BIG IoT – Bridging the Interoperability Gap of the Internet of Things

Deliverable 3.3b: Security and Privacy Design for Smart Objects

Version 1.2

Delivery Date: 29.09.2017

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 688038.

Document Information	
Responsible Person and Affiliation	Stefan Schmid (BOSCH)
Due Date / Delivery Date	September 29, 2017
State	Final (One of: Draft, Reviewed, Final)
Reviewers	Sebastian Kaebisch (SIEMENS), Mirco Schindler (TUC)
Version	1.2
Confidentiality	Public

Version	Description of Changes	Date of Resolution
1.0	Draft version for internal reviewers	08.09.2017
1.1	Final version for the internal reviewers	26.09.2017
1.2	Final version	29.09.2017

List of Authors

Organization	Authors	Main organizations' contributions
BOSCH	Stefan Schmid	Contributions to section 5 and 6.
UPC	Juan Hernández-Serrano, Jose L. Muñoz, Oscar Esparza, Olga León, Ferran Quer	Coordination of discussion/roadmap. Contributions to sections 1, 2, 3, 4, 5 and 6
AAU	Lars Mikkelsen,	Contributions to sections 1, 3 and 6

	Hans-Peter Schwefel, Tatiana Madsen	
SIEMENS	Arne Bröring	Contributions to sections 1, 2, 3 and 6.
ATOS	Wolfgang Schwarzott	Contributions to sections 1, 2, 4 and 6

Abbreviations

Abbreviation	Meaning
BCN	Barcelona (Pilot)
BIG IoT	Project title: Bridging the Interoperability Gap of the Internet of Things
DOA	Description of Action
GRP	Incentive-based Green Route Planning (Use Case Cluster)
HBN	Healthy Bike Navigation (Use Case Cluster)
IoT	Internet of Things
MRO	Multi-modal Route Optimizer (Use Case Cluster)
NG	Northern Germany (Pilot)
PIE	Piedmont (Pilot)
PTO	Public Transport Optimization (Use Case Cluster)
SBS	Smart Bike Sharing (Use Case Cluster)
SC	Smart Charging (Use Case Cluster)
SP	Smart Parking (Use Case Cluster)
STM	Smart Traffic Management (Use Case Cluster)

Table of Contents

List of Figures

List of Tables

Summary

This is the second version of D3.3 – Security and Privacy Design for Smart Ob-jects. The document presents the work performed in T3.3 of the BIG IoT project up to month 21. For a better understanding of the contents, it is advisory to also read BIG IoT architecture deliverable D2.4, as concepts from this deliverable will not be fully reiterated here.

This document is structured as follows. Section 1 (minor updates from D3.3a) presents an Introduction to the BIG IoT project from a security point of view. Section 2 and 3 (minor updates from D3.3a) identify and propose security and privacy requirements and best practices for the BIG IoT ecosystem. Section 4 (new in D3.3b) proposes a Risk Rating Methodology for the BIG IoT ecosystem that would be later used to assess the risks in the BIG IoT pilot services. Section 5 (completely revisited from D3.3a and also expanded) analyses the authentication/authorization flows in the BIG IoT API/Marketplace and details the ASVS approach to the BIG IoT API. Section 6 (new in D3.3b) presents a vulnerability and risk analysis of services currently online for the BIG IoT pilots. Finally, section 7 (minor updates from D3.3a) shows the concluding remarks.

This deliverable will be updated and expanded by D3.3.c (M30).

1. Introduction

In the past years, the Internet of Things (IoT) has largely expanded and the number of IoT devices is evermore increasing. Today, IoT use cases span over a wide variety of application domains, ranging from smart homes over e-health systems to industrial environments. Things used in such applications are made available through IoT platforms. These platforms can be located on the device, fog, or cloud level.

A multitude of such platforms exists today. In order to enable cross-platform and even cross-domain application development, different initiatives are determined to form IoT ecosystems. An example for such an ecosystem initiative is BIG IoT [1]. The BIG IoT project comprises currently 8 IoT platforms and will quickly grow in the coming months. To ignite such an IoT ecosystem, BIG IoT focuses on establishing interoperability across platforms.

BIG IoT has two main objectives. The first one is defining a shared interface, i.e., the so-called BIG IoT API comprising common functionalities such as discovery, access, and event handling. This API needs to be supported by all participating platforms, often in addition to their existing proprietary interface, as illustrated in Figure 1. The second objective is establishing a centralized marketplace where platforms as well as value-adding services can be registered, searched, and subscribed for by applications. In the BIG IoT project, these technologies are deployed in multiple pilot scenarios and involving various IoT platforms, services, and applications from the Smart Cities domain. We provide examples of these scenarios in Sec. 6.

Besides the evident benefits that can be achieved by such IoT ecosystems, there are crucial challenges to deal with. In particular, new security threats must be addressed to allow the continued growth of such ecosystems. Frequently, sensitive data are stored, sent, or received by IoT platforms. Thus, security mechanisms are needed to protect these data from unauthorized access. Consider a patient who is wearing a glucose sensor that transmits its results to the IoT platform of a medical centre. Security vulnerabilities may allow other entities to misuse this information or even put at risk the physical safety of the patient if these data are forged.

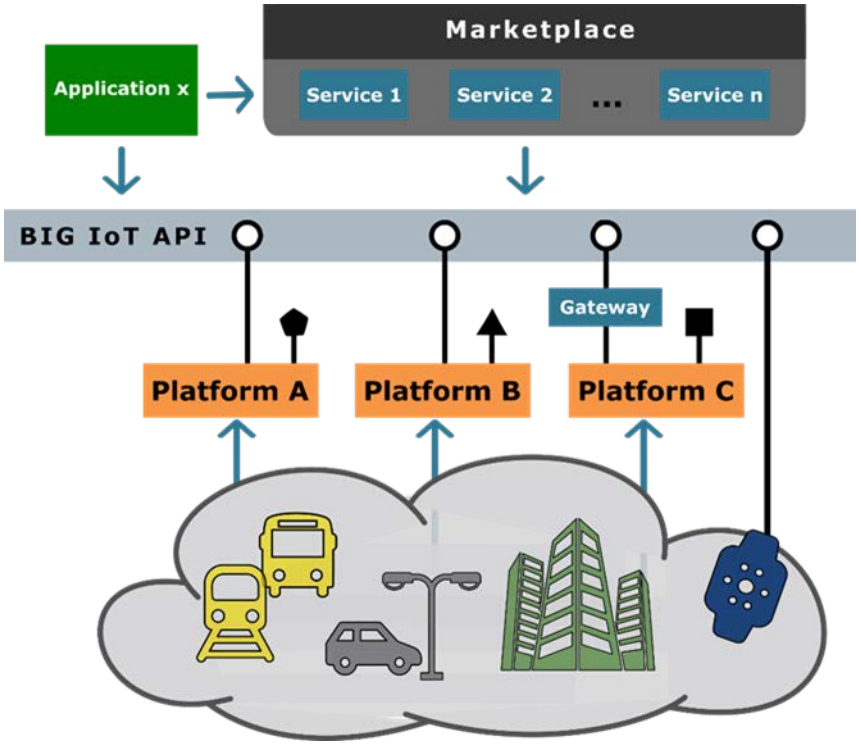

Figure 1 - The BIG IoT approach for building an ecosystem of IoT platforms.

Dealing with IoT security risks is challenging and can be more complex than in conventional networks, particularly for companies entering IoT ecosystems without any experience in the security field. Moreover, as new security vulnerabilities may be discovered over time, there is a need for updating IoT platforms on a regular basis. This might be hard to achieve in some cases either due to the simplicity of some device-level IoT platforms, or due to the lack of awareness of users or platform administrators that forget or just skip updates. Finally, it may happen that some IoT platform manufacturers decide not to provide ongoing support nor security updates in order to reduce costs.

Last, but not least, privacy must be a mandatory concern. A privacy analysis should find an appropriate answer to the question: do the collected data allow drawing conclusions on individual human beings or onto small, specific groups of human beings? Note that such conclusions may be drawn by an unauthorized eavesdropper, and then this discussion is overlapping with confidentiality. The purpose of this document is to outline current discussion, analysis, and specific actions with regard to security and privacy in IoT ecosystems, and particularly to the BIG IoT realization of such an ecosystem. Requirements and best practices presented here

will help to secure all the assets of the BIG IoT ecosystem and to prevent abuse of sensitive data.

2. Securing the BIG-IoT ecosystem: a general approach

The BIG IoT marketplace, the common API, and all the plat-forms/services/applications in the ecosystem must comply with a set of security requirements. After an analysis of the BIG IoT needs, seven security requirements were identified, which are presented in Section 2.1. Moreover, in order to face these risks, some solutions were already discussed (see Section 2.2).

2.1. Requirements

1. End-to-end security: IoT communications typically spread over several nodes and technologies. In particular, BIG IoT is not another IoT platform; it is a framework for a heterogeneous set of platforms, services, and applications. A possible solution to provide security would be to leave the mechanisms already in use for each platform, and then to de-fine adaptation policies of these mechanisms in the boundary points of platforms. The definition of these "low-level" relationships would highly increase complexity, as each individual security protocol (suite) provided by a component would have to be mapped to each protocol (suite) offered by each component it communicates with, which may fail, and hence should be avoided. The solution adopted in BIG IoT is to provide security at the API level, because it is common for all platforms. Therefore, there is no need to adapt protection mechanisms between platforms, as the API is end-to-end by nature and assures that security remains independent of low level platform components.

2. "Batteries included but swappable": BIG IoT has to be designed to be capable of ageing in place while still addressing evolving risks [2]. There may appear new attacks, crypto systems, counter measures, techniques, and topologies, but the IoT system must be capable of dealing with these emerging concerns long after the system was deployed. Consequently, BIG IoT must ship a default but swappable security implementation, not hard-coded to specific security protocols/systems. Therefore, the aforementioned API-level security mechanisms need to be modelled and made exchangeable. An initial implementation of course still needs to be included as mandated by Req. 1.

3. Flexible authentication/authorization: The authentication and authorisation systems used in the BIG IoT ecosystem must ease the management of identities and permissions. Features

like single sign on and authentication without intervention of the BIG IoT authentication manager are key. Therefore, decentralized, federated or delegated authentication must be supported.

4. Ownership transfer: BIG IoT should support safe transfer of ownership, even if a component is sold or transferred to a competitor; something that often happens during the lifespan of IoT nodes/components.

5. Accounting and charging: The BIG IoT must implement a secure ac-counting of resources consumption. This accounting must generate enough charging data, typically in the form of a Charging Data Record (CDR), so that the desired charging policies can be enforced. As a result of a charging policy, a billing system may be necessary to generate in-voices for service consumers. All these systems must be flexible enough to implement different business models and monetization strategies of services that can be implemented in the BIG IoT ecosystem. The BIG IoT marketplace must support offline and online charging and billing.

6. Continuous security: The BIG IoT system should be ready to respond to hostile participants, compromised nodes, and any other adverse event. Therefore, it is necessary to implement mechanisms and/or tools to re-issue credentials, exclude participants, distribute security patches, up-dates, swap algorithms, or protocols, etc.

7. Secure development: Security must be a key part during the design phase of every BIG IoT software, but a secure design would be useless if development errors open unexpected attacks and/or vulnerabilities. Using a Secure Software Development Life Cycle (S-SDLC) and secure Source Code Analysis (SCA) would help developers to build more se-cure software and address security compliance requirements.

2.2. Addressing the security requirements in BIG IoT

Even though many strategies or decisions are still to be taken, some actions have already been adopted in order to address the above requirements. Requirement 1 is directly meet as the BIG IoT API is an HTTP(s) based API, and so it is end-to-end by design. Moreover, in order to comply with Requirement 2, the API should be flexible enough to handle any protocol and/or content. BIG IoT handles this by defining a very generic API; semantic annotations of the syntactic descriptions of each registered service and platform are then used to clarify the details on how to establish communication with these components.

Requirement 3 states that there is a need of providing flexible authentication in the IoT ecosystem. I.e., BIG IoT must implement an authentication and authorization system to be shared by participating platforms, services, applications, and end-users. Moreover, BIG IoT has to be able to work even when the authentication man-agers are not available. To solve this, BIG IoT uses an approach that is similar to the ones used by other widely-known IoT initiatives (e.g., [3]): signed manifests or tokens. A client presents a signed manifest to a server to demonstrate that it is able to perform a given action on a given asset.

