulb to go on and off. Thus, when we got home our Smartphone would detect the presence of our lightbulb and, depending on the time, the application would establish whether the bulb needs to be on or off. This simple case is perfectly valid, although it would be far more convenient if we could have a bulb that went on or off according to a much more realistic factor such as the degree of luminosity rather than the time of day. In this case, we should somehow communicate with the bulb and with a sensor. Complications could arise, such as the lack of a common communication protocol between the bulb and any other device, like the sensor. To sum up, nowadays it is difficult to develop Smart Objects networks by different manufacturers and which have a common platform that allows communication amongst them all, carrying out data analyses and making decisions.

We have, once again, come to the same conclusion: the new sector is making the same mistakes when they decide to hinder standardization and the use of open protocols.

III. Security and Privacy

Another important subject that seems to be overlooked for the abovementioned parties is that of security and privacy. Manufacturers will launch devices with the sole objective of making our lives easier but, at the same time, these devices gather highly sensitive information concerning users and their habits, which could potentially be dangerous if a third party accessed them with malicious purposes.

The exploitation of vulnerabilities also affects new devices. Let us give as an example the registered case of hacked televisions [10], by which a hacker would gather information by means of the built-in webcam that these televisions incorporate. This will soon extend to connected cars, in which everything is controlled by means of sensors and servos, resulting in hackers being able to access the braking or the accelerating system, among others [11]. Recently, a BMW executive warned us about the interest that many companies are showing lately on the enormous amount of data issued by modern vehicles [12].

A good example of such behavior on the part of manufacturers is that of traditional routers connecting households and small and medium-sized businesses to broadband networks. A decrease in their price resulted in less investment on the development of their firmware, and therefore on their security and privacy mechanisms, which resulted in our current scenario, with millions of routers sold in the last few years presenting great vulnerabilities, since sales were more important than security [13].

To sum up, given the insufficient quality of security nowadays, it is a matter of time before Smart Objects get hacked. This situation is due to manufacturers having ignored any engineering processes concerning the design of security for their devices. This result in a reduction of the costs incurred in during the development of the product and enables a higher number of products to be launched in a much shorter period of time.

IV. Cloud Computing

The Cloud is an essential component of any application and, according to analysts, the Cloud services market will grow exponentially in the next few years, along with the Internet of Things industry. However, the use of the Cloud has been limited to simply acting as a data storage application, granting said information an “apparent ubiquity”, whereas in actual terms it does not provide any advantages for the end user. The Cloud, along with other broadband data lines, smartphones and tablets, has changed the way we create and use content. Along with its flexibility and pay-per-use model, one of the greatest advantages of the Cloud is its actual ubiquitous access to applications and services on the Cloud. However, nowadays this ubiquity is very much limited to its ability to access applications and services adapted to multiple devices, as these solutions can be:

- Expensive, for they require complex and specific developments for each platform

- Incomplete, because, although there is some degree of adaptation of the graphic interface to each device, it is not a dynamic adaptation and, more often than not, a change of device will imply a change in the model of interaction. The majority of services in the Cloud are useful in terms of their server functionality without user interaction, as most of them focus on infrastructure.

If we looked back a decade ago, we would be able to clearly identify the major software developers worldwide. However, nowadays young people's skills at monetizing their innovation and creativity go mostly unnoticed. With technologies like the Cloud, html5 and other capacities, the application economy has changed dramatically. Amateur programmers can now make use of their creativity and innovative spirit to develop new applications or services and offer them through app stores. Unlike in past decades, today it is not necessary to count on a major capital investment in order to start writing code and start selling new applications or services. Costs have decreased drastically as a result of cloud computing and Infrastructure as Service (IasS). The Cloud has changed the way modern-day software is created.

This ubiquity becomes apparent because, in effect, when sharing data among several applications in different devices and from any location, it becomes clear that we have universal access to that information. The context of the knowledge of applications resides on those data in the Cloud and not on the actual application that has been physically installed on a device. But this "universal" availability is a first step that gets taken for granted by any user, it is the bare minimum, but does not go any further than that. The Cloud at a user level is clearly underused, since it does not allow users to go one step beyond, it cannot be used somehow to fill the niche of applications that are independent of platforms or devices. However, we are sure that there would be many more possibilities of use. Later in this article we will present a number of proposals that may contribute to the abovementioned ubiquity not to refer to data only, but also to the actual user interfaces.

V. Crossplatform development

The current model for application development is not resalable, for a developer will have to create as many versions of their application as platforms exist in the market. This not only means smartphones, as we have seen, but any other type of electronic device that is connected.

