Towards a Trustworthy Service Marketplace for the Future Internet

Francesco Di Cerbo*, Michele Bezzi, Samuel Paul Kaluvuri, Antonino Sabetta, Slim Trabelsi, and Volkmar Lotz

SAP Research
805, Av. du Docteur Maurice Donat
06250 Mougins, France
francesco.di.cerbo@sap.com

Abstract. Digital economy is moving towards offering advanced business services, integrated into different applications and consumed from heterogeneous devices. Considering the success of actual software marketplaces, it is possible to foresee that *Service Marketplaces* (SM) will play a key role for the future Internet of Services. At present, on all offered software, marketplace operators define requirements that are common, and are validated before admitting them. However, the requirements, the validation process, and its results are not completely evident to the service consumers, resulting in a significant shortcoming especially with respect to security characteristics. In addition, having common security requirements for all services and applications makes the validation possibly inadequate to address the specific requirements that consumers may have.

In order to address these points, we propose the concept of a *trustworthy service marketplace* for the upcoming Internet of Services, where the security characteristics of services are certified and treated as first-class entities, represented in a machine-processable format. This allows service consumers – either human end-users or computer agents – to reason about these security features and to match them with their specific security requirements.

Keywords: Security, Trustworthiness, Trust, Service Marketplace.

1 Introduction

The marketplace metaphor is increasingly pervasive in today's digital economy. A software marketplace is a virtual place, where software providers can advertise their "apps" or services, and customers can browse and buy them; software marketplaces offer a centralized application distribution mechanism that reaches immediately many potential customers, all over the world. Marketplaces dedicated to specific devices or operating environments are nowadays proliferating and they represent a valuable business opportunity for software vendors. In many

* Corresponding author.

F. Álvarez et al. (Eds.): FIA 2012, LNCS 7281, pp. 105–116, 2012.

cases, like for the Apple Store[4], Windows Marketplace[17], or the Amazon Kindle Store[1], they are evolving to become gateways to entire ecosystems, with a potential audience of millions.

Similarly to apps, services can leverage on the marketplace distribution channel. Services relieve consumers from the burden of acquiring and managing their own operational infrastructure, on top of the benefits of component-based software [24]. Nowadays, following the SaaS (Software-as-a-Service) model, services are more and more commonly consumed as "commoditized" pieces of functionality, and are extensively adopted as a means to increase flexibility and to optimise IT expenditure.

The very nature of the model aims at simplifying software consumption by insulating the consumers from the complexity related to deployment, operation, and management of the software. However, in the process, important information about the quality of the software are not evidently reported to consumers, raising a relevant challenge with regard to the trust of the consumers on software providers. In addition, the centralized nature of most of software marketplaces results in "one size fits all" security checks, which are not appropriate for many security-critical applications, typically characterized by domain-specific requirements.

We believe that addressing these challenges is key to the success of the future Internet of Services, especially with respect to services that are considered highly valuable, sensitive critical or in the context of serious applications. Two key factors can contribute to that: the availability of a more detailed description of the security features of services and the possibility to include some additional guarantees on the quality of security mechanisms provided by established, domain-specific security experts (security certifications).

It is crucial that this information be provided to service consumers (human or software agent) in a machine-readable form such that they can check directly and just-in-time what specific security features are provided, what assurance they can get from a software product and how this assurance is provided. In this paper we introduce the concept of a trustworthy *Service Marketplace* (SM) that is suitable for hosting a larger class of security- and business-critical services, and service compositions for both businesses and end-users.

The remainder of this paper is organised as follows: Section 2 contains an overview on the state of the art in software marketplaces, Section 3 details the major challenges to be addressed towards a trustworthy SM, with particular attention to the limitation of current security certification schemes. Sections 4 presents our approach to tackle these challenges, while Section 5 illustrates the vision of a trustworthy SM. Finally, Section 6 concludes the chapter.