When the server receives the signed manifest, it can trust the contents be-cause the manifest is signed by a common centre of trust. Many state-of-the-art technologies have already dealt with the fact of using such signed manifests. Most solutions for the Web use JSON, CBOR, or XML encodings and rely on JSON Web Encryption (JWE) [4], JSON Web Signature (JWS) [5], XML Encryption (XML-Enc) [6] or XML Signature (XML-Sig) [7]. Obviously, one can decide to design a custom solution from scratch, which may seem at-a-glance a better suited solution. However, experience tells us that security protocols are subtle and often tricky. Consequently, the BIG IoT position is to adopt existing, already tested, security technologies. Given that the BIG IoT API relies on HTTP REST, potential candidates are SAML [8], OAuth [9], OAuth 2.0 [10], OpenID Connect [11] (on top of OAuth 2.0), supporting delegated authorization and authentication/identification. Optionally, defining custom authorization/authentication flows to address some of the specifics of the BIG-IoT API.

Requirement 4 should also be considered in the choice of the previous technology. The authentication/authorization system has to be defined with focus on easy management of identities and permissions, easing actions that are quite com-mon in the IoT. This includes safe transfer of resources' ownership and quick response to dynamic topologies with frequent admissions and withdrawals.

Section 5.2 addresses in detail the different authentication/authorization flows in use in both the BIG IoT marketplace and the BIG IoT API.

Requirement 5 states that an appropriate accounting is key to develop charging/billing systems, both offline or online. An offline charging system just stores a CDR containing the relevant accounting and charging information (starting and ending time, data used, bandwidth, etc.). Then, the user is charged after resources have been used. In general, users being charged offline provide a bank account to pay the corresponding bill. On the contrary, when using online charging, the user typically buys a prepaid amount of credit. In this case, the charging

system has to monitor online the resources consumption and then, needs to stop (or constrain) the service when the credit limit is reached. In both approaches (offline and online), it should be desirable to have non-repudiation proofs for both, the users and the marketplace to be able to verify consumptions, bills, etc. and to solve possible inconsistencies.

Requirement 6 forces the marketplace to host a secure repository where to securely download software and software updates/patches. This is a challenge that has often been addressed in the past and present. Experience here says that, apart from security, success depends on the ease of use for both end users and developers. The app stores of Apple, Google and Amazon are good examples, but BIG IoT is aiming for a more open approach for this component.

Requirement 7 makes mandatory the use of S-SDLC. To accomplish this, BIG IoT developers have to make use of the best practices for secure software development set up by the Open Web Applications Security Project (OWASP) [12]. First, the organization itself has to fulfil security related activities and software security practices, which are described in the OWASP Software Assurance Maturity Model (SAMM) [2] framework. Second, the applications have to meet requirements based on the OWASP Application Security Verification Standard (ASVS) [13]. Third, the application source code has to be analysed according the OWASP Code Review Guide [14]. Finally, fourth, the application will be tested for vulnerabilities and design flaws according the OWASP testing guidelines [15].

The OWASP SAMM framework builds the foundation of a secure development environment and organization. The BIG IoT development organization shall follow the twelve security practices and carry out the activities listed there at least to maturity level 2 but the ultimate goal should be to incorporate also the level 3 activities.

BIG IoT engineers currently lean towards the OWASP ASVS to define the security requirements for the applications and services. This standard (in its current version) defines 19 verification requirements. All these requirements have three security verification levels, with each level increasing in depth: ASVS Level 1 "Opportunistic" is meant for all software and its compliance adequately defends against application security vulnerabilities that are easy to discover; ASVS Level 2 "Standard" is meant for applications that contain sensitive data, such as business-to-business transactions, including those that process health-care information, implement business-critical or sensitive functions, or process other sensitive assets; and ASVS Level 3 "Advanced" is meant for the most critical applications, that is, applications that perform high value

transactions, contain sensitive medical data, or any application that requires the highest level of trust. Responsibilities include controls for ensuring confidentiality (e.g. encryption), integrity (e.g. transactions, input validation), availability (e.g. handling load gracefully), authentication (including between systems), non-repudiation, authorization, and auditing (logging). Each ASVS level contains a list of security requirements, and each of these requirements can also be mapped to security-specific features and capabilities that must be built into software by developers.

For BIG IoT, developers should (at least) follow the ASVS level 2 requirements, and they could complete these with level 3 requirements according to the appropriate criticality. The task of SCA for the BIG IoT software will be based on the recommendations listed in the OWASP SCA guidelines. The Second Edition of the Code Review Guide has been developed to advise software developers on the best practices in secure code review, and how it can be used within S-SDLC. The SCA for the BIG IoT software should be done for all code by means of source code analysis tools, specialized on finding security related bugs. Also, all critical software parts will be manually reviewed.

Security testing of BIG IoT applications and web services will be based on the OWASP testing guidelines. The testing shall be performed manually by skilled penetration testers, but supported by a wide variety of automated tools. In the design phase, developers should use automated tools for as much testing as possible, executing unit and integration tests for specific and relevant fuzz and abuse cases.

3. Best practices for privacy in IoT ecosystems

Igniting an IoT ecosystem involves handling big data. Often these data contain sensitive information and therefore their use could be a threat to users' privacy.

The BIG IoT consortium is fully mindful of the ethical aspect and the social impact of the IoT ecosystem and it absolutely respects the ethical rules and standards of H2020, as well as those reflected in the Charter of Fundamental Rights of the European Union and the General Data Protection Regulation (GDPR) (Regulation (EU) 2016/679). The primary objectives of the GDPR are to give control back to citizens and residents over their personal data and to simplify the regulatory environment for international business by unifying the regulation within the EU and this is something that BIG IoT must encourage among its users and developers.

The FTC published in 2015 a guide containing best practices for privacy in IoT [16] that is summarized with the following statement: while flexibility in terms of data gathering is key to innovate around new uses of data, the amount of data storage should be balanced with the interests in limiting the privacy and data security risks to consumers. With such a goal, the FTC mainly focuses on three topics: data minimisation, strong accountability, transparency and easy access.

The best practices published by the FTC are also clearly aligned with the new GDPR, which states "In order to be able to demonstrate compliance with this Regulation, the controller should adopt internal policies and implement measures which meet in particular the principles of data protection by design and data protection by default. Such measures could consist, inter alia, of minimising the processing of personal data, pseudonymising personal data as soon as possible, transparency with regard to the functions and processing of personal data, enabling the data subject to monitor the data processing, enabling the controller to create and improve security features". However, it is important to notice that the EU place an emphasis on introducing privacy by default and privacy by design policies; and the BIG IoT consortium is committed to it.

In the following, we provide the main ideas behind the FTC recommendations and how BIG IoT is addressing them. Nevertheless, they are rather generic and they should be always complemented with a specific analysis of every use case (an ex-ample is provided in Sec. 5).

3.1. Data minimisation

Data minimisation is a long-standing principle of privacy protection [17] that means that a data controller should limit the collection of personal information to what is directly relevant and necessary to accomplish a specific purpose. Since users' privacy is (or it should be) key for a wide adoption of the IoT, data minimisation is key to fostering the IoT ecosystem. Indeed, data minimisation can help guard against two privacy-related risks.

First, storing huge volumes of data increases the likelihood of receiving a data breach since there is more potential harm derived from such an event.

Second, collecting and storing large amounts of data also increases the risk of using the data in a way that departs from consumers' reasonable expectations.

To minimise these risks, organizations should develop data minimisation policies and practices providing answers to questions like what types of data it is collecting, to what end, and how long it should be stored. Such an exercise is part of a privacy-by-design approach and helps ensure that a company is sensitive with data collection practices.

In the EU, the data minimisation principle derives from Article 5.1(b) and (c) of the GDPR, which state that personal data shall be "collected for specified, explicit and legitimate purposes" and it shall be "adequate, relevant and limited to what is necessary in relation to the purposes for which they are processed".

When a company needs to gather and store sensitive data with a business purpose, it should consider whether it could do so with a deidentified data set. Deidentified data can reduce potential consumer harm while still promoting beneficial societal uses of the information.

A key to effective deidentification (anonymization) is to ensure that the data cannot be reasonably reidentified even with external cross-sources. This usually requires removing identifiers or pseudoidentifiers. Although, at first glance it seems quite affordable, recognizing non-evident identifiers is quite a challenge that often has to be faced in a manual, use-case-specific manner.

In BIG IoT, for every specific use case, an analysis of potential identifiers among the data and/or metadata stored/exchanged is being performed. Data minimisation is encouraged specifically for the BIG IoT platforms and should account for cross data not only from other BIG

IoT platform/services, but also from any other external source. An example for such data minimization technologies in the context of a BIG IoT use case is given in Sec. 5.

Notice that there is a common misconception about the added costs for data minimisation. Enhancing privacy by means of data minimisation techniques does not necessarily imply added costs. Indeed, data minimisation reduces the sensitive-ness of data and hence lower security would be required. Consequently, for in-stance, in BIG IoT, important saving can be obtained in development costs due to a reduced ASVS level compliance.

3.2. Strong accountability

As aforementioned, deidentified data sets can reduce many privacy risks. However, there is always a chance that supposedly deidentified data could be reidentified; especially because of the technology advances. For this reason, companies should have accountability mechanisms in place. In this context, the FTC has stated that companies stating that they maintain deidentified or anonymous da-ta must meet three actions: (1) take reasonable steps to deidentify data, including by keeping up with technological developments; (2) publicly commit not to reidentify the data; and (3) have enforceable contracts in place with any third parties with whom they share the data, requiring the third parties to commit not to reidentify the data. This approach ensures that if the data are reidentified in the future, regulators can hold the company responsible.

Consequently, BIG IoT platforms, services, and applications should provide proper accounting mechanisms to securely log any action by any actor dealing with sensitive data.

3.3. Transparency and easy access

The centrepiece legislation at EU level in the field of data protection is, until the final EU-wide adoption of the GDPR on May 2018, the “Data Protection Directive” [17] which is implemented in EU Member States through national laws. This directive aims to protect the rights and freedoms of persons with respect to the processing of personal data by laying down guidelines that determine when the processing is lawful. The guidelines mainly relate to the quality of the data, the legitimacy of the processing, the processing of special categories of data, information to be given to the data subject, the data subject’s right of access to data, the right

to object to the processing of data, the confidentiality and security of processing and the notification of the processing to a supervisory authority. The Directive also sets out principles for the transfer of personal data to third countries and provides for the establishment of data protection authorities in each EU Member State.

In general, the conclusion is that EU's individuals need better information on data protection policies and about what happens to their data when it is processed by online services. As a result, the EU will require European organizations to pub-lish transparent and easily accessible data protection policies. In this context, simple icons on websites and applications could explain how, by whom and under whose responsibility personal data will be processed. As a consequence, users are better informed about how and if their personal data is being exploited.

According to that conclusion, the upcoming GDPR clearly states "Any processing of personal data should be lawful and fair. It should be transparent to natural persons that personal data concerning them are collected, used, consulted or otherwise processed and to what extent the personal data are or will be pro-cessed. The principle of transparency requires that any information and communication relating to the processing of those personal data be easily accessible and easy to understand, and that clear and plain language be used."

BIG IoT is completely mindful and committed to the GDPR. For this reason, the semantic description of what a given service or application consumes and provides should account for privacy issues. For example, it should account for:

- ***Purpose:*** ***non intended***, data are allowed to be used for not initially in-tended purposes); or ***intended***, data are allowed to be used only for in-tended purposes.
- ***Business Transfers:*** ***allowed***, data can be bartered or sold; ***not allowed*** (data cannot be bartered or sold); ***allowed informing***, if business assets are sold or transferred, corresponding information regarding customers could also be transferred.
- ***Law enforcement:*** ***allowed***, data may be given to law enforcement even when legal process is not followed; or ***non allowed***, data may be given to law enforcement only when legal process is followed.
- ***Advertisers:*** ***allowed***, your data can be given to advertisers; or ***non allowed***, your data cannot be given to advertisers.
- ***Retention Policy:*** time that your data is kept (including indefinitely).

Moreover, to better picture this fact, the use of user-friendly, self-explanatory icons has been proposed. Figure 2 shows an example set of privacy icons (proposed by Aza Raskin [18]) that may be used or be a source of inspiration for designing new ones. BIG IoT services and/or applications offering descriptions in the BIG IoT marketplace should use this kind of icons to state clearly how data is going to be pro-cessed. This is a work in progress that it is still to be incorporated in the BIG IoT Marketplace.