The need to develop different versions of applications and to adapt them to different devices entails a high cost. Adaptability is essential in order to create a sustainable application ecosystem.

Currently, the different operative systems are at war, which will result in the need to create more versions of the same application if the ultimate goal is to make said applications available to each and every user. This, however, would not be viable in the current scenario. Fragmentation thus becomes unsustainable.

In such a scenario, developers end up having to sacrifice a number of versions for the sake of the two or three versions of their application that would reach the highest number of potential users, thus maximizing monetization. The remaining options become automatically discarded.

The user interface management system that traditional operative systems use (Windows, Mac, Android, etc.) gives programmers more freedom through the use of its APIs, so much so that it enables them to communicate with the user by setting up interactive objects such as buttons or menus with windows and dialogues that are transferred to the user in order to show them the new information. Once the user has filled in the information required by the application, the window or user process dialogue moves on to validate the received information.

This paradigm for the development of user interfaces is not sustainable in a complete cross-platform environment, since the interaction device followed by the user at any given time does not guarantee the existence of windows or buttons (let us imagine, for instance, that the device were only able to recognize and utter speech).

We cannot state, however, that there are not any solutions in the market that advocate for a cross-platform development environment which would allow the coveted approach of "develop once, deploy many times". These, though, are still limited to specific platforms and services. For instance, there are a number of development environments that enable us to automatically export an application into iOS, Android or Windows Phone, to name a few. There are some systems that may include even more versions. However, there are not any universal solutions that are not focused on smartphones only and that apply to all kinds of devices, as we have mentioned: TVs, cars or any kinds of Smart Objects.

If we look beyond the actual devices, we can find even more problems when it comes to dealing with different interaction systems. An application that has been designed to be used in smartphones will not be easily adapted to a TV and the user experience would not be suitable, either. Moreover, if we wanted to use the same application in a small Smart Object, we would be faced with more frustration and problems related to subjects ranging from the actual human-machine interaction (haptic, voice, gestural) to the visualization of results of said interaction. It cannot be expected to use an application with a user interface if we do not have a screen to see it on.

VI. Solutions

If we take into account both conceptual and lexical design when proposing universal access to the knowledge base of an application, we can state that, even when accurately planned, said universality cannot be guaranteed, as there will be situations where the interface has not been designed for its general use. This is why it will be necessary, in most situations, to develop several versions of said interfaces which can be adapted to the needs of both devices and human groups.

In order to avoid this, said conceptual and lexical levels have to be dynamically generated according to the contextual and interaction requirements of that moment, creating a user interface management system that is intelligent and able to design and construct interactive dialogues in real time that can be especially adapted to the cognitive, perceptive and motor characteristics of an active user, as well as to the technical characteristics of the device that the user is using at that very moment, granting a great variety of users and devices full access to the interface, including those users who suffer from certain disabilities. Thus, the functionality of the application would be completely independent from the interface, which would hugely increase the system's efficiency in terms of usability and user experience. The computing capacity of the Cloud would be a solution to potential load issues in small devices, whose only concern would be the user interface.

As it is impossible to know about the device's interaction characteristics when designing and compiling, a change of paradigm becomes necessary, where the programmer would determine their needs in terms of UI (types of data required to launch a process, in/out parameters, response, etc.) and an independent service would decide in real time how to communicate with the user and their interaction device in order to obtain the information required. We can now introduce a new concept: the concept of Cloud UI, user interfaces in the Cloud.

Nowadays, we are able to maximize the Cloud's calculus power and the ubiquity that it provides us with in order to use them for the greater benefit of users and applications alike. The goal should be that applications become an essential actor in the user interface, rather than just a final solution regarding data treatment and visualization. This idea is not new, however, and there exist solutions and R+D projects based on these ideas.

Active Video[14], with its platform CloudTV, proposes a solution of this kind for TVs and set top boxes, and it is based on transforming the user interface into a video stream. This would enable the user to visualize the user interface from their device and interact with it as if it were a native application. The difference lies in the fact that the data issued throughout the interaction are sent to the Cloud, processed and sent back to the user as a response in the shape of a new video stream.

With the new interface model in the Cloud, we achieve the coveted paradigm of cross-platform applications. Regardless of the device's operative system, brand or model, we only need a small connector that would enable us to automatically use all the applications that are designed to work on such a platform. It would not be necessary to replicate and adapt applications to different systems; it would only have to be done once.