2 State of the Art

Before introducing the concept for trustworthy SMs, we analyse the state-of-the art in software marketplaces, and their relevant security checks. We focus mainly on mobile software markets, as they provide a large user base and are the

subject of many studies. This section is composed of two parts: we review the main marketplace approaches to security, then we focus on the security checks performed when an application/service has to be admitted in the marketplace (the *vetting process*).

2.1 Software Marketplaces

Security and trust play a major role for software market customers, for professional service users but also in the mobile device application consumption, due to the high sensitivity of information typically kept in mobile phones. Interestingly enough, in many markets, like the Apple App Store, security is not guaranteed [5], even if sometimes users' perception is different: what is provided is the availability of prompt security procedures, like the "kill switch" option, i.e., the automatic removal of any application, instructed for instance by Apple or Google, on all their produced mobile phones, without user intervention.

Marketplace operators can adopt different approaches to deal with security while delivering applications to end users: in particular, Barrera and Van Oorschot[5] propose three categories, "Walled garden", "Guardian" and "User control"; they range from a rigorous assessment of any applications on the market, to a completely open model, where security checks are upon user's responsibility. They also propose a classification of vetting tests for applications to be advertised on a (mobile software) marketplace. The seven categories are: "smoke tests", "hidden-API checks", "functionality checks", "intellectual property, liability and terms-of-service checks", "UI checks", "bandwidth checks", and "security checks".

In many software markets, the vetting processes are not described in details, we will discuss this aspect in the following Section 2.2.

Researchers have different opinions with respect to the role that software marketplaces can play in improving trustworthiness, and in bringing security to end-users, by means of their security assessments. Some authors stressed the difficulty to define a common concept of "security" for all users, given the multitude of different security requirements, according to contexts, users, and applications[14]. In the same paper, McDaniel and Enck argue about the possibility to introduce automated tests at application publishing phase, to check configuration settings, binaries and source code. The results should be then pushed to end users, allowing them to take the final decision about installing an application or not, based on their own security definition and requirements.

Other authors underline the role that central application repositories can have in filtering out malware applications. In particular, Gilbert *et al.* [10] analysed the benefits provided by a dynamic-analysis security validation tool that could be integrated in the software market approval process, but also scanning periodically the software market applications. The authors claim that such tool could be useful for specific purposes, like for instance for protecting the end users from privacy threats. Lastly, there is a certain emphasis given by some authors on publicly disclose the obtained results of software market assessments. For McDaniel and Enck, but also for Gilbert et al., the public availability of the

evaluation process can contribute to a more conscious use of technology by end users; especially with respect to sensible information leaking, letting them to be aware of risks they can be exposed to.

2.2 Vetting Process

We present in Table 1 a number of relevant marketplaces, together with their publicly disclosed security assessment criteria.

Salesforce releases a customer relationship management (CRM) system on the cloud that has a number of companion tools.

It permits third-parties to publish and advertise their applications (or extensions to existing Salesforce applications) that can operate on customers' data and information, on a specific marketplace with defined security review policies [23]. Google Apps Market is a store where third-parties can advertise complementary services for Google Apps services. Google explicitly inform its customers that no security checks are conducted on advertised applications [11]. Windows Azure Market is the official marketplace for Windows Azure (Platform-as-a-Service). Third parties can advertise their services, that apparently are not verified by Microsoft [16]. Existing marketplaces adopt the previously-described "User Control" approach. App Store and iOS, instead, can be seen as examples of "Walled Garden", meaning that anything that runs on served mobile devices must be explicitly approved by Apple. The app review process is not publicly disclosed; in a response to a FCC request in 2009, Apple disclosed some information[3], that are contained in Table 1. Microsoft offers Windows Marketplace [17] to users of its Windows Mobile OS. Application publishing and review process is documented in MSDN[15], the reference guide for any development effort with Microsoft technologies. Also Nokia has a specific certification process for publishing apps on its market [19], the Nokia Store[20]; nevertheless, newer Nokia's Windows mobile phones should follow Microsoft guidelines. RIM's App World is the reference software market for BlackBerry devices. Almost no public information on security assessment could be found, except those contained in [22]. In summary, where applicable, none of the above marketplaces discloses:

- the details of its security assessments, or
- the results of the vetting process for each applications.

