Leveraging Use of Software-license-protected Applications in Clouds

Wolfgang Ziegler[1], Hassan Rasheed[2] and Karl Catewicz[2]
[1]*Fraunhofer Institute SCAI, Department of Bioinformatics, 53754 Sankt Augustin, Germany*
[2]*Fraunhofer Institute FIT, User-Centered Computing Department, 53754 Sankt Augustin, Germany*

Keywords: Cloud Computing, Software, IPR.

Abstract: Running software license-protected commercial applications in IaaS or PaaS Cloud environments is still an issue that is not resolved in a satisfying way that benefit both the independent software vendor (ISV) and its customers. Due to the mandatory centralised control of license usage at application run-time, e.g. heartbeat control by the license server running at the home site of a user, traditional software licensing practices are not suitable especially when the distributed environment stretches across administrative domains. Although there have been a few bilateral agreements between ISVs and Cloud providers in the past to allow customers of these ISVs to run some of the ISVs license-protected applications in Clouds of certain providers a general solution is still lacking. In this paper we present an approach for software licensing that allows location independent use of software licenses both in form of delegation of already purchased on-site licenses to the Cloud and with authorisations for individual application executions in the Cloud.

1 INTRODUCTION

Cloud computing starts fulfilling its promise to provide a more flexible and cost-effective approach delivering infrastructure services than traditional IT services. However, cost-effectiveness and flexibility are foiled when the Cloud customer starts using commercial software on the acquired infrastructure resources since both the licensing technology and the business models of the independent software vendors (ISV) are not in line with the Cloud computing paradigm. Running software license-protected commercial applications in IaaS or PaaS Cloud environments is still an issue that is not resolved in a satisfying way that benefit both the independent software vendor (ISV) and its customers. The current technology and contracts force the customer to restrict the use of license-protected applications to internal private Clouds without the possibility of Cloud bursting or the use of multi-Clouds. The ISV on the other side is faced with increased usage of its software under the same yearly flat rate. Although there have been a few bilateral agreements between ISVs and Cloud providers in the past (see Section 2) to allow customers of these ISVs to run some of the ISVs' license-protected applications in Clouds of certain providers a general solution is still lacking. The objectives of the work described in the following sections are facilitating the use of license protected software in Clouds, increasing the flexibility of end-users to run their commercial applications in the most suitable and/or less costly environment, protecting the IPR of the ISV, and laying the foundations for new business models of ISVs by providing a novel software licensing and license management technology that is designed for todays distributed computing infrastructures. The solution is

- facilitating the use of license protected software in Clouds,
- increasing the flexibility of end-users to run their commercial applications in the most suitable and/or less costly environment,
- protecting the IPR of the ISV,
- laying the foundations for new business models of ISVs.

In this paper we present research and development results from the European project OPTIMIS (OPTIMIS, 2013) and their application in the industrial domain through a number of experiments in the European project Fortissimo (Fortissimo, 2016). The work in OPTIMIS was using the results of the European project SmartLM (SmartLM, 2011) as a basis for developing software licensing and license management for Cloud computing. The OPTIMIS results extend the prototype for use in multi-Cloud environments,

Ziegler, W., Rasheed, H. and Catewicz, K.
Leveraging Use of Software-license-protected Applications in Clouds.
In *Proceedings of the 6th International Conference on Cloud Computing and Services Science (CLOSER 2016) - Volume 2*, pages 265-273
ISBN: 978-989-758-182-3

both in form of delegation of already purchased on-site licenses to the Cloud and with authorisations for individual application executions in the Cloud. The approach supports both the traditional yearly flat rate licensing but also the pay-per-use approach, which is more appropriate for agile Cloud usage.

The remainder of this article is organised as follows. Section 2 presents related work in the area of license management in Clouds, Section 3 describes the basic technology upon which the OPTIMIS solution was build upon. The developments in OPTIMIS are described in Section 4. Section 5 presents the results of the evaluation of the implementation in a commercial Cloud infrastructure and Section 6 concludes the paper.