Another element to discuss would be the fact that user interfaces, instead of being tailored to needs, should become description systems. The VISIO project, developed jointly by the University of Oviedo and Zed Worldwide, is based on this concept. Communication with the user is based on a set of minimum requirements in terms of the application's functionality, which are as follows:

- The information the user must have
- How they must have it
- How it should be communicated to the system

Bearing these in mind, the actual user interface gets relegated to the background and what becomes important is the fact that any person would be able to use a specific application regardless of the device or the kind of visualization and interaction interface they are using. Through the use of mechanisms that describe an application's interface and its behavior towards the user, it would be plausible for the same application to be used in completely different devices:

- Smartphones, Smart TVs, connected vehicles and even small Smart Objects,
- Devices with a screen with, for example, small LED displays or even devices without a screen where an interface reads and describes the application or emits sound effects when the task has been processed successfully.
- Devices with various interaction systems: haptic, gestural, voice, etc.

To automatically generate the applications' interfaces in real time would enable us not only to make a general use of them, as we have just seen, but also it would improve the quality of the user experience by gradually adapting the adaptations and/or alterations of the interface to the user context so that they do not occur suddenly. For instance, through the use of logic engines, a rule defines the size of a button as BIG if the degree of driving precision is LOW and the degree of visual precision is HIGH. The engine defines the probability of an interactive object to be LONG as a combination based on the probabilities of the engine precision and the visual precision. Interaction with touchscreens in work or home environments will be completely different to the interaction that takes place in a moving vehicle (such as a car, metro or bus). These situations could be detected (based on the sensors of the actual device) and we would be able to apply various methods of interaction to each of them. Regarding interaction in movement, the degree of precision for the user suffers significant degradation and this would be detected by the system. When we apply logic rules, when we increase the probability of the driving precision to be low, the chances of generating a bigger size button increase. This is exactly what would be expected in the context we have described, since the use of longer buttons increases the level of precision of the interaction in a moving environment. Once the user has left said environment and the driving precision increases once again, the size of the buttons decreases again.

This simple example that appears to be obvious is not viable to be used with the current user interface design paradigms. All the elements are static and would appear in the same format as they have been previously defined, regardless of the conditions of the user context. Again, there is a seeming ubiquity, but it is completely de-contextualized.

User interfaces tend to be designed to satisfy the needs of the "typical user". The classic design of interfaces does not consider human individuality and diversity. As a consequence, the resulting systems present serious flaws regarding user experience, since those who do not fall into the category of "typical user" will face problems and frustration on using these systems. The use of systems based on the definition of interfaces would be a solution for this problem: the adaptation for users with disabilities. Applications would be possible to adapt to visually impaired users or to users suffering from reduced mobility. The actual system can be adapted, for it has not been designed to be used in a specific way, but thinking of its utilization, of the information that it must know and of what the data exchange with said system should be like.

However, we must not claim that this would be a perfect, problem-free solution. There is a major problem that is quite obvious: the total dependence of the system on connectivity, without which it would be impossible to render the user interface. Despite the advantages that the use of local caches might have, as well as the possibility to use the offline mode, it becomes quite clear that this system might still be proposing a limited scenario. In any case, we must also take into consideration that the vast majority of applications we use today, despite unfolding interfaces locally, depend almost entirely on a connection to be able to operate. The apparent ubiquity we mentioned earlier on this paper and that stems from the synchronization of data from the Cloud would still be hindering the optimal use of the application, which, even when it is clearly a weakness, does not necessarily mean that native applications are entirely free from it.

Along the same lines, another issue could be a delay in the network that could cause great frustration to the user if there is not an almost immediate response between call and return. Once again, this is a situation that might well happen today, as our dependence on Cloud services is almost total.

As per the advantages of this system, besides the ones that we mentioned in a previous section of this paper, i.e. the fact that they can be cross-platform, used in multiple devices and for general purposes for every type of user, regardless of their condition, we must mention other additional benefits:

- Scalability. It is scalable not only at a computational level, but also at a functional level. Given the fact that all the processes reside in the Cloud, it is relatively simple to add new models of interaction. This scalability applies equally to the number of devices that can be connected to our system.
- A much more efficient user experience. Improvements in the system can be universally applied to all users and devices. With a numerous user contingent we can undertake mass usability tests, significantly increasing the processes of interaction and thus generating functional improvements.

VII. Conclusion

Every strategy designed to reduce the human and technological limitations and achieve a significant increase in the potential use of any technological tool should include the creation of a wide range of communication channels through which humans and machines can exchange information. According to the type of user or device that makes use of the service, we can adapt not only the communication channel to be used but also the way in which information is perceived by the user, i.e. the interface, using even the knowledge that the system might possess about the user context.

The human-machine interaction must be considered as something more than the simple use of a touchscreen and comprehensively and universally include any type of device, interaction and human condition, so that technology becomes a means to unite, not to separate, for those who use different platforms, for those who suffer from certain disabilities or for those who, in certain situations, cannot use certain types of interaction.