This means that users have to cope with a "one-size-fits-all" definition of security, like in the majority of cases, having no option but to trust blindly marketplaces' procedures; or they have to face the absence of security assessments, having no option but to trust third-parties.

3 Challenges for Trustworthy Service Marketplaces

As discussed in the previous section, most marketplace operators enforce some sort of review and evaluation processes on applications before they are admitted to their marketplace. Security evaluation may involve security experts from the

Table 1. Security Features Of Existing Software Markets. Information marked with '*' are not completely publicly disclosed by providers.

Market name	Code reviews	Architectural review	Hands-on assessment	Periodic security review	Application Removal
Salesforce AppExchange	No	Yes	Yes	Yes	Yes
Google AppsMarket	No	No	No	No	Maybe
Windows Azure Marketplace	No*	No*	No*	No*	Maybe
App Store	No*	No*	No*	No*	Yes
Android Market	No	No	No	No	Yes
Windows Mobile Marketplace	No	Yes	No	Yes	Yes
Nokia Store	No	No	No	No	Yes
BlackBerry App World	No*	No*	No*	No*	Yes

marketplace operator and/or a third party security organization, who approve an application if it satisfies the security requirements defined by the marketplace operator.

The admission process of the marketplace compels the application providers to develop applications that address the security criterion specified by the marketplace operator.

However, this approach does not scale for different software provisioning scenarios.

Though the vetting processes increase the trust of the consumer on the security of the applications offered through the marketplaces, especially in the vision of a *service* marketplace, there are important problems that need to be addressed:

1. There is no information about the outcome of an evaluation available for the consumer, and the evaluation process is not disclosed in detail. Hence, trust in the secure operation of an application can only be built based on the reputation of the marketplace operator.
2. Consumers have specific security requirements for applications based on the operating domain and/or usage of the applications. However, marketplace operators have limited application- and domain-specific knowledge which is essential to perform any meaningful and effective evaluation on the security of the application, in a way that addresses the specific security requirements of consumers.
3. Current admission processes require the marketplace operators to own/control the execution environment of the applications, which is true for most of the current marketplaces. However this may not be the case in future marketplaces, especially in service marketplaces.
4. Admission checks cannot provide end-to-end security assurance for an application, especially when applications consume external services.

This means that the security requirements for a service significantly depend on the application domain, the application context, and the business context (intended usage). Hence, the security properties that a service provides should be evaluated and consequently certified by specialized entities that have the required domain- and application-specific knowledge. The lack of assurance on the

security of services is one of the key reasons of the trust deficit of consumers on such services [18]. Security certification of services can bridge this trust deficit by providing the required assurance on service's security. Though current security certification schemes are successful in providing assurance in monolithic software systems, they suffer from severe limitations when applied in a service environment due to economic and technological factors.

In addition, the stakeholders, the consumption models of current certification schemes are modelled for monolithic software and hence current schemes are inadequate to provide the security assurance in a service environment.

Some of the shortcomings of current certification schemes have conceptual reasons. Schemes such as Common Criteria are intentionally designed to be flexible and generic, in order to be able to certify different products ranging from software, firmware to hardware [25]. However this prevents these schemes to be prescriptive and so comparing certificates of different products becomes complex.

In addition, current certification schemes are structured in a manner that they cater to software provisioning paradigms where the consumer has control over the operation and execution of the product. However, in the service-oriented computing paradigm, the consumer does not have any control over the operational environment nor on the execution environment.