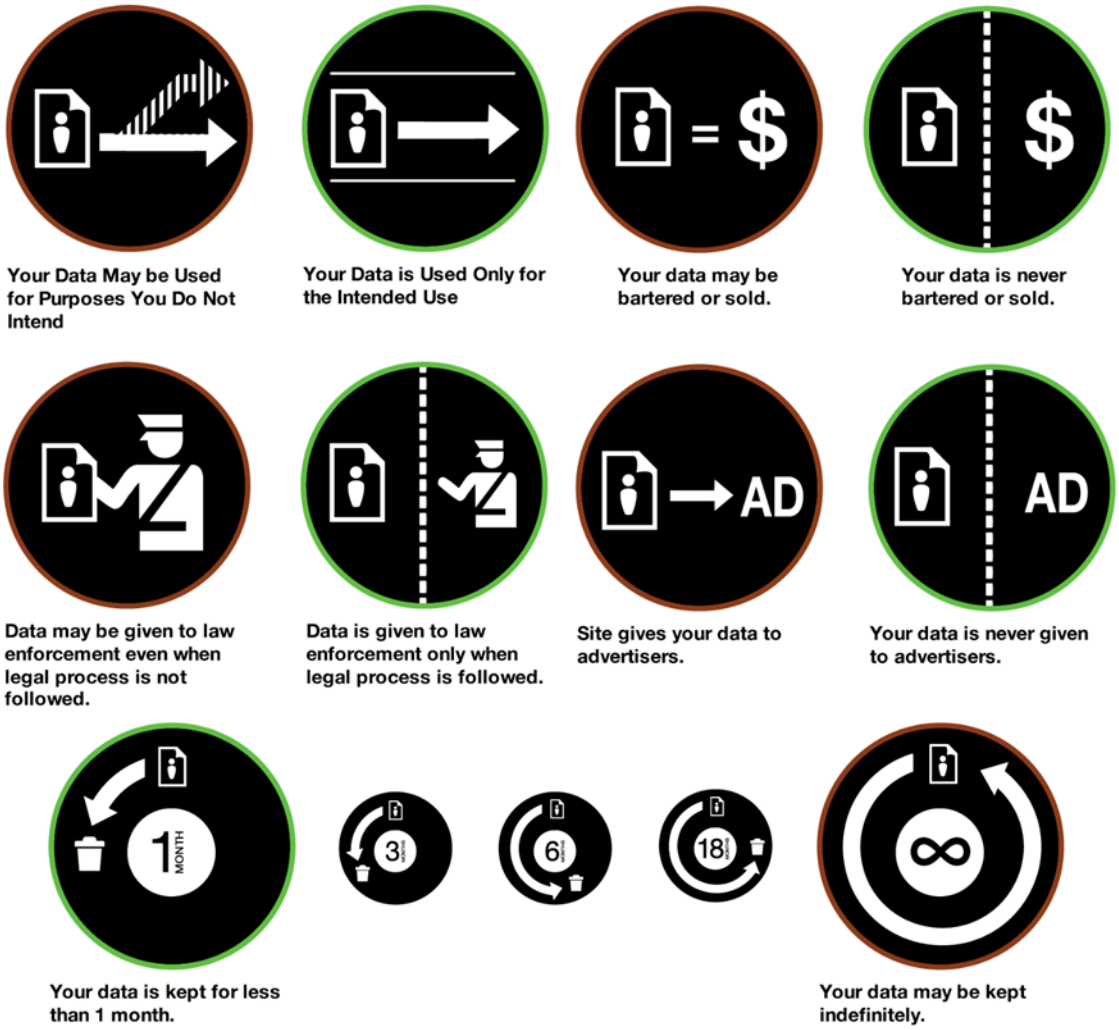

Figure 2 - Privacy icons proposal by Aza Raskin [18]

4. The BIG IoT risk rating methodology

One of the main reasons to integrate security and privacy analysis into the Software Development Life Cycle is to early discovering potential vulnerabilities that could be fixed. However, estimating the associated risk to the business is just as important since the resources available to fix the vulnerabilities will often depend on it.

BIG IoT risk rating methodology is mainly based on the OWASP Risk Rating Methodology [19] with some modifications to address the specifics of the BIG IoT ecosystem. By following the approach below, it is possible to estimate the severity of all of these risks to the business and make an informed decision about what to do about those risks. The obtained rating will help to ensure the business doesn't get distracted by minor risks while ignoring more serious risks that are less well understood.

While the idea would be to get a global risk rating methodology that could be applied in any case, vulnerabilities that could be critical to one organization may not be for another. As a result, we present here a customizable risk rating methodology that can be tuned for a particular organization or use case.

There are many different approaches to risk analysis [20] [21] [19]. The OWASP approach, and therefore the BIG IoT one, is based on these standard methodologies.

Mainly two factors are taking into account in a standard risk model: ***likelihood*** and ***impact***; being the risk usually measured as

$$Risk = likelihood \times impact$$

In the following sections, we first broke down the subfactors that make up "likelihood" and "impact", and show the tester how to combine them to determine the overall severity for the risk. Moreover, we describe the parameters of the model that can be tuned.

4.1. Step 1: Identifying a risk

The first step is to identify a security and/or privacy risk that needs to be rated. The tester needs to gather information about the threat agent involved, the attack that will be used, the vulnerability involved, and the impact of a successful exploit on the business. There may be multiple possible groups of attackers, or even multiple possible business impacts. In general,

it is advisable to be cautious by using the worst-case option, which will result in the highest overall risk.

4.2. Step 2: Factors for estimating likelihood

Once the tester has identified a potential risk and wants to figure out how serious it is, the first step is to estimate the "likelihood". At the highest level, this is a rough measure of how likely this particular vulnerability is to be uncovered and exploited by an attacker. It is not necessary to be over-precise in this estimate. Generally, identifying whether the likelihood is low, medium, or high is sufficient.

There are a number of factors that can help determine the likelihood. The first set of factors are related to the threat agent involved. The goal is to estimate the likelihood of a successful attack from a group of possible attackers. Note that there may be multiple threat agents that can exploit a particular vulnerability; therefore, it is usually best to use the worst-case scenario. For example, an insider may be a much more likely attacker than an anonymous outsider, but it depends on a number of factors.

Note that each factor has a set of options, and each option has a likelihood rating from 0 to 9 associated with it. A weighted sum of these numbers will be used later to estimate the overall likelihood.

4.2.1. Threat agent factors

The first set of factors are related to the threat agent involved. The goal here is to estimate the likelihood of a successful attack by this group of threat agents. Use the worst-case threat agent.

Skill level

How technically skilled is this group of threat agents? The more skills required the less likelihood to find an attacker. Therefore, security penetration skills (1), network and programming skills (4), advanced computer user (5), some technical skills (7), no technical skills (9)

Motive

How motivated is this group of threat agents to find and exploit this vulnerability? Low or no reward (1), possible reward (4), high reward (9). Estimating the potential reward is so difficult

that the weight of this factor can be usually reduced. As a reference, in BIG IoT a default value of 0.5 is assumed.

Opportunity - Resources

What resources and opportunities are required for this group of threat agents to find and exploit this vulnerability?

- Full access or expensive resources required (0). E.g. Access to the equipment which is locked and has an alarm system (so that the lock cannot be easily broken); sniffing of data traffic by breaking into a protected ISP network first.
- Special access or resources required (4)
- Physical access to the equipment (which is not strongly protected) or some more complicated SW tools to overcome ICT barriers (7). E.g. Physically attaching to an Ethernet that is not protected otherwise and physically accessible; sniffing on a weakly protected (e.g. Using former WEP) WLAN with known-plaintext attacks via sending email.
- Can be done by readily available off-the-shelf equipment, without any constraints on physical presence or special software (9). E.g. Attack can be performed from standard PC connected to the Internet with standard SW.

Size

The scope of the attack. How many potential victims (users or devices) can be involved after a successful attack: just one (1); tens of individuals (5); thousands (9).

4.2.2. Vulnerability factors

The next set of factors are related to the vulnerability involved, if the vulnerability is already known. The goal here is to estimate the likelihood of the particular vulnerability involved being discovered and exploited. If a vulnerability is not known (although a risk has been identified), the following factors, with the exception of Intrusion Detection, should be weighted to 0.

Ease of discovery

How easy is it for this group of threat agents to discover this vulnerability? Practically impossible (1), difficult (3), easy (7), automated tools available (9)

Ease of exploit

How easy is it for this group of threat agents to actually exploit this vulnerability? Theoretical (1), difficult (3), easy (5), automated tools available (9)

Awareness

How well known is this vulnerability to this group of threat agents? Unknown (1), hidden (4), obvious (6), public knowledge (9)

Intrusion detection

How likely is an exploit to be detected? Active detection in application (1), logged and reviewed (3), logged without review (8), not logged (9)

4.3. Step 3: Factors for estimating impact

When considering the impact of a successful attack, it is important to realize that there are two kinds of impacts. The first is the "technical impact" on the application, the data it uses, and the functions it provides. The other is the "business impact" on the business and company operating the application.

Ultimately, the business impact is more important. However, you may not have access to all the information required to figure out the business consequences of a successful exploit. In this case, providing as much detail about the technical risk will enable the appropriate business representative to make a decision about the business risk.

Again, each factor has a set of options, and each option has an impact rating from 0 to 9 associated with it. We will use these numbers later to estimate the overall impact.

4.3.1. Technical impact factors

Technical impact can be broken down into factors aligned with the traditional security areas of concern: confidentiality, integrity, availability, and accountability. The goal is to estimate the magnitude of the impact on the system if the vulnerability were to be exploited.

Below factors are defined to address both security and privacy analyses. Therefore, when performing a privacy-only analysis, they should be weighted accordingly; e.g. Loss of integrity may be not applicable and then weighted to 0.

Loss of privacy

How much data could be disclosed and how sensitive is it?

- Minimal non-sensitive data disclosed or information that is anyways also published via other channels, e.g. websites (2).
- Information that can be used to infer about the behaviour about a human being without being able to identify the human being (4).
- Minimal critical data disclosed o extensive non-sensitive data disclosed (6).
- Information that allows to infer about the behaviour about an individual or a group of people and can be linked back to an identity or group of identities with additional knowledge (7)
- Personal information that is directly linked to an individual (9)

Loss of integrity

How much data could be corrupted and how damaged is it? Minimal slightly corrupt data (1), minimal seriously corrupt data (3), extensive slightly corrupt data (5), extensive seriously corrupt data (7), all data totally corrupt (9)

Loss of availability

How much service could be lost and how vital is it? Minimal secondary services interrupted (1), minimal primary services interrupted (5), extensive secondary services interrupted (5), extensive primary services interrupted (7), all services completely lost (9)

Loss of accountability

Are the threat agents' actions traceable to an individual? Fully traceable (1), possibly traceable (7), completely anonymous (9)

4.3.2. Business Impact Factors

The business impact stems from the technical impact, but requires a deep understanding of what is important to the company running the application. In general, you should be aiming to support your risks with business impact, particularly if your audience is executive level. The business risk is what justifies investment in fixing security problems.

The factors below are common areas for many businesses, but this area is even more unique to a company than the factors related to threat agent, vulnerability, and technical impact.

Financial damage

How much financial damage will result from an exploit? Less than the cost to fix the vulnerability (1), minor effect on annual profit (3), significant effect on annual profit (7), bankruptcy (9)

Reputation damage

Would an exploit result in reputation damage that would harm the business? Minimal damage (1), Loss of major accounts (4), loss of goodwill (5), brand damage (9).

Privacy violation

How much personally identifiable information could be disclosed? One individual (3), hundreds of people (5), thousands of people (7), millions of people (9)

4.4. Step 4: Determining the Severity of the Risk

In this step the likelihood estimate and the impact estimate are put together to calculate an overall severity for this risk. This is done by figuring out whether the likelihood is low, medium, or high and then do the same for impact. The 0 to 9 scale is split into three parts as shown in Table 1.

Likelihood and Impact Levels	
0 to <3	LOW
3 to <6	MEDIUM
6 to 9	HIGH

Table 1 - Likelihood and impact levels

4.4.1. Informal Method

In many environments, there is nothing wrong with reviewing the factors and simply capturing the answers. The tester should think through the factors and identify the key "driving" factors that are controlling the result. The tester may discover that their initial impression was wrong by considering aspects of the risk that were not obvious.

4.4.2. Repeatable Method

If it is necessary to defend the ratings or make them repeatable, then it is necessary to go through a more formal process of rating the factors and calculating the result. Remember that there is quite a lot of uncertainty in these estimates and that these factors are intended to help the tester arrive at a sensible result. This process can be supported by automated tools to make the calculation easier: An excel spreadsheet is provided (see ANNEX A. Risk Assessment Template) to help testers with the analysis of risks with the BIG IoT risk rating methodology.

The first step is to select one of the options associated with each factor and enter the associated number in the table. Then simply take the weighted average of the scores to calculate the overall likelihood. An example can be found in Table 2.