2 RELATED WORK

So far, commercial software has been and still is rarely used in Grids and public Clouds due to the limitations both with respect to the license management technology and the missing business models of the independent software vendors (ISV) for using their software in the Grid or the Cloud. Only in 2009 MathWorks has provided a technical solution (and a business model) allowing to use their MATLAB suite in the EGEE Grid environment (MATLAB, 2009). However, this is a bilateral agreement only and has so far no implications for using MathWorks software in other Grids like the German D-Grid. Lately, IBM launched a cooperation with Amazon allowing IBM's customers to use own software licenses for a limited number of applications under certain conditions in the Amazon Elastic Compute Cloud (EC2), which is extending BYOSL from the IBM Cloud to a public Cloud. However, the use of the "bring your own software and license" (BYOSL) (BYOSL, 2013) option would have to be settled by IBM with each Cloud provider where a user wants to deploy and use an application. In addition to iBM's BYOSL presented above there are first offerings of companies for Software Digital Rights Management technologies to be used in Clouds. Following we present approaches of those three that together have a market share of more than 80% (Sullivan, 2010): SafeNet, Flexera and Wibu (in descending order of the individual market share).

SafeNet as the market leader provides their product *Sentinel Cloud Service* (Sentinel, 2014), which is a framework for protecting software and data in the Cloud. This software has been provided since 2011. Sentinel Cloud Services is software for licensing and entitlement management delivered as a service for cloud services. It is supporting vendors of SaaS and PaaS service offerings. SafeNet Sentinel Cloud Services provides an alternative to traditional billing and payments services through its catalog-driven licensing and entitlement management solution. The solution includes the ability to package and re-package service offerings and pricing models, the ability to control what aspects of a service a specific end user can access including the particular access rights. In addition to the traditional approach of a license server on-site with firewall ports opened to control the application execution in the Cloud or a license server deployed in the Cloud, there is no specific support for IaaS. Drawbacks of these approaches are the additional security risks having open ports for bi-directional protocols and additional licenses liable to pay costs needed for the deployed license server.

In 2010, *Flexera* announced they would provide in 2011 Cloud-specific extensions to their product *FlexNet Licensing* (FLEXERA, 2014). The basic idea of the technology is binding the licensing process to the hosting environment or to the underlying hardware. The developments were made including cooperation with VMWare (vmware, 2014) and allow replacing the current binding of a license to a physical server (where the license server is running) to a virtual machine hosting the license server or the hardware the virtual machine is running on. However, the information on the Flexera web-site about the outcome of this cooperation still are more than vague. From the Flexera web-site it is not evident as to whether or not the development has achieved a mature state until today as planned.

Wibu, the third company among the big three, is a German SME providing solutions for protecting intellectual properties ranging from software to media. While the initial focus of the company was on dongle-based solutions, the company has extended its technology to software-based protection mechanisms. For Cloud environments, Wibu started research cooperation with the Fraunhofer institute ITWM to investigate in a solution based on the (Dalheimer and Pfreundt, 2009) license management (development of Fraunhofer ITWM). The state of the joint project S4Cloud is unclear as both partners don't provide more actual information other than a short article on the Wibu web-pages and ITWM's press release of 2010 (S4Cloud, 2015), (ITWM, 2010). The S4Cloud approach is based on software tokens that include a Hash of the input data of an application run. While this is a similar approach as taken by SmartLM, the major difference is that when the token is created, the ISV of the license-protected applications have to be contacted to get an online electronic authorisation. In

contrast, SmartLM does not need online authorisation of the ISV for creating a token since the ISV already authorised the license server of the licensee to act on his behalf.

To the best of our knowledge little public research has been focusing on licensing technology since the new IT infrastructure paradigms - Grids, Clouds and SOA - became serious extensions and replacements of traditional IT infrastructures. Early approaches like (Dong et al., 2005), (FU et al., 2007) and (Guofu et al., 2006) propose front-ends to the FlexNet License manager (FlexnetManager, 2015) providing scheduling and reservation of licenses. However, both approaches assume open firewall ports at runtime to enable the communication between license manager and application. (Dong et al., 2006) focusses on maximisation of license usage and resource usage in Grids. Like the previously mentioned approaches open firewall ports at runtime are a prerequisite. Other approaches like (Kwok and Lui, 2002) and (Liu et al., 2007) stem from the P2P environment. The former addressing licensing of music sharing while the second one is more generally addressing content sharing. However, both approaches grant unlimited access or usage once a license has been issued and thus do not support a business model useful for ISVs. (Katsaros et al., 2009) finally is proposing a license mechanism suitable for SOA environments. However, the paper sketches the architecture and some possible interactions but lacks an implementation and experiments with real applications. Moreover, the approach also assumes open firewall ports at runtime. Only recently when these new paradigms gained ground in productive environments where e.g. more commercial simulation codes are used than in the e-Science domain license technology came to the fore. In (Li et al., 2008) the authors give an overview on current licensing technology and models and describe two approaches developed in European projects to overcome the limitations. One of the presented approaches breaks with the current technology and has been implemented as prototype in the SmartLM project while the second approach circumvents some of the limitations imposed by the de-facto standard of software licensing. In the European project BEinGRID another approach was developed which allows the use of existing licenses in Grid environments through tunnelling of the communication of the license server to the application (Raekow et al., 2009). While technically feasible this approach raises a number of legal issues since many license contracts prohibit the use of a software license outside a company or outside a certain radius from the company. Furthermore, this approach is no longer maintained since the end of the BEinGRID project.