Another limitation is the application of current schemes in practice is a very expensive and time consuming process, often requiring years even for medium-level security assurance [25]. This is a major obstacle for services, where time-to-market can be a critical factor for the success of the service. Schemes such as Common Criteria allow a lightweight certification, but they lead to very low assurance. Also the evaluation is focused more on the accompanying documentation (Architecture, Design, Process related etc.,) or on the security processes followed, rather than the actual implementation of the product, especially at lower assurance levels.

The certification process, and results of the evaluation are captured in a human readable form that do not allow automated reasoning and processing to be performed. This is one of the major challenges that hampers the usage of current security certification schemes to service marketplaces where the security requirements of the consumers must be easily matched with the security properties of the services.

4 Building Blocks of Service Marketplaces of the Future Internet

4.1 Security Certification for Services: Assert4Soa

Current certification schemes have to tackle new challenges when approaching Internet of Services (IoS), for expressing, evaluating and certifying security properties for service-oriented applications. Therefore, novel models, techniques and tools are much needed; the Assert4Soa project aims at providing answers to these requests, defining a specific methodology as well as companion artefacts and tools [2,6].

Similarly to current security certification schemes, in ASSERT4SOA the assessment of the security properties of a service is performed by an independent third party (certification authority), who issues a corresponding signed assessment (*Assert*), bound to the service. The certification of a security property in an *Assert* is based on either a formal proof or on service testing that has been carried out before the certificate is issued. These formal proofs and tests must have been carried by the dedicated evaluation entity that has been accredited by the certification authorities. The ASSERT4SOA certification process will be semi-automated by using extensive tool support, as opposed to current certification schemes that depend heavily on manual effort.

A core feature of the ASSERT4SOA approach is a language, designed to express the security properties of a service as machine-readable, digitally signed statements (*asserts*), as opposed to existing security certificates that are expressed in a human readable form. The language allows the security features of a service to be represented at different levels of granularity ranging from abstract security properties to actual security functionalities that are implemented in the service. This is done in order to cater to the specific needs of different types of consumers that can range from users who have limited knowledge of service security to security experts of organizations who have specific requirements on the security functionalities of a service. The language also enables the representation of an abstract model of the service as part of the target of evaluation. This not only provides a description of the service to the consumers, but also serves to mitigate the concerns of the consumers on the lack of transparency of services.

In addition to the certified security properties, the language allows the representation of the information about the certification authority that has issued the certificate as well as the evidence that underpins the certified properties, i.e., the test suites or formal proofs used to evaluate the service. Hence, Asserts provide comprehensive descriptions of the security properties of the service.

Another important feature of the ASSERT4SOA project is the service-discovery framework. The service discovery framework provides consumers a query language through which they can express the functional and security requirements on the services. The query language allows the consumers to express the security properties at different levels of granularities as well as their preferences on the type of evidences for those security properties. The discovery engine, which is at the core of the service discovery framework, processes consumers requirements and performs matchmaking on the functional and security requirements using the functional and security matchmakers.

4.2 Component: USDL-SEC

Services published in marketplaces should be described in a manner that enables their discovery based on not only the functional requirements but also the security requirements of the consumer. However, the current description languages are not capable of describing the security properties of services. Though, some languages such as OWL-S [13] recommend using existing standards such as WS-Security [8], SAML [7] to describe security-relevant properties, they do

not provide a comprehensive specification. In order to overcome this limitation, we propose USDL-SEC, a new security specification model, that describes the security properties of services. This specification can extend existing service description languages such as USDL [21]. Service providers can use this specification to describe the security features of their services, and thus to support users in finding adequate alternatives to fulfil their needs.

The USDL-SEC model described here is currently being developed in the context of the EU-Funded FI-WARE[1] project. This model is globally organised in three main layers:

- **Security Topic:** This is a high level representation of the security feature of a service.
- **Security Solution:** This is a security mechanism that contributes towards satisfying a particular security topic.
- **Security Technology:** It refers to the technical implementations of the security solutions.

This three-layered model is materialized into a more concrete description model, depicted in Figure 1.