	Threat agent factors				Vulnerability factors			
	Skill level	Motive	Opportunity		Ease of discovery	Ease of exploit	Awareness	Intrusion detection
Weight	1	1	1		1	1	1	1
Score	5	2	7		3	6	9	2
	Overall likelihood (weighted average)=4.375 (MEDIUM)							

Table 2 - Computing overall likelihood

Next, the tester needs to figure out the overall impact. The process is similar here. In many cases the answer will be obvious, but the tester can make an estimate based on the factors, or they can average the scores for each of the factors. Again, less than 3 is low, 3 to less than 6 is medium, and 6 to 9 is high. An example can be found in Table 3.

	Technical Impact					Business Impact	
	Loss of privacy	Loss of integrity	Loss of availability	Loss of accountability		Financial damage	Reputation damage
	1	.5	.75	.5		.7	1
	9	7	5	8		1	2
	Overall technical impact=7.36 (HIGH)					Overall business impact=1.59 (LOW)	

Table 3 - Computing overall impact.

4.4.3. Determining Severity

Once the tester estimates likelihood and impact, as shown in Table 4, it can now combine them to get a final severity rating for this risk. Note that if they have good business impact information, they should use that instead of the technical impact information. However, if they have no information about the business, then technical impact is the next best thing.

Overall Risk Severity				
Impact	HIGH	Medium	High	Critical
	MEDIUM	Low	Medium	High
	LOW	Note	Low	Medium
		LOW	MEDIUM	HIGH
		Likelihood		

Table 4 - Overall risk severity.

In the example above, the likelihood is medium and the technical impact is high, so from a purely technical perspective it appears that the overall severity is high. However, note that the business impact is actually low, so the overall severity is best described as low as well. This is why understanding the business context of the vulnerabilities you are evaluating is so critical to making good risk decisions.

5. Security analysis of the BIG IoT Interface and Marketplace

5.1. Use-case integration modes with Interface and Marketplace

In this section we will analyse BIG IoT applications, services and platforms in terms of the interfaces they will use in their interaction with the marketplace. The interface types are described in more detail BIG IoT D2.4.b.

The goal of this analysis is to highlight how the usage of the different interface types will support security and privacy.

5.1.1. BIG IoT Platforms (Smart Object Platforms)

In the following we will now evaluate the BIG IoT interfaces used along with their security considerations and requirements. In general, IoT sensors make their data available through various IoT platforms that can be integrated into the BIG IoT ecosystem using different integration modes. The default integration mode is mode 1, where the developer extends his IoT platform directly through the BIG IoT Provider Lib. The BIG IoT enabled platforms will all make their data available to BIG IoT Services and Applications by exposing it using the BIG IoT Provider Lib. Integration modes 2 (based on a Gateway Service) and mod 4 (based on a Proxy Service) will also use the BIG IoT Provider Lib, and thus follow the same requirements. Integration mode 3 (for the direct integration of legacy platforms) depends entirely on the functionality already supported by the legacy platform and is thus still under discussion within the project. The relevant security considerations and requirements are therefore omitted here.

Interfaces BIG IoT Platforms will use:

- ***P1*** – programming API used to interact with the BIG IoT Provider Lib. The Provider Lib will make the other interfaces available that are relevant for offering providers.

Interfaces the Provider Lib will use:

- ***M1*** – used by the Provider Lib for authentication on the marketplace. This will also be used to control who is allowed to access an offering. The authentication on the Marketplace will ensure that not everyone is allowed direct access to the data that potentially could be privacy sensitive.

- ***M2*** – used by the Provider Lib for registration of the data as offerings on the marketplace. This is needed to enable services and apps to discover and access the data via the marketplace.
- ***M5*** – used by the Provider Lib to send accounting data to the marketplace. This data is collected during access of Offerings, and then provided to the marketplace as a basis for charging and billing.

Interfaces the Provider Lib or the Platform will implement:

- ***A1*** – the access interface provided by the Provider to allow Consumers (applications or services) to access the data offered on the Marketplace. Here it is important that the A1 interface supports encrypted communication to ensure privacy and security of the information while transferring it.

Looking at the interfaces used, the security will depend much on the Mx interface, especially in terms of authentication and authorization. Note that the P1 interface is an application programming interface and thus internal to the developed or extended platform. The Mx interfaces however are external interfaces. They will only rely meta data about the offerings, i.e. not the actual data offered, as well as accounting and reporting data. In Release 1+2 of the BIG IoT architecture, those interfaces are encrypted using SSL. Upon successful authentication on the marketplace (M1), the Provider Lib will also obtain a valid JWT, which is used in all interactions with the Marketplace. The actual data is transferred via the A1 interface, where it is vital that sufficient security and privacy mechanisms are implemented. Besides the SSL based encryption of the data access between a Consumer and Provider, access control is done based on a so called Offering Accesss Token (JWT), which a Consumer obtains from the Marketplace upon successful subscription to an offering. This token is specifically generated for a given Consumer and subscribed Offering. Upon an access request from a Consumer, a Provider will first check the validity of the Offering Access Token. Only if this has been signed by the Marketplace, and is still valid, the Provider grants the Consumer access to the corresponding Offering.

5.1.2. BIG IoT Services

A BIG IoT Service will consume the data provided by platforms and/or other services and, after processing of the data, they will in turn provide information to other services and apps. For that reason, they must adhere to the security defined for the following interfaces.

Interfaces BIG IoT Services will use

- ***P1*** – see Section 5.1.1.
- ***P2*** – programming API used to interact with the BIG IoT Consumer Lib. The Consumer Lib will make other interfaces available, allowing BIG IoT Applications and Service to discover, subscribe to, and consume offerings on the marketplace.

Interfaces the Provider and Consumer Libs will use:

- ***M1*** – see Section 5.1.1.
- ***M2*** – see Section 5.1.1.
- ***M3*** – used by the Consumer Lib for discovering offerings on the marketplace.
- ***M4*** – used by the Consumer Lib for subscribing to offerings on the marketplace. During subscription to an offering, the marketplace will provide the Consumer Lib an Offering Access Token, which is required by the Consumer to access the offering. The Offering Access Token is a JWT, that is specifically generated by the marketplace for a the subscribing Consumer and the subscribed offering.
- ***M5*** – see Section 5.1.1.
- ***M6*** – used by the Consumer Lib to send reporting data, which are obtained from monitoring the service-level (e.g. response times, throughput, failure rate) obtained while accessing a particular offering on a Provider, to the marketplace. These reports will be used by the marketplace to rate offerings and their Providers. The ratings will be used during offering discovery and selection, e.g. to exclude offerings with a low rating.
- ***A1*** – used by the Consumer Lib to grant a Consumer access to data offered on a Provider. The access to an offering, however, is only available once the Consumer has subscribed to the particular offering via the Consumer Lib. Only upon subscription to an offering, the marketplace will generate and provide the needed Offering Access Token for the subscribed offering to the particular subscribing Consumer. This JWT is used to grant access to the offering resources on the Provider end.

Interfaces the Provider Lib will implement:

- ***A1*** – see Section 5.1.1.

The Px interfaces are application programming interfaces and thus only internal to the developed or extended service. The Mx interfaces are external and connect the Provider and Consumer Libs with the marketplace. In addition to the M1, M2 and M5 interfaces of the Provider Libs, BIG IoT Services will also use the M3, M4 and M6 interfaces of the Consumer Lib. However, the same security concerns and solutions as discussed in Section 5.1.1 apply.

The data is transferred via the A1 interface, where security and privacy mechanisms are needed to protect the data from potential attackers. The relevant security concerns and solutions are discussed in Section 5.1.1.

5.1.3. BIG IoT Applications

BIG IoT Applications are consumer-only. I.e. they use information from various services and/or platforms. For that reason, they must adhere to the security defined for the following interfaces.

Interfaces BIG IoT Services will use

- ***P2*** – see Section 5.1.2.

Interfaces the Provider and Consumer Libs will use:

- ***M1*** – see Section 5.1.1.
- ***M3*** – see Section 5.1.2.
- ***M4*** – see Section 5.1.2.
- ***M5*** – see Section 5.1.1.
- ***M6*** – see Section 5.1.2.
- ***A1*** – see Section 5.1.2.

Most of the interfaces that an application uses will handle meta data and credentials exchange that the application must support in order to be allowed to access the required offerings. As for the services, the application must adhere to the various security mechanisms defined by the providers, such as encryption and credential checking, when accessing the actual data.

5.1.4. Conclusion

From this analysis of interfaces, it can be concluded that implementing encryption on the Mx and A1 interfaces will significantly increase security, as sensitive data is exchanged. This is

especially a requirement posed to the Marketplace and the Offering Providers, because if these define security mechanisms the consumers of the offerings must also follow these to gain access to the offerings. As a consequent, we implement SSL based encryption on those interfaces.

5.2. BIG IoT access control

This section gives a quick overview of the access control on the Marketplace and the BIG IoT API.

5.2.1. Overview

The analysis of the authentication/access control process with regard to the BIG IoT ecosystem and API in this deliverable covers the following stakeholders and architectural components:

- ***Developer/User***: the human authenticating on the Web Portal of the BIG IoT Marketplace to, e.g., sign up and register a new Organization as well as BIG IoT Consumers or Providers.
- ***Identity Provider (IdP):*** an external authentication service the developer uses as entry point for single sign on, authenticating to the IdP, which in turn provides a user identifier to the BIG IoT ecosystem.
- ***Provider:*** A BIG IoT software component offering IoT resources on the marketplace.
- ***Consumer:*** A BIG IoT software component discovering and subscribing to Offerings via the marketplace, and accessing IoT resources on the Provider end.
- ***Marketplace:*** The BIG IoT backend acting as an intermediary between users, developers, providers, and consumers.

At the time of writing this deliverable, developer/user authentication is implemented based on external Identity Providers, like Google and GitHub. We are also assuming in this discussion that we are performing access control on BIG IoT platforms and services based on the marketplace provider and consumer identities (except for integration mode 3, see D2.4.b).

5.2.2. Developer (User) accesses a Marketplace Portal

Initially, a developer needs to register and then log into the marketplace. To support Single Sign-on, this is performed according to Oauth2 and Auth0 as IdP Hub. GitHub and Google accounts can be used by developers to sign in on the marketplace.

During the initial access to the Marketplace Web Portal, the Web request gets redirected to a login page. Since we want to use existing accounts, users get redirected to external sites (e.g. GitHub, Google) to authenticate themselves there (based on Oauth2 Implicit Grant Type). Upon success, the Browser obtains an Oauth2 bearer token, which is used in subsequent accesses to the Marketplace (in the authorization header of the HTTP request). This is illustrated in Figure 3, and will be described in more detail below. Note that the description only applies to the Oauth case, which is used in the actual implementation. The discussion is based on material available on the Web[1].

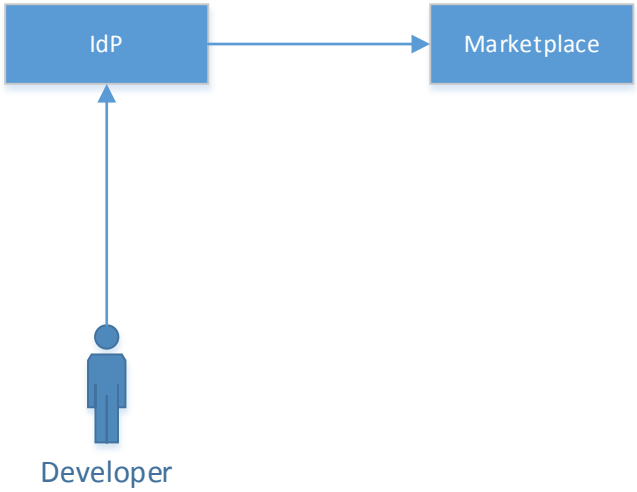

Figure 3 - User authentication using IdP

In more detail: the BIG IoT Marketplace provides a HTML button/Link that contains something like the following URL as a target:

```
https://login.idp.com/oauth?response_type=code&client_id=xxx&re-
direct_uri=xxx&scope=read
```

[1] https://stormpath.com/blog/what-the-heck-is-oauth

When the developer/user clicks this button, their browser will be directed to `login.idp.com`. There, the user/developer will have to authenticate, and may have to allow/deny additional information or operations the BIG IoT Marketplace has requested.

After this, there will be a redirection of the browser, returning it to the BIG IoT marketplace, which was encoded in the `redirect_uri` parameter. Because of the authentication process, an authorization code is now also available, which will be encoded in the URL like this:

```
https://market.big-iot.org/oauth/callback?code=xxx
```

The `code` query string value will then be exchanged for an access token using the IdP:

```
POST https://api.idp.com/oauth/token?grant_type=authoriza-
tion_code&code=xxx&redirect_uri=xxx&client_id=xxx&client_se-
cret=xxx
```

As response to this post request, the marketplace will receive an *access token* that then can be used to make additional calls to the IdP and retrieve the user's/developers information needed.