As of today (and as far as can be judged from the publications on the respective web-sites) the approaches of the major players in he field of Software Digital Rights Management are still requiring network connectivity to the user's site license server during application execution in the Cloud, manual interaction with the ISV and are not transparent for the user. The SmartLM baseline technology for licensing and license management extended in OPTIMIS overcomes these limitations.

3 OPTIMIS LICENSE MANAGEMENT BASELINE TECHNOLOGY

The key for flexible software licenses that can follow applications into Clouds without the need to access an on-site license server at runtime is the separation of (1) authorisation for license usage and (2) authorisation for application execution. (1) is done at the site of the user considering the local policies for using licenses, e.g. department quota. The result of a successful authorisation is a reservation of a license for a certain time and a software token, that contains all information to allow the policy enforcement point in the application to decide on the execution. (2) is done later in the Cloud where the token is examined when starting the application. The prototype solution developed in the European project SmartLM implemented this separation and provides a software token mechanism that contains all authorisation information required by the application API to validate the request for executing an application and forward the license information to the applications policy enforcement point in case of a successful validation of the token.

While usually unreachability of license servers, e.g. due to firewall rules, leads to applications aborting during start-up the token provides off-line access to licenses for authorising the execution of an application in a Cloud beyond the administrative domain of the site running the license server .

4 CLOUD ENVIRONMENT-SPECIFIC PROVISIONS

Figure 1 depicts the structure of a license token highlighting the different levels of protection against fraud

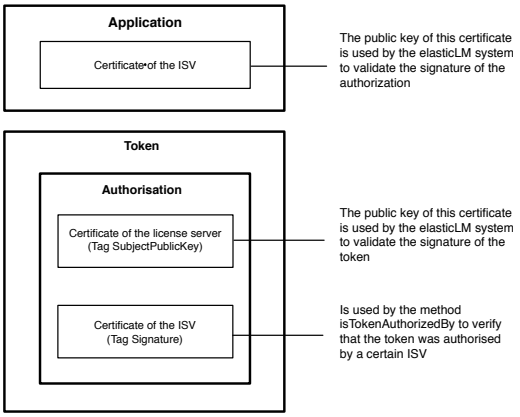

Figure 1: Structure of the license token.

though a chain of trust created with signatures of all actors involved in the creation of a token. Besides this internal protection the token may also contain a SHA-2 hash of the input data needed to run the application. This has binds the token to the input data and allows execution of an application only with these input data. As a result, the token can be copied but the copies can never used with other data than the token was created for.

In OPTIMIS additional approaches have been implemented enhancing the SmartLM solution: (i) dynamic deployment of a trusted instance managing a number of tokens for one or multiple applications and for one or multiple users, and (ii) dynamic deployment of a full license service with a subset of the licenses available at the home organisation of the user. The configuration of the dynamically deployed license service is also managed by the VM contextualiser. The second approach is especially useful when the same Cloud resources are used over a longer period of time for running license protected applications. In the first approach both the user and the contextualiser can be responsible for configuring and deploying the trusted instance for the respective network environment and to transfer tokens.

4.1 Contextualisation

All necessary applications, tools and probably corresponding input data to be processed using Cloud resources can be included in images prepared in advance for deployment in a IaaS Cloud. These images can be used multiple times in different environments. As a consequence, environment specific data, like e.g. networking configuration, security customisation or software license information are not part of these images and need to be inserted prior to deployment. As part of this contextualisation process in OPTIMIS also the necessary license tokens for executing a license-protected application are created dynamically and inserted into the image (Armstrong et al., 2011), hence, realising License as a Service (LaaS). In case of multiple applications in a VM, e.g. for a workflow, the VM contextualizer assures that all required licenses are in place when the applications start up. No communication between the application and the license server that issued the token is required at runtime.