The model in composed by the following elements:

- **Security Profile:** the root node of the model and the entry point from USDL to USDL-SEC. This node should appear as a pointer element of USDL to the security properties of the service. This pointer can assume two different values, reflecting the categorization expressed in the previous section "USDL-SEC *target*": *Security service*, that refers to the Security-as-a-service paradigm, or *service with security features*, indicating that the service is a generic service with security capabilities.
- **Security Goal:** the security goal refers to the highest abstraction layer referring to a security topic. It can take the values of the most well known security concepts like Anonymity, Confidentiality, Privacy, Authentication etc. This list is defined using a security ontology ([12]).
- **Security Mechanism:** is a set of security solutions that can achieve a security goal. These mechanisms are theoretical solutions that answer to specific security requirements like Access control, Cryptography, Obligations, etc. These solutions can be applied under three realization levels: The network level, the application level, and the service level.
- **Security Technology:** is a set of concrete implementations and tools that realizes the security mechanisms. Like for example the encryption on the network level is implemented by IPSec [9].

As a use case example, the *Data Handling GE* service being developed in the FI-WARE project is described using USDL-SEC, as shown in Listing 1.1. This is a security service that protects sensitive data, by associating to each data transfer a specific privacy policy, and by enforcing its application. This service is assumed to be described in USDL for its business-related features. The USDL-SEC security

[1] `www.fi-ware.eu`

profile illustrates the security goals of the service (Privacy and Authorization); it also indicates the security mechanisms and technologies adopted to meet the security goals (Obligation and PPL Language in one case, AccessControl and XACML in the other).

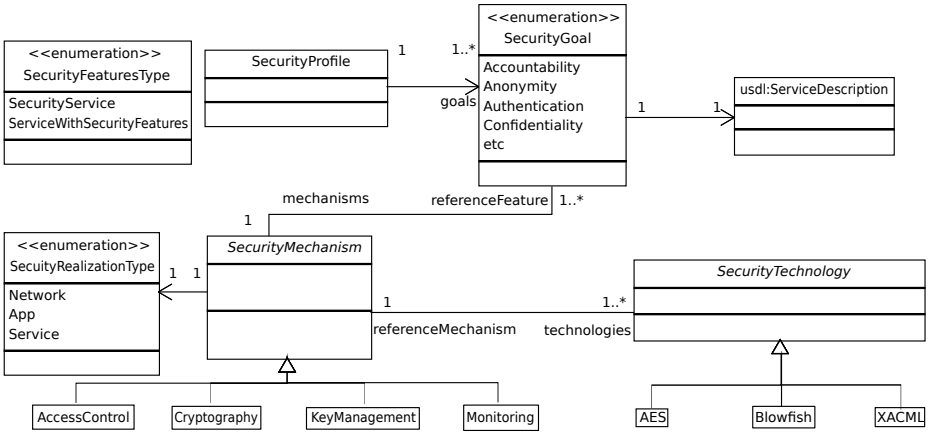

Fig. 1. USDL-SEC Specification Model

Listing 1.1. Draft for Data Handling GE in Fi-Ware platform

```
<#usdlSecDHGESecurityProfile> a sec:SecurityProfile;
        dc:title "Security profile of Data Handling GE";
        sec:providesSecurityFeature [
                a sec:SecurityFeature;
                sec:hasRealizationLevel sec:Service;
                sec:hasSecurityMechanism
                  [
                    a sec:Obligation;
                    sec:hasImplementation sec:PPL
                  ],
                  [ a sec:AccessControl;
                    sec:hasImplementation sec:XACML ]
                  ];
        sec:hasSecurityGoal sec:Privacy, sec:Authorization.

:PPLService a usdl:Service;
        sec:hasSecurityProfile <#usdlSecDHGESecurityProfile>.
```

5 Towards Trustworthy Service Marketplaces

Consider the scenario of a service consumer, who uses a SM to discover a service providing file storage functionality, in addition the consumer also has a security requirement that the file should be stored in a confidential manner. Now let us assume that there exits a service s, that provides confidential file storage. In the current SMs, the consumer cannot discover this service, as service

discovery based on security properties of services is not supported. Even if the consumer is able to discover the service s, there is still a lack of assurance that the security property of the service is indeed implemented correctly. We aim to overcome these limitations through the concept of Trustworthy Service Marketplace (TSM).