Response:

```
HTTP/1.1 200 OK
Content-Type: application/json;charset=UTF-8
Cache-Control: no-store
Pragma: no-cache
Set-Cookie: ac-count=eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiIxNDQyN-
mQxMy1mNThiLTRhNDEtYmVkZS0wYjM0M2ZjZDFhYzAiLCJpYXQi-
OjE0Mzg3MDQxMjg-
sInN1YiI6Imh0dHBzOi8vYXBpLnN0b3JtcGF0aC5jb20vdjEvY-
WNjb3VudHMvNW9NNFdJM1A0eEl3cDRXaURiUmo4MCI-
sImV4cCI6MTQzODk2MzMyOH0.wcXrS5yGtUoe-
wAKqoqL5JhIQ109s1FMNopL_50HR_t4; Expires=Wed, 05-Nov-2016
16:02:08 GMT; Path=/; HttpOnly
{
    "access_token":  "eyJhbGciOiJIU-zI1NiJ9.eyJqdGkiOiIxNDQyN-
mQxMy1mNThiLTRhNDEtYmVkZS0wYjM0M2ZjZDFhYzAiLCJpYXQiO-
jE0Mzg3MDQxMjgsInN1YiI6Imh0dHBzOi8vYXBpLnN0b3JtcGF0aC5jb20vdjEv-
```

```
YWNjb3VudHMvNW9NNFdJM1A0eEl3cDRXaURiUmo4MCI-
sImV4cCI6MTQzODk2MzMyOH0.wcXrS5yGtUoe-
wAKqoqL5JhIQ109s1FMNopL_50HR_t4",
    "expires_in": 259200,
    "token_type": "Bearer"
}
```

The JSON Web Token (JWT) / OAuth2 Bearer Token is later used by the Web client as follows:

```
GET /admin HTTP/1.1
Authorization: Bearer eyJhbGciOiJIU-zI1NiJ9....
```

After successful authentication, the developer can then interact with the marketplace, and e.g. register new BIG IoT Provider and Consumer instances (see Figure 4). This forms the basis for the next steps, where the Providers and Consumers have to authenticate on the marketplace independently of the developer/user.

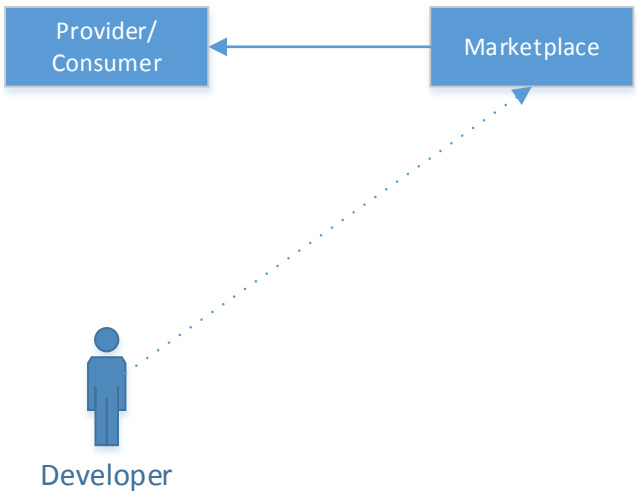

Figure 4 - User interacts with the Marketplace Web Portal (e.g. to create new Provider or Consumer instances)

5.2.3. Provider or Consumer Instance accesses a Marketplace (Mx interfaces)

When a run-time instance of a Provider or Consumer accesses the Marketplace to authenticate, register, and discover information via the Mx interfaces, the Consumer/Provider requests a JWT / Oauth2 Bearer token from Marketplace (based on ***Oauth2 Client Credentials Grant Type***), using ***grant type***: ***client_credentials*** which uses `client_id` and `client_secret`.

Upon successful authentication of a Provider or Consumer, the Marketplace will provide a JWT / Oauth2 Bearer Token in the response, which is used in all subsequent interactions on the Mx interfaces.

The use of `client_credentials` will look something like this:

```
https://market.big-iot.org/accessToken?clientId=<Provider-ID>&clientSecret=<Provider-Secret>
```

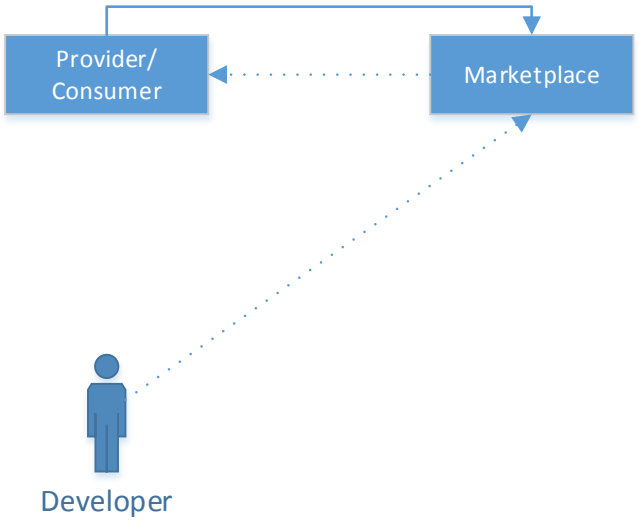

Figure 5 - Providers/Consumers authenticate at marketplace (Mx interfaces)

The overall flow for Providers/Consumers authenticating to the marketplace is illustrated in Figure 5.

5.2.4. Consumer Instance accesses a Provider Endpoint (A1 interface)

The communication between a Consumer and a Provider is a special case in that no delegation is involved. Therefore, it does not follow any of the OAuth flows, but requires that the previously performed OAuth flows and interactions with the marketplace provide appropriate credentials to the involved providers and consumers to enable secure communications.

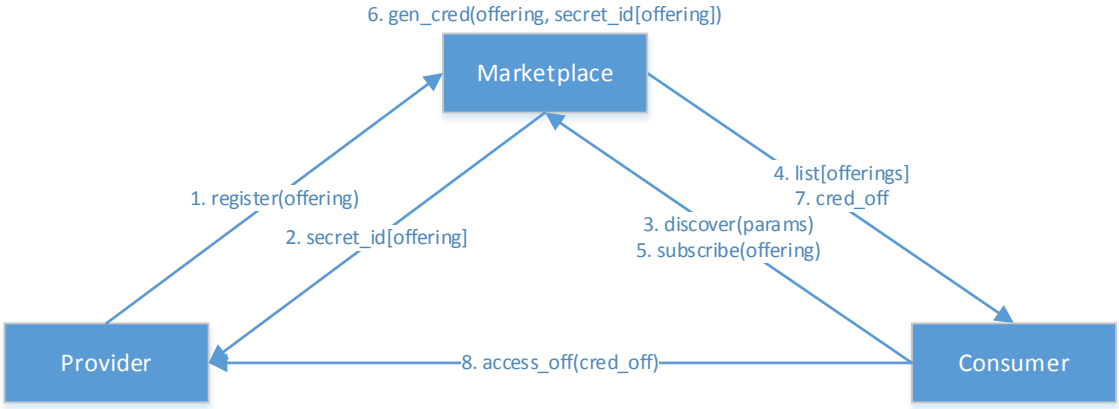

Figure 6 - Detailed flow of Provider/Consumer Access Control (Mx -> A1 interfaces)

As depicted in Figure 6, a run-time instance of a Consumer accesses the offered resources via the Offering Endpoint. During the subscription (step 5) of an Offering, the Consumer obtains the so-called *Offering Access Token* (step 7) from the Marketplace.

The JWT access token is signed by the Marketplace using the Provider's secret and used by the Consumer in the access request (step 8). The token is added in the access request (on the A1 interface) using, for example, the HTTP Authorization Header:

```
GET /bigiot/access/<endpoint> HTTP/1.1
Authorization: Bearer eyJhbGciOiJIU-zI1NiJ9...
```

Prior to processing the access request, the Provider validates the *Offering Access Token* and authorizes the Consumer request to the respective Offering according to the information in the token (i.e. Offering ID, Consumer ID, Subscription ID). Once the Offering Access Token reaches its expiration date, the Consumer re-subscribes to the Offering in order to obtain a new token.

5.3. ASVS analysis of the BIG IoT API of the Marketplace

In the following you will find a list of verification requirements based on OWASP's Application Security Verification Standard (ASVS) v3.0.1 [23]. The verifications are ordered as they appear in the original ASVS document and some the results of the tests have been reported to the BIG IoT API and Marketplace development teams to be address (if required) and thus incorporated to the development lifecycle.

Since this is an on-going work and the resources are limited, the target of this work is to at least address the items considered with high priority. For this release verification item sets V1 (Architecture, design and threat modelling), V2 (Authentication Verification Requirements), V3 (Session Management Verification Requirements), V10 (Communications security verification requirements) and V11 (HTTP security configuration verification requirements) have been addressed, since they are the one more suited to the specifics of BIG IoT. There is an internal Wiki page that is constantly updated with new tests and the following is just a snapshot of its current status.

If you are familiar with ASVS, with the latest version of ASVS there appear to be missing requirements and sections. Most of the numeration gaps between items are due to requirements merge, deprecation and move to other sections. This is explained in "What happened to…" section of the ASVS document.

For the sake of clarity, since we are not considering the use of ASVS level 3 (the highest security level), requirements that only apply to level 3 have been deliberately omitted.

For better understanding of the following tables, Table 5 details the meaning of the headers.

Team	Big IoT subteam responsible for the verification items, and contact person leading the team. By default, all checks are responsibility of the Security team; however, due to an inside code check or system design, some of them should be transferred to the teams developing the affected system parts.
Codable	A security black box test is considered doable.
Priority	High maps to ASVS level 1. Medium maps to ASVS level 2.
Status	Status of the review process: • ***Pending***: Task is described but work is still to be started. • ***Work in progress***: Team in charge is working on the task. • ***Need collaboration***: Work in progress but help requested to another team.

- ***Review***: A non-security team considers the task finished and asks security team for review.
- ***Fail:*** evaluation has failed and should be addressed.
- ***Pass***: evaluation has passed because risk doesn't apply, or because it's assessed correctly.
- ***Externalized***: this security measure is already checked by external parties.
- ***Won't fix***: evaluation has failed and risk is accepted definitely.

Processes of fixing the issues do not affect this code until they end, and ask for a review again. This means that all "Done" statuses are subject to changes on code, configuration, or deployment. The flow of statuses is detailed in Figure 7.

Table 5 - Explanation of ASVS table headers.

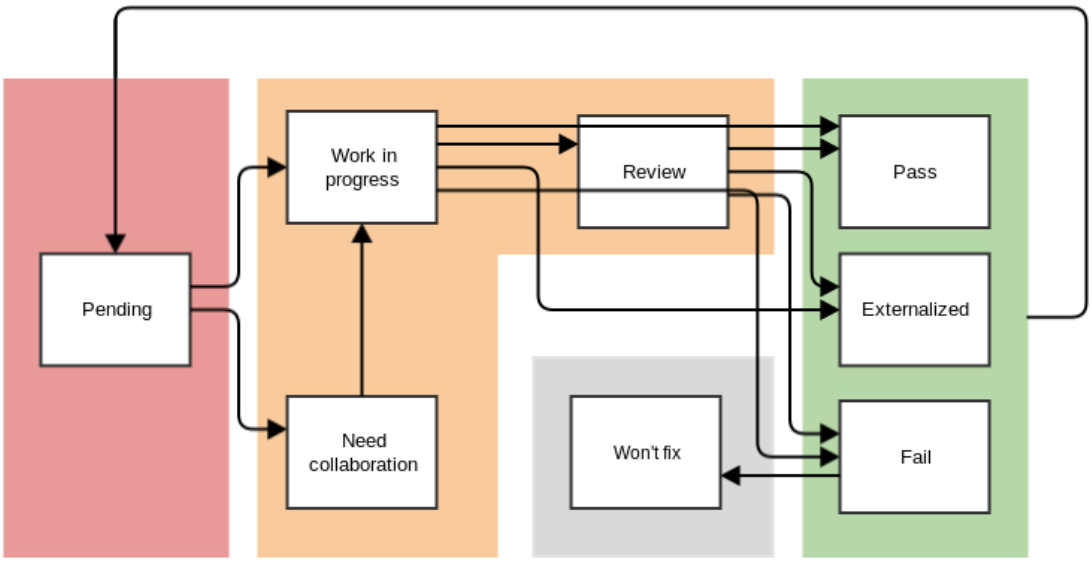

Figure 7 - Flow of statuses of ASVS requirement verifications

5.3.1. V1: Architecture, design and threat modelling

ID	Codable?	Priority	Status	Description

1.1	No	High	Pending	Verify that all application components are identified and are known to be needed.
1.2		Medium		Verify that all components, such as libraries, modules, and external systems, that are not part of the application but that the application relies on to operate are identified.
1.3	No	Medium	Pass	Verify that a high-level architecture for the application has been defined.
1.7		Medium		Verify all security controls (including libraries that call external security services) have a centralized implementation.
1.8		Medium		Verify that components are segregated from each other via a defined security control, such as network segmentation, firewall rules, or cloud based security groups.
1.9		Medium		Verify the application has a clear separation between the data layer, controller layer and the display layer, such that security decisions can be enforced on trusted systems.
1.10		Medium		Verify that there is no sensitive business logic, secret keys or other proprietary information in client side code.
1.11		Medium		Verify that all application components, libraries, modules, frameworks, platform, and operating systems are free from known vulnerabilities.