Even though it is down to users to demand these technological advances, it is also the responsibility of device and operative systems manufacturers to approach technology with a long-term vision, through the use of open standards and protocols, thus facilitating benefits for all the parties involved. It might be that today's business models, based on pay-per-download, or even in-app sales and subscriptions that are inaccessible for providers in app stores are not ideal and should be looked into. It may also be convenient to test new models that, in accordance with the ubiquity requirements that future electronic devices might present, continue to allow providers to have satisfactory account balances.

An understanding approach to the user context in the upcoming years will become one of the major axis of online sales channels (particularly from mobile devices), offering the user what can be inferred that they are going to need and/or want according to who they are, where they are and their previous activity with information systems. If these means can be used to "recommend" the purchase of specific goods and services, why should they not be used to improve the interface user experience through the use of contextual ubiquity?

Total ubiquity is advantageous not only for users themselves, but also for any sphere of their everyday lives. It is not a technological matter, but a social and ethical one. The Internet's neutrality is a widely known concept and this neutrality should be made extensive to applications and services, since these represent the digital profile of every individual. The owner should thus be free to export said profile to any platform that they deem appropriate.

Acknowledgements

This paper is being supported by the VISIO Project (http://www.bitmonlab.com/rdi/Visio/index.html), funded by the "Acción Estratégica de Economía y Sociedad Digital" programme of the Ministry of Industry, Energy and Tourism of Spain. The VISIO project is collaboration between the University of Oviedo, Spain and Zed Worldwide, Spain.

References

[1] Apple App Store press release, July 14, 2008.

[2] http://www.apple.com/pr/library/2008/07/14iPhone-App-Store-Downloads-Top-10-Million-in-First-Weekend.html

[3] Removal of restrictions can decrease music piracy

[4] http://business.rice.edu/jgsb2011_2column.aspx?id=4294967941

[5] SAMSUNG Revolutionizes the Viewing Experience With Innovative New SUHD TV, January 05, 2015. http://www.samsung.com/us/news/24393

[6] Panasonic and Mozilla to Collaborate on New Firefox OS Open Platform for Next Generation Smart TVs, January 6, 2014. http://www2.panasonic.com/webapp/wcs/stores/servlet/prModelDetail?storeId=11301&itemId=705511

[7] LG Electronics Acquires webOS from HP to Enhance Smart TV, February 2, 2013. http://www.lg.com/us/press-release/webos-release

[8] Apple Car Play. https://www.apple.com/ios/carplay/

[9] Android Auto. http://www.android.com/auto/

[10] Mirrorlink. http://www.mirrorlink.com/

[11] The Internet of Things. How the Next Evolution of the Internet Is Changing Everything. Dave Evans, Cisco, Cisco Internet Business Solutions Group (IBSG), April 2011. http://www.cisco.com/web/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf

[12] Samsung Smart TV a spy in the living room as webcam hack revealed, August 2, 2013. http://www.slashgear.com/samsung-smart-tv-a-spy-in-the-living-room-as-webcam-hack-revealed-02292655/

[13] Hacker Says Attacks On 'Insecure' Progressive Insurance Dongle In 2 Million US Cars Could Spawn Road Carnage. Forbes, January 1, 2015. http://www.forbes.com/sites/thomasbrewster/2015/01/15/researcher-says-progressive-insurance-dongle-totally-insecure/

[14] BMW sounds alarm over tech companies seeking car data. Fiancial Times. http://www.ft.com/intl/cms/s/0/685fe610-9ba6-11e4-950f-00144feabdc0.html#axzz3PG1kItGq

[15] Lizard Stresser Runs on Hacked Home Routers. Krebs on Security, January 2015. http://krebsonsecurity.com/2015/01/lizard-stresser-runs-on-hacked-home-routers/

[16] Active Video. http://www.activevideo.com/

Jesus Iglesias Feijoo has worked in Software Development over fifteen years. After creating his own Internet company in 2000, he joined Zed Worldwide where he has been leading international software projects during the last ten years. Nowadays he works as r&d project manager and leads an innovative development team. He maintains his entrepreneur activity developing successful apps for several platforms.

Guillermo Amat Gomariz was graduated in Computing Science in the Universitat Jaume I, Castellón, Spain, in 2003. He is a R&D Project Manager at Zed World Wide where he has lead several Smart City projects. His major research interests are Geographical Information Systems applied to Smart Cities, the Internet of Things and software portabilityInformation Systems applied to Smart Cities, the Internet of Things and software portability.