Our vision for TSM combines the description of service security features with supporting security certificates. While USDL-SEC allows the representation of the security features, the *Asserts* (Security Certificates) provide assurance to the consumers on the security features of services by providing evidences used to evaluate the services. These two approaches complement each other and together contribute towards increasing the trust of the consumer on the services offered through the marketplace.

The service providers should describe the security features implemented in the service using the USDL-SEC specification model before publishing them on the SM. They should describe the security topic, the security solutions and the security technologies implemented in the service. Though description of security features enables consumers to discover services that meet their security requirements, there is a lack of assurance that the security features are actually present and implemented correctly. In order to provide this assurance, service providers can obtain a security certification that evaluates the service thoroughly by using test suites or formal models.

The SM operator should use an advanced query language, that can be used by the consumers to express not only their business requirements, functional requirements but also their security requirements, assurance requirements and preferences. The USDL query language developed in the FI-WARE project allows the consumers to express their business, technical and functional requirements among others. The query language developed in the ASSERT4SOA project can be used for expressing the specific security, assurance requirements, and security preferences on the services. In this manner, a wide range of requirements can be used for querying the SM.

The traditional service discovery engines of the SMs should be augmented to use the USDL-SEC *Engine* and the *Assert Service Discovery* (ASD) framework. The USDL-SEC engine matches the requirements of the consumer with the security features of the services based on their USDL-SEC descriptions. The (ASD) framework allows the SM to discover certified services based on their security and assurance requirements and present them to the consumer. The ASD framework employs a matchmaking system that ranks services based on their *degree of fit* to the consumer's requirements. Though at a high level, there is an apparent overlap in the functionalities of the USDL-SEC engine and the ASD Framework, the functionalities complement each other in practise, where the USDL-SEC engine performs matchmaking on the abstract security requirements with service security descriptions, and the ASD framework performs matchmaking on the refined security requirements with certified properties of services along with their evidences. Together they provide a ranked list of services (recommendations) that

match the business, functional, technical, security, and assurance requirements of the consumers.

In addition to using the USDL-SEC and *Asserts* the SM operator could employ a vetting process, however the processes, the results of the vetting process must be made transparent to the consumer. The SM operators could also prescribe the services to comply with a standard USDL-SEC profile, accompanied by a security certification performed by independent Certification Authorities.

In the scenario mentioned above, if the consumer uses the TSM, he would not only be able to discover the service s based on the functional and security requirements, but also have assurance that the security requirements are actually met by the service.

6 Conclusions

Trustworthy Service Marketplaces can represent a key factor for opening new market perspectives for the future Internet of Services, especially with respect to sensitive, critical services and service composition. Trustworthy SMs will serve all their stakeholders with advanced and more secure services, as well as with transparent and evidence-based vetting processes. They will enable refined service discovery operations in marketplaces, also according to specific security requirements. Candidate services shall be then presented to users, along with their security certificates and evidences. In this way, a customer could evaluate each alternative according to her specific operational scenario. Trustworthy SMs could set certain security thresholds, such that a minimal security standard will have to be met by any of their advertised element. To sustain this vision, new technologies and standards are in development: digitally consumable service descriptions, covering business , technical, security and contextual aspects (USDL/USDL-SEC in FI-WARE); new assessment and certification methodologies, as well as digitally consumable certificates (ASSERT4SOA). Relying on assumptions and constraints expressed, more functionalities will come, like for instance a support for secure service compositions, through analysing security requirements and prerequisites of services, and secure deployment of services. We believe that trustworthy SMs can increase the trust and confidence in Internet-based systems, thus enabling even more sensitive operations to take place, in a secure, reliable and effective way.