5.3.2. V2: Authentication Verification Requirements

ID	Codable?	Priority	Status	Description
2.1		High	need collaboration	Verify all pages and resources by default require authentication except those specifically intended to be public (Principle of complete mediation).
2.2		High	Pass	Verify that forms containing credentials are not filled in by the application. Pre-filling by the application implies that credentials are stored in plaintext or a reversible format, which is explicitly prohibited.
2.4		High	Pass	Verify all authentication controls are enforced on the server side.
2.6		High	Pending	Verify all authentication controls fail securely to ensure attackers cannot log in.
2.7	X	High	externalized	Verify password entry fields allow, or encourage, the use of passphrases, and do not prevent password managers, long passphrases or highly complex passwords being entered.
2.8		High	externalized	Verify all account identity authentication functions (such as update profile, forgot password, disabled / lost token, help desk or IVR) that might regain access to the account are at least as resistant to attack as the primary authentication mechanism.

2.9	X	High	external-ized	Verify that the changing password functionality includes the old password, the new password, and a password confirmation.
2.12		Medium		Verify that all authentication decisions can be logged, without storing sensitive session identifiers or passwords. This should include requests with relevant metadata needed for security investigations.
2.13		Medium		Verify that account passwords are one way hashed with a salt, and there is sufficient work factor to defeat brute force and password hash recovery attacks.
2.16		High	Pass	Verify that credentials are transported using a suitable encrypted link and that all pages/functions that require a user to enter credentials are done so using an encrypted link.
2.17		High	Pass	Verify that the forgotten password function and other recovery paths do not reveal the current password and that the new password is not sent in clear text to the user.
2.18		High	Won't fix	Verify that information enumeration is not possible via login, password reset, or forgot account functionality.
2.19		High	pending	Verify there are no default passwords in use for the application framework or any components used by the application (such as “admin/password”).

2.20		High		Verify that anti-automation is in place to prevent breached credential testing, brute forcing, and account lockout attacks.
2.21		Medium		Verify that all authentication credentials for accessing services external to the application are encrypted and stored in a protected location.
2.22		High	External-ized	Verify that forgotten password and other recovery paths use a TOTP or other soft token, mobile push, or other offline recovery mechanism. Use of a random value in an e-mail or SMS should be a last resort and is known weak.
2.23		Medium		Verify that account lockout is divided into soft and hard lock status, and these are not mutually exclusive. If an account is temporarily soft locked out due to a brute force attack, this should not reset the hard lock status.
2.24		High	Pass	Verify that if shared knowledge based questions (also known as "secret questions") are required, the questions do not violate privacy laws and are sufficiently strong to protect accounts from malicious recovery.
2.25		Medium		Verify that the system can be configured to disallow the use of a configurable number of previous passwords.
2.26		Medium		Verify that risk based re-authentication, two factor or transaction signing is in place for high value transactions.

2.27	X	High	external-ized	Verify that measures are in place to block the use of commonly chosen passwords and weak passphrases.
2.31		Medium		Verify that if an application allows users to authenticate, they can authenticate using two-factor authentication or other strong authentication, or any similar scheme that provides protection against username password disclosure.
2.32		High	Need collabo-ration	Verify that administrative interfaces are not accessible to untrusted parties.
2.33		High	Pass	Browser autocomplete, and integration with password managers are permitted unless prohibited by risk based policy.

5.3.3. V3: Session Management Verification Requirements

ID	Codable?	Priority	Status	Description
3.1		High	Work in progress	Verify that there is no custom session manager, or that the custom session manager is resistant against all common session management attacks.
3.2	X	High	Fail	Verify that sessions are invalidated when the user logs out.
3.3	X	High	Pass	Verify that sessions timeout after a specified period of inactivity.

3.4		Medium	Pass	Verify that sessions timeout after an administratively-configurable maximum time period regardless of activity (an absolute timeout).
3.5		High	Pass	Verify that all pages that require authentication have easy and visible access to logout functionality.
3.6		High	Pass	Verify that the session id is never disclosed in URLs, error messages, or logs. This includes verifying that the application does not support URL rewriting of session cookies.
3.7	X	High	Pass	Verify that all successful authentication and re-authentication generates a new session and session id.
3.10		Medium		Verify that only session ids generated by the application framework are recognized as active by the application.
3.11		High	Pass	Verify that session ids are sufficiently long, random and unique across the correct active session base.
3.12		High	Pass	Verify that session ids stored in cookies have their path set to an appropriately restrictive value for the application, and authentication session tokens additionally set the “HttpOnly” and “secure” attributes
3.16	X	High	Need collaboration	Verify that the application limits the number of active concurrent sessions.

3.17		High	Fail	Verify that an active session list is displayed in the account profile or similar of each user. The user should be able to terminate any active session.
3.18		High	Won't fix	Verify the user is prompted with the option to terminate all other active sessions after a successful change password process.

5.3.4. V10: Communications security verification requirements

ID	Codable?	Priority	Status	Description
10.1		High	Pass	Verify that a path can be built from a trusted CA to each Transport Layer Security (TLS) server certificate, and that each server certificate is valid.
10.3		High	Pass	Verify that TLS is used for all connections (including both external and backend connections) that are authenticated or that involve sensitive data or functions, and does not fall back to insecure or unencrypted protocols. Ensure the strongest alternative is the preferred algorithm.
10.6		Medium		Verify that all connections to external systems that involve sensitive information or functions are authenticated.
10.10		Medium		Verify that TLS certificate public key pinning (HPKP) is implemented with production and backup public keys. For more information, please see the references below.

10.11		High	Fail	Verify that HTTP Strict Transport Security headers are included on all requests and for all subdomains, such as Strict-Transport-Security: max-age=15724800; includeSubdomains
10.13		High	Pass	Ensure forward secrecy ciphers are in use to mitigate passive attackers recording traffic.
10.14		High	Fail	Verify that proper certification revocation, such as Online Certificate Status Protocol (OCSP) Stapling, is enabled and configured.
10.15		High	Fail	Verify that only strong algorithms, ciphers, and protocols are used, through all the certificate hierarchy, including root and intermediary certificates of your selected certifying authority.
10.16		High	Pass	Verify that the TLS settings are in line with current leading practice, particularly as common configurations, ciphers, and algorithms become insecure.

5.3.5. V11: HTTP security configuration verification requirements

ID	Codable?	Priority	Status	Description
11.1		High	Pass	Verify that the application accepts only a defined set of required HTTP request methods, such as GET and POST are accepted, and unused methods (e.g. TRACE, PUT, and DELETE) are explicitly blocked.

11.2		High	Pass	Verify that every HTTP response contains a content type header specifying a safe character set (e.g., UTF-8, ISO 8859-1).
11.3		Medium		Verify that HTTP headers added by a trusted proxy or SSO devices, such as a bearer token, are authenticated by the application.
11.4		Medium		Verify that a suitable X-FRAME-OPTIONS header is in use for sites where content should not be viewed in a 3rd-party X-Frame.
11.5		High		Verify that the HTTP headers or any part of the HTTP response do not expose detailed version information of system components.
11.6		High		Verify that all API responses contain X-Content-Type-Options: nosniff and Content-Disposition: attachment; filename="api.json" (or other appropriate filename for the content type).
11.7		High		Verify that a content security policy (CSPv2) is in place that helps mitigate common DOM, XSS, JSON, and JavaScript injection vulnerabilities.
11.8		High		Verify that the X-XSS-Protection: 1; mode=block header is in place to enable browser reflected XSS filters.

Detailed updated reports of every specific test can be found in our internal collaboration Wiki. While we have decided not to add such an amount of information, details can be revealed upon specific requests.

6. Security and privacy analysis of the BIG IoT pilots

The BIG IoT use cases serve the purpose of demonstrating how the developed technological solution can achieve the BIG IoT overall goal of establishing syntactic and semantic interoperability for smart object platforms. They are aimed at showing how a service or application provider can offer their assets on top of different smart object platforms through the achieved interoperability. With such a purpose, three large-scale regional pilots (Northern Germany – NG, Barcelona – BCN and Piedmont – PIE). The different use cases addressed in the three pilots have been classified into eight use case clusters that are detailed in Deliverable 2.2.

Use Case Cluster	Pilot		
	NG	**BCN**	**PIE**
Smart Parking			
Smart Traffic Management			
Public Transport Optimization			
Healthy Bike Navigation			
Smart Bike Sharing			
Incentive-based Green Route Planning			
Multi-modal Route Optimizer			
Smart Charging			
Blue highlighting indicates relevant use case clusters for the specific pilots.			

Table 6 - Use case clusters.

It is important to note that from the end user point of view, in a typical use case, all the functionalities are accessed with a single BIG IoT application; however, behind the scenes this app is consuming data from several services, which in turn consume data provided by different platforms.

Services are an important abstraction layer in BIG IoT because they allow code re-utilization and simplify the process of building BIG IoT applications. For example, a service can aggregate data acquired from different platforms and then, present a unified dataset to apps. Another service could manage the history of data, allowing the user to access to past data (data not currently available in the source platform). Another service could create forecasts from data (acquiring data from a platform or from another service). Finally, another interesting use could be a service that anonymises an underlying dataset. This could allow lower levels of ASVS security in the upper layer (app).

With BIG IoT use case clusters shared between the three pilots, we conduct a security and privacy analysis of the different platforms and services involved regardless of the specific pilot.

Note: The following subsections do not address a complete list of all the services and platforms that are already in development or planned. The analysis is an on-going process that is about to end in month 30 and the final list could also change. Consequently, to date only a subset of the services/platforms has been addressed. For every analysed service or platform an overview of such an analysis is presented; more details about the performed risk assessment can be found in "ANNEX A. Risk Assessment Template" and the vulnerability assessments in "ANNEX B. Vulnerability Analyses of BIG IoT services".

The risk assessment has been conducted following the methodology in Section 4. After some tuning and discussion (which will continue during the next months), the array of weights agreed for the risk assessment of the pilots is detailed in Table 6. Some of the factors have weights less than one because they have been considered to be less relevant for the BIG IoT pilot scenarios.

	Factor	Weight
Likelihood	Required Skill level	1
	Motive	0.5
	Opportunity / Resources	1
	Size	1
	Ease of discovery	0.25
	Ease of exploit	0.25
	Awareness	0.25
	Intrusion detection	1
Tech-nical impact	Loss of privacy	1
	Loss of integrity	0

	Loss of availability	0
	Loss of accountability	0.5
Business impact	Financial damage	0.5
	Reputation damage	0.5
	Privacy violation	1

Table 7 - Array of weights for the BIG IoT risk assessment.

6.1. BIG IoT Pilot Services

Services are an important abstraction layer in BIG IoT because they allow code re-utilization and simplify the process of building BIG IoT applications. For example, a service can aggregate data acquired from different platforms and then, present a unified dataset to apps. Another service could manage the history of data, allowing the user to access to past data (data not currently available in the source platform). Another service could create forecasts from data (acquiring data from a platform or from another service). Finally, another interesting use could be a service that anonymises an underlying dataset. This could allow lower levels of ASVS security in the upper layer (app).

A list of all the services analysed that are currently running on the BIG IoT is presented in the following. Notice that this is not a complete list with all the available services. The analysis is an on-going process that is about to end in month 30.

6.1.1. AirQualityMonitoringService

Related Use Case Clusters: HBN, STM, GRP

Pilots involved: BCN, PIE

Connected platforms: Bezirk (SEAT cars)

Recommended ASVS level (with comments if needed): 2

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- "Air Quality Sensor Information" at https://ec2-52-57-230-235.eu-central-1.compute.amazonaws.com:9999/bigiot/access/airqualitydata.

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Vehicle tracking	CRITICAL	CRITICAL	Cars are sharing their position along with the measured pollution in this spot. Data should be anonymized. Use of identifiers or pseudo-identifiers is discouraged.
Access/hacking of the in-vehicle information bus.	MEDIUM	MEDIUM	A Bezirk "thing" is connected to the in-vehicle information bus to retrieve pollution data. If an attacker gains access to the Bezirk device, it could also try to hack the internal bus. Input data to the Bezirk device should be avoided and, if not, properly parametrized. Bus access should be only allowed to collect the necessary data (pollution).
Altering noise data over a given area or a given set of areas.	LOW	LOW	Tampering noise data could allow and attacker to artificially create noisy areas (by reporting no noise at all) or extremely peaceful (by reporting

			a high level of noise) in detriment of other areas.

6.1.2. Alternative Transport Information Service

Related Use Case Clusters: STM

Pilots involved: BCN

C**onnected platforms**: Transport Metropolitans de Barcelona (non BIG IoT)

Recommended ASVS level (with comments if needed): 2

Vulnerability assessment: the service is planned to be online for month 30 (second iteration). The analysis is therefore delayed.

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Bad implementation of the connection to the TMB API could lead to abuse of use by other stakeholders.	LOW	LOW	API credentials should be securely stored, securely managed and not exposed during connection.

The Alternative Transport Information Service in the Barcelona Pilot is currently only fed with data from Transports Metropolitans de Barcelona (TMB), withis the main public transport operator in the Barcelona metropolitan area. TMB already has a restricted API [25] where to obtain routes to destinations with different public transports: trains, metro, buses. The service gets the TMB API into the BIG IoT ecosystem. Since all the data involved in the service are public, no specific requirements in terms of privacy are required. Regarding security, unless properly justified, the software development has to comply with standard ASVS level 2.

6.1.3. Bikes Availability Service

Related Use Case Clusters: SBS, STM, GRP

Pilots involved: PIE, BCN, NG

Connected platforms: CSI, VMZ, Barcelona City Hall Open Data

Recommended ASVS level (with comments if needed): 2

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- NG Pilot: "BikesharingAvailabilityServiceProvider_offering" at https://bigiot.provider.vmz.services:9005/bigiot/access/availablebikesharinginfo.
- BCN Pilot: "BikesOffering" at https://gibo.fib.upc.edu:50008/bigiot/access/bikes.
- PIE Pilot: analysis of PIE offerings is scheduled for month 30.

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Service disruption – fake information	MEDIUM	MEDIUM	If an attacker manage to modify or create fake information about stations, the service will be useless for the user that will leave it out soon.

6.1.4. Charging Station Availability Service

Related Use Case Clusters: SC

Pilots involved: NG, BCN

Connected platforms: VMZ, Ecove

Recommended ASVS level (with comments if needed): 1

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- NG Pilot: "EVChargingAvailabilityServiceProvider_offering" at https://bigiot.provider.vmz.services:9003.
- BCN Pilot: "Charging Points Offering" at https://gibo.fib.upc.edu:50009/bigiot/access/chargePoints

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Service disruption – fake information	MEDIUM	MEDIUM	If an attacker manage to modify or create fake information about stations, the service will be useless for the user that will leave it out soon.

6.1.5. Live Bus Location Service

Related Use Case Clusters: PTO, SP, SBS

Pilots involved: NG, BCN, PIE

Connected platforms: Bosch Smart City Platform

Recommended ASVS level (with comments if needed): 1

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- NG & BCN Pilots: "Live Bus Location" at https://bigiot.lab.es.aau.dk:19063/bigiot/access/buslocation.
- PIE Pilot: analysis of PIE offerings is scheduled for month 30.

Risk Assessment

Risk/Attack/Vulnerability	Technical risk severity	Business risk severity	Notes / actions
Breach of confidentiality of bus schedule information - i.e. bus schedule information is extracted from services without bus company consent	HIGH	LOW	Secure bus schedule information with access control and update authentication steps

The listed risk can relatively easy me handled by securing the access of this information and also logging all access requests.

6.1.6. Parking Availability Service

Related Use Case Clusters: SP

Pilots involved: NG, BCN, PIE

Connected platforms: VMZ, WorldSensing's FastPrk, CSI

Recommended ASVS level (with comments if needed):

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- NG Pilot: "ParkingAvailabilityServiceProvider_offering" at https://bigiot.provider.vmz.services:9001/bigiot/access/availableparkinginfo.
- BCN Pilot: "Les Corts Parking All Offering" at https://gibo.fib.upc.edu:50014/bigiot/access/lescorts-parking-all.
- PIE Pilot: analysis of PIE offerings is scheduled for month 30.

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**

Monitoring vehicles/user activities by requesting specific status of monitored spots.	HIGH	MEDIUM	If an attacker knows where someone has parked its car, it can monitor when he/she leaves by checking the spot status. If data provided by the platforms refer to individual parking spots, (k-)anonymity measures should be taken into account before providing this information.
Accessing service stored data to analyse vehicle activities.	MEDIUM	MEDIUM	if an attacker knows where and when someone parked its car, it can monitor when he/she leaves by checking the stored spot statuses. Stored data should be protected at least with ASVS level 2.

This service is fed with parking spot status provided by the connected platforms. In addition, the service provides and may store data about parking spot status. Therefore, the same privacy recommendations as for the platforms applies here. Both the service and their connections must be protected with ASVS level 2.

6.1.7. People Density Estimation on Bus Service / People Density Estimation in Area Service

Related Use Case Clusters: PTO, STM

Pilots involved: NG, BCN

Connected platforms: Bosch Smart City Platform

Recommended ASVS level (with comments if needed): 1 or 2 based on the data collected being sufficiently anonymized before storage.

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- NG & BCN Pilots:
  - "Live People Count In Area" at https://bigiot.lab.es.aau.dk:19063/bigiot/access/buslocation.
  - "Live Bus Occupancy" at https://bigiot.lab.es.aau.dk:19061/bigiot/access/busoccupancy.

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Eavesdropping on the deanonymized WLAN probing data - Physical access to the sensor node to install malicious SW code to send deanonymized WLAN probes to man in the middle	Medium	Medium	Password protect and encrypt internal storage of sensor, and place sensor in a locked compartment, and log access requests
Eavesdropping on the deanonymized WLAN probing data - Sensors are remotely hacked and wlan probe data is sent before it is anonymized	High	Medium	Close ports for remote access, and log access requests

Eavesdropping on anonymized WLAN probing data - eavesdropping on the cellular sensor node communication	High	Medium	Log access requests
Eavesdropping on anonymized WLAN probing data - accessing probing information on the service execution environment	High	Medium	Ensure that secure access control is in place and log access requests
Eavesdropping on anonymized WLAN probing data - writing a malicious application (or service) that access WLAN probing information directly (instead of bus occupancy)	High	Medium	Implement secure access control and log access requests
Usage pattern of application by a user is tracked by attacker; e.g. the application execution environment is compromised and the attacker installs malicious	Medium	Medium	Update application regularly, implement secure access control and log access requests

It can be seen that most of the risks can be significantly minimized by reducing remote access possibilities to the sensors and the servers hosting the services and apps. Furthermore, in case of breach a logging system would significantly improve the chances of the breach being discovered, and allowing for actions to be taken.

6.1.8. Traffic Monitoring Service

Related Use Case Clusters: STM, HBN, GRP

Pilots involved: BCN, PIE

Connected platforms: OpenIoT, Worldsensing's Bitcarrier, TBD

Recommended ASVS level (with comments if needed): 2.

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- BCN Pilot:
  - "RondesTravelTimeAll" at https://gibo.fib.upc.edu:50013/bigiot/access/rondestraveltime-all.
  - "RondesVectorSpeedAll" at https://gibo.fib.upc.edu:50013/bigiot/access/rondes-vector-speed-all.
  - "RondesVectorTrafficStatusAll" at https://gibo.fib.upc.edu:50013/bigiot/access/rondes-vector-trafficstatus-all.

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Eavesdrop of service data.	CRITICAL	MEDIUM	An attacker could track vehicles/users if they are identified No data about individual vehicles should be shared. General indicators about traffic conditions on a given link should be used instead
Traffic disruption	LOW	MEDIUM	An attacker able to mangle routing responses could be able somehow manage the traffic: making the traffic heavier in a zone or lighter in another.

Access to the information provided by the platforms to the service would allow an attacker to track vehicle/users	CRITICAL	HIGH	Data provided by the different platforms should be anonymized. If it is not the case, no identifiable data should be stored by the service and the connection between service and platform must be protected.

This service is providing traffic status on specific requested links or segments. Assuming that data provided by the platforms is related to identifiable vehicles but to general status, no specific requirements in terms of privacy are required.

6.1.9. Traffic Recommendations Management Service

Related Use Case Clusters: STM, GRP

Pilots involved: BCN

Connected platforms: OpenIoT

Recommended ASVS level (with comments if needed):

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- BCN Pilot:
  - "Set Recommendation Offering" at https://gibo.fib.upc.edu:50015/bigiot/access/set-recommendation.

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Denial of Service	LOW	LOW	If an attacker or a group of attackers floods the system with recommendations, the

			system provided data could become useless. Implementation of flooding protection is encouraged. Reputation services can also help in filtering the recommendations.

6.2. BIG IoT Pilot Platforms

In the following we analyse the BIG IoT platforms involved in the pilots. Only the BIG IoT interfaces to those platforms are under analysis, since analysing existing platforms is out of the scope of this project.

6.2.1. BOSCH Bezirk

Related Use Case Clusters: HBN, STM, GRP

Pilots involved: BCN, PIE

Recommended ASVS level (with comments if needed): 2

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- "Air Quality Sensor Information" at https://ec2-52-57-230-235.eu-central-1.compute.amazonaws.com:9999/bigiot/access/airqualitydata

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Vehicle tracking	CRITICAL	CRITICAL	Cars are sharing their position along with the measured pollution in this spot.

			Data should be anonymized. Use of identifiers or pseudo-identifiers is discouraged
Access/hacking of the in-vehicle information bus.	MEDIUM	MEDIUM	A Bezirk "thing" is connected to the in-vehicle information bus to retrieve pollution data. If an attacker gains access to the Bezirk device, it could also try to hack the internal bus. Input data to the Bezirk device should be avoided and, if not, properly parametrized. Bus access should be only allowed to collect the necessary data (pollution).
Altering noise data over a given area or a given set of areas.	LOW	LOW	Tampering noise data could allow and attacker to artificially create noisy areas (by reporting no noise at all) or extremely peaceful (by reporting a high level of noise) in detriment of other areas.

6.2.2. Worldsensing's Bitcarrier (WiFi/Bluetooth antennas)

Related Use Case Clusters: STM

Pilots involved: BCN

Connected services: Traffic Monitoring Service

Recommended ASVS level (with comments if needed): 2 and 3 (see details below)

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- BCN Pilot:
  - "RondesTravelTimeAll" at https://gibo.fib.upc.edu:50013/bigiot/access/rondestraveltime-all.
  - "RondesVectorSpeedAll" at https://gibo.fib.upc.edu:50013/bigiot/access/rondes-vector-speed-all.
  - "RondesVectorTrafficStatusAll" at https://gibo.fib.upc.edu:50013/bigiot/access/rondes-vector-trafficstatus-all.

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Vehicle tracking. Eavesdrop of service data.	MEDIUM	MEDIUM	An attacker could track vehicles/users if they are identified No data about individual vehicles should be shared. General indicators about traffic conditions on a given link should be used instead
Vehicles/users could be tracked or profiled based on the stored data	CRITICAL	HIGH	Stored data must be appropriately secured/anonymised to avoid any kind of leakage (mistakenly or on purpose)

Traffic disruption	LOW	MEDIUM	An attacker able to mangle routing response could be able somehow manage the traffic: making the traffic heavier in a zone or lighter in another.
Access to the information provided by the platforms to the service would allow an attacker to track vehicle/users	CRITICAL	HIGH	Data provided by the different platforms should be anonymized. If it is not the case, no identifiable data should be stored by the service and the connection between service and platform must be protected.

This platform is fed with data gathered by WiFi/Bluetooth antennas placed at several street crosses. The technology detects vehicles/users by their unique MAC addresses and provides average travel times, speed, and even street congestion. MAC addresses are very sensitive data, which could be used to track or profile vehicles' (and even users') habits. This potential privacy invasion should be avoided. To do so, all the parties involved have to agree the necessary legal contracts in which they accept to use the data stored/exchanged properly. In addition, these data must be appropriately secured/anonymised to avoid any kind of leakage (mistakenly or on purpose). Under this assumption, BIG IoT approach is to anonymise immediately unique addresses using a one-way cryptographic-hash function. This function uses as inputs: (1) the address to be anonymised and (2) a key. This key is updated periodically, e.g. ranging from minutes to days. In this manner, one device cannot be tracked for more than a period. How often the key is updated is part of the privacy policy.