Acknowledgements. This work is partially supported by projects FI-WARE (FP7-2011-ICT-FI - `www.fi-ware.eu`) and ASSERT4SOA (Grant No. 257351 - `www.assert4soa.eu`).

References

1. Amazon. Kindle, http://www.amazon.com/kindle-store-ebooks-newspapers-blogs
2. Anisetti, M., Ardagna, C.A., Guida, F., Gürgens, S., Lotz, V., Maña, A., Pandolfo, C., Pazzaglia, J.-C.R., Pujol, G., Spanoudakis, G.: ASSERT4SOA: Toward Security Certification of Service-Oriented Applications. In: Meersman, R., Dillon, T., Herrero, P. (eds.) OTM 2010. LNCS, vol. 6428, pp. 38–40. Springer, Heidelberg (2010)
3. Apple inc. FCCs answers, http://www.apple.com/hotnews/apple-answers-fcc-questions/
4. Apple inc. Official apple online store, http://store.apple.com/us
5. Barrera, D., van Oorschot, P.: Secure software installation on smartphones. IEEE Security & Privacy 99, 1 (2010)
6. Bezzi, M., Sabetta, A., Spanoudakis, G.: An architecture for certification-aware service discovery. In: Proc. of IWSSC (co-located with NSS 2011) (2011)
7. Cantor, S., Kemp, I., Philpott, N., Maler, E.: Assertions and protocols for the oasis security assertion markup language. OASIS Standard (March 2005)
8. O. W. S. S. Committee. OASIS web services security (WSS) TC OASIS, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss
9. Doraswamy, N., Harkins, D.: IPSec: the new security standard for the Internet, intranets, and virtual private networks. Prentice Hall (2003)
10. Gilbert, P., Chun, B., Cox, L., Jung, J.: Vision: automated security validation of mobile apps at app markets. In: Proceedings of the Second International Workshop on Mobile Cloud Computing and Services, pp. 21–26 (2011)
11. Google inc. Evaluate a marketplace app's security, https://support.google.com
12. Herzog, A., Shahmehri, N., Duma, C.: An ontology of information security. International Journal of Information Security 1(4), 1–23 (2007)
13. Martin, D., Burstein, M., Hobbs, J., Lassila, O., McDermott, D., McIlraith, S., Narayanan, S., Paolucci, M., Parsia, B., Payne, T., et al.: OWL-S: semantic markup for web services. W3C Member Submission 22, 200704 (2004)
14. McDaniel, P., Enck, W.: Not so great expectations: Why application markets haven't failed security. IEEE Security & Privacy 8(5), 76–78 (2010)
15. Microsoft inc. Market, http://msdn.microsoft.com/en-us/library/gg490776.aspx
16. Microsoft inc. Windows azure: Terms of use, https://datamarket.azure.com/terms
17. Microsoft inc. Windows marketplace, http://www.windowsphone.com/marketplace
18. Nasuni. Security and control are greatest concerns preventing enterprises from adopting cloud storage, http://www.nasuni.com/news/press_releases/
19. Nokia. Nokia ovi store content guidelines, http://support.publish.nokia.com
20. Nokia. Packaging and signing, http://www.developer.nokia.com/
21. Pedrinaci, C., Leidig, T.: Linked-USDL, http://linked-usdl.org/ns/usdl-core
22. RIM inc. BlackBerry app world, http://us.blackberry.com/developers/appworld/
23. Salesforce. Security review, http://wiki.developerforce.com/page/Security_Review
24. Szyperski, C., Gruntz, D., Murer, S.: Component software: beyond object-oriented programming. Addison-Wesley Professional (2002)
25. Zhou, C., Ramacciotti, S.: Common criteria: Its limitations and advice on improvement. Information Systems Security Association ISSA Journal, 24–28 (2011)