The use of cryptographic hash functions allows anonymising the data while keeping a trapdoor that could be used to re-identify vehicles/users. The platform operator must keep secret the temporal keys used to anonymise the identifiers.

However, operators may be forced to disclose these keys under some circumstances, e.g. a law enforcement requirement when a legal process is followed. From the above reasoning, this platform should comply at least with recommended base-line ASVS level 2 "standard". In addition, the management of the anonymization keys should comply with ASVS level 3 "advanced", as it may allow an attacker to identify/track users and/or vehicles.

6.2.3. Worldsensing's Fastprk (on-street parking spot status)

Related Use Case Clusters: SP

Pilots involved: BCN

Connected services: Parking Availability Service

Recommended ASVS level (with comments if needed): 1 o 2 (see details below)

Vulnerability assessment: Annex B includes reports for associated BIG IoT offerings:

- BCN Pilot: "Les Corts Parking All Offering" at https://gibo.fib.upc.edu:50014/bigiot/access/lescorts-parking-all

Risk Assessment			
Risk/Attack/Vulnerability	**Technical risk severity**	**Business risk severity**	**Notes / actions**
Monitoring vehicles/user activities by requesting specific status of monitored spots.	HIGH	MEDIUM	if an attacker knows where someone has parked their car, it can monitor when he/she leaves by checking the spot status. Before sending any data, (k-)anonymity measures should be taken into account.

This platform can provide individual status of parking spots over a predefined monitored area. For instance, for the BIG IoT Barcelona use cases, this platform currently offers status information for 600 on-street parking spots. This kind of data entails specific privacy risks due to correlation with other sources: if an attacker knows where someone has parked their car, it can monitor when he/she leaves by checking the spot status. Obviously, a straightforward countermeasure would be, e.g., to provide free spots in a given street segment (a virtual lot). This approach will guarantee k-anonymity (a given individual cannot be differentiated from another k = 1) of monitored vehicles/users with k being the number of vehicles parked on the same segment. The greater the segments are, the more anonymous the service is, but it will provide less specific, potentially less useful, information. Intuitively, it seems that obtaining the exact free parking spot position or the segment where there is one (or more) free parking spots is likely to be equally useful for the end user; although looking for the appropriate trade-off between privacy and usability requires further technical discussion and studies of real users' needs.

While some applications/services may allow different per-user degrees of privacy, this is not the case for this scenario. Therefore, testing user feedback about the suitability (or not) of just providing free spots on the street without their specific location cannot be done on an individual basis; it should be a global approach with, e.g., a pilot project.

Since the data stored by the platform can be somehow used to track/monitor end users, we recommend securing this BIG IoT platform following ASVS verification level 2. However, if the platform just stores and provides free parking spots in a pre-defined segment/lot, ASVS verification level 1 could be considered.

7. Conclusions & Outlook

Nowadays, a plethora of IoT platforms and solutions exist, but yet no large-scale and cross-platform IoT ecosystems have been developed. This is mainly due to the fragmentation of IoT platforms and interfaces, as this variety results in high market entry barriers. The BIG IoT project aims at establishing interoperability across platforms in order to ignite an IoT ecosystem. Core technological pillars of BIG IoT are a common API as well as a marketplace for all participants of the IoT ecosystem, including devices, end-users, and service providers. Key to the success of BIG IoT is to achieve appropriate levels of security and privacy.

In this deliverable, we have identified seven requirements to be followed when creating and/or deploying BIG IoT components. Such requirements affect the design of the BIG IoT API and the marketplace, as well as any software in the BIG IoT ecosystem. Following this analysis, we have outlined how these requirements will affect the architectural approach of BIG IoT.

We have also proposed three recommendations regarding privacy that need to be followed by any IoT ecosystem participant: 1) data minimisation, i.e., that a data controller should limit the collection of personal information to what is directly relevant and necessary to accomplish a specified purpose; 2) strong accountability, i.e., to provide mechanisms to securely log any action by any actor dealing with sensitive data; and 3) transparency and easy access, i.e., any data controller should publish transparent and easily accessible data protection policies that clearly show how their data is being processed to the end users.

Notice that protecting users' privacy does not necessarily imply added costs. In fact, storing anonymised data can help in saving development and operational costs due to a reduced security-level compliance.

Besides general recommendation and advice, in this deliverable we have presented an analysis of the authentication/authorization flows of the BIG IoT API, a description of how the ASVS approach is been applied to the BIG IoT API, a specific risk rating methodology for the BIG IoT actors and its use to perform a risk assessment of the services currently running on the BIG IoT pilots and finally a vulnerability analysis of the endpoints of the pilot services.

Part of the work carried so far has been published as a conference paper in [24].

In the future, we will continue collaborating to guarantee privacy and security in the implementation and deployment of the various services and applications in the pilots of the BIG IoT

project, which all need to follow the security and privacy framework outlined here. This will lead to sharpened and proven guidelines for the creation of IoT ecosystems in general, which we aim to contribute to our on-going engagement with standardization at W3C's Web of Things group [26].

In the future, we will focus our research agenda towards combining IoT security solutions with Semantic Web [27] technologies. The already available semantic descriptions of services and platforms in the BIG IoT project will enable us to develop ontologies that describe different security aspects. This will allow us to automate the selection of reasonable security measures and options per IoT ecosystem participant.

References

[1] A. Bröring, S. Schmid, C.-K. Schindhelm, A. Khelil, S. Käbisch, D. Kramer, D. Le Phuoc, J. Mitic, D. Anicic and E. Teniente, "Enabling IoT ecosystems through platform interoperability," *IEEE Software*, vol. 34, no. 1, pp. 54-61, 2017.

[2] OWASP Internet of Things project, "Principles of IoT Security," [Online]. Available: https://www.owasp.org/index.php/Principles_of_IoT_Security. [Accessed 18 07 2017].

[3] Allseen Alliance. Alljoyn Framework, "Linux Foundation Collaborative Projects," [Online]. Available: https://allseenalliance.org/framework. [Accessed 18 07 2017].

[4] M. Jones and J. Hildebrand, "JSON Web Encryption (JWE)," 05 2015. [Online]. Available: https://tools.ietf.org/html/rfc7516. [Accessed 20 07 2017].

[5] M. Jones, J. Bradley and N. Sakimura, "JSON Web Signature (JWS)," 2015 05 2015. [Online]. Available: https://tools.ietf.org/html/rfc7515. [Accessed 20 07 2017].

[6] T. Imamura, B. Dillaway and E. Simon, "XML Encryption Syntax and Processing," 10 12 2002. [Online]. Available: http://www.w3.org/TR/xmlenc-core/. [Accessed 20 07 2017].

[7] M. Bartel, J. Boyer, B. Fox, B. LaMacchia and E. Simon, "XML Signature Syntax and Processing (Second Edition)," 10 06 2008. [Online]. Available: https://www.w3.org/TR/xmldsig-core/. [Accessed 18 07 2017].

[8] Organization for the Advancement of Structured Information Standards (OASIS), "Official Wiki of the OASIS Security Services (SAML) Technical Committee," [Online]. Available: https://wiki.oasis-open.org/security/FrontPage. [Accessed 18 07 2017].

[9] IETF OAuth Working Group, "OAuth 1," [Online]. Available: https://oauth.net/1/. [Accessed 18 07 2017].

[10] IETF OAuth Working Group, "OAuth 2.0," [Online]. Available: https://oauth.net/2/. [Accessed 18 07 2017].

[11] OpenID Foundation, "OpenID Connect," [Online]. Available: http://openid.net/connect/. [Accessed 18 07 2017].

[12] "The Open Web Application Security Project (OWASP)," [Online]. Available: https://www.owasp.org/. [Accessed 20 07 2017].

[13] OWASP, "Software Assurance Maturity Model (SAMM)," [Online]. Available: https://www.owasp.org/index.php/OWASP_SAMM_Project. [Accessed 18 07 2017].

[14] OWASP, "Application Security Verification Standard (ASVS) Project," [Online]. Available: https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project. [Accessed 20 07 2017].

[15] L. Conklin, G. Robinson, J. Curiel, E. Keary, I. A. Mennouchi, A. Naderi, C. Pantelides and M. Hidalgo, "Code Review Guide 2.0," 14 07 2017. [Online]. Available: https://www.owasp.org/index.php/File:OWASP_Code_Review_Guide_v2.pdf. [Accessed 12 09 2017].

[16] OWASP Testing Project, "OWASP Testing Guide v4," 17 09 2014. [Online]. Available: https://www.owasp.org/images/1/19/OTGv4.pdf. [Accessed 20 07 2017].

[17] FTC Staff, "Internet of Things. Privacy & Security in a Connected World," 01 2015. [Online]. Available: https://www.ftc.gov/system/files/documents/reports/federal-trade-commission-staff-report-november-2013-workshop-entitled-internet-things-privacy/150127iotrpt.pdf. [Accessed 20 07 2017].

[18] The European Parliament and of the Council of the European Union, "Directive 95/46/EC on the protection of individuals with regard to the processing of personal data and on the free movement of such data," 24 10 1995. [Online]. Available: http://data.europa.eu/eli/dir/1995/46/oj. [Accessed 20 07 2017].

[19] A. Raskin, "Privacy Icons," [Online]. Available: https://www.flickr.com/photos/azaraskin/5304502420/sizes/o/. [Accessed 20 07 2017].

[20] OWASP, "Risk Rating Methodology," 17 09 2014. [Online]. Available: https://www.owasp.org/index.php/OWASP_Risk_Rating_Methodology. [Accessed 20 07 2017].

[21] R. S. Ross, "Guide for Conducting Risk Assessments," 17 09 2012. [Online]. Available: https://www.nist.gov/publications/guide-conducting-risk-assessments. [Accessed 12 09 2017].

[22] R. Caralli, J. Stevens, L. Young and W. Wilson, "Introducing OCTAVE Allegro: Improving the Information Security Risk Assessment Process," 2007.

[23] Greenbone, "OpenVAS - Open Vulnerability Assessment System," [Online]. Available: http://www.openvas.org/. [Accessed 12 09 2017].

[24] OWASP, "Application Security Verification Standard 3.0.1," 07 2016. [Online]. Available: https://www.owasp.org/images/3/33/OWASP_Application_Security_Verification_Standard_3.0.1.pdf. [Accessed 20 07 2017].

[25] Transports Metropolitans de Barcelona, "TMB Open Data," [Online]. Available: https://www.tmb.cat/en/web/tmb/about-tmb/open-data. [Accessed 18 07 2017].

[26] J. Hernández-Serrano, J. L. Muñoz, A. Bröring, Ó. Esparza, L. Mikkelsen, W. Schwarzott, O. León and J. Zibuschka, "On the Road to Secure and Privacy-Preserving IoT Ecosystems," in *Interoperability and Open-Source Solutions for the Internet of Things: Second International Workshop, InterOSS-IoT 2016, Held in Conjunction with IoT 2016.*, Stuttgart (Germany), 2016.

[27] W3C, "Web of Things," [Online]. Available: http://www.w3.org/WoT/. [Accessed 18 07 2017].

[28] N. Shadbolt, T. Berners-Lee and W. Hall, "The Semantic Web Revisited," *IEEE Intelligent Systems*, vol. 21, no. 3, pp. 96-101, 2006.

[29] OWASP, «Zed Attack Proxy (ZAP) Project,» [En línea]. Available: https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project. [Último acceso: 12 09 2017].

ANNEX A. Risk Assessment Template

This is Annex A of Deliverable 3.3b - Security and Privacy Design for Smart Objects. This Annex Bs an excel spreadsheet to help testers with the analysis of risks with the BIG IoT risk rating methodology.

The risk assessment has been conducted following the methodology in Section 4. The array of weights initially set on the template is is detailed in Table 6, but obviously could be tuned and recomputation of risks will be automatic. Remember that there is quite a lot of uncertainty in these estimates and that these factors are intended to help the tester arrive at a sensible result.

The template can be downloaded from http://big-iot.eu/download/d3-3b-annex-a-riskassessmenttemplate/#.

ANNEX B. Vulnerability Analyses of BIG IoT services

This is Annex B of Deliverable 3.3b - Security and Privacy Design for Smart Objects. In this document we present reports of vulnerability analyses performed against the severs/cloud hosting the BIG IoT services and the HTTP endpoints of the service offerings.

The servers/cloud have been carried out using the well-known, widely-used, open-source Open Vulnerability Assessment System OpenVAS [23].

The HTTP endpoints have been analysed by means of other well-known, widely-used, open-source tool offered by the OWASP foundation and called ZED Attack Proxy – ZAP [29].

All the reports can be accessed here: http://big-iot.eu/download/d3-3b-annex-b-vulnerability-analyses-of-big-iot-services/#.