4.2 License Delegation

This approach is based on the license delegation developed in OPTIMIS and the deployment of a license server as part of the contextualisation. It further requires a trustworthy Cloud provider willing to provide an additional service for retrieving environment-specific data in form of a hash value of this data. We suggest using SHA-2, which we consider being safe enough given the time constraints encoded in the token and the limited life-time of the delegated licenses while the time requirements for this algorithm are suitable for a user-steered on-line process. The trusted instance is operating on tokens that are created beforehand when it is known in advance which applications will be used and which licenses are required. The tokens can be prepared at the user's premises. The license server deployed on the IP's Cloud infrastructure supports dynamic, on-the-fly creation of tokens that are needed for running the applications. See Figure 2 for the details of the splitting process. Since the license server creating the tokens is running in the Cloud environment, it may use the provider's service to get the hash of some environment-specific parameters.

When creating the token the hash can be included and be verified online when the token is processed in the Cloud. The deployed license service (Server B) is a copy of the license service that runs locally at the premises of the user (Server A). This license service is part of the VM so it can be deployed on the designated Cloud infrastructure together with the applications and data. However, the total number of licenses and features initially procured from the ISV must remain the same when a license service is running in the premises of the user and in the Cloud. To achieve this, the licenses and features made available for the license service in Server B in the Cloud are blocked in Server A and cannot be used locally anymore. Figure 2 depicts the process of preparing a subset of the licenses and features available at Server A to be added to Server B in the Cloud (license delegation).

As shown above, the initial authorisation issued by the ISV for Server A to install and use a license

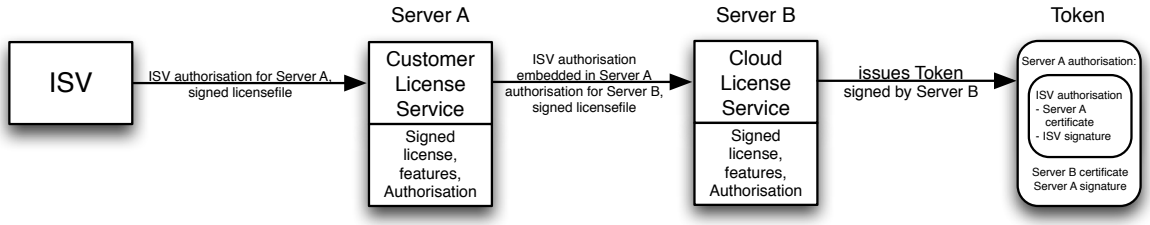

Figure 2: Process of license splitting.

procured from the ISV can be delegated by Server A to Server B. This delegated authorisation includes the authorisation of the ISV, the certificate of Server B and is signed by Server A. Server B includes the authorisation of Server A into each created token. This allows the API to validate the entire chain of trust up to the ISV when a token is processed. The authorisation for using the application is not rejected when the chain is broken, i.e. when the token is not created by Server B but by another copy of the license service running on another server. Moreover, the API can check at runtime whether the license and features contained in the token are blocked for local use at Server A.

As an extension of this mechanism we consider the Cloud provider offering a service that allows retrieving key parameters of the Cloud environment encoded in a SHA-2 hash key. The parameters should be significant for the respective environment and thus deliver a distinct hash key per Cloud environment. Some kind of standardized set of parameters should possibly be defined to achieve the goal of uniqueness across different providers. A first set of parameters to start the evaluation included: Cloud provider's name, Cloud provider's access point IP address, type of Virtualisation Software, operating system, date/time until the hash key is valid, some unique properties of the hardware the VMs will be executed upon.

With this extension, the license delegation mechanism would look like the diagram depicted in Figure 3. The tokens created by the license server in the Cloud contain the hash key of the local Cloud environment as additional information. When the application starts executing, the API validates the token as usual. In addition, it also makes a call to the provider's information service to retrieve the hash key. If the hash key matches the one in the token, the API continues providing the license information in the token to the policy enforcement point in the application to grant or deny further execution of the application.

Adding the hash key of the provider's environment adds additional protection against fraud. E.g. just cloning the VM with the license server and running it in a different Cloud environment is not easily possible. Of course, the level of desirable protection (and the effort spent for achieving this) strongly depends on the value that is to be protected. Also, the effort put into breaching a mechanism depends on the value that is protected by this mechanism. While this approach is able to increase the token protection against fraud, it also limits the flexibility that the usage of Cloud resources may provide. Binding the token to information retrieved from the execution environment inhibits Cloud dynamics like Cloud bursting. In this approach we would need a provider to cooperate and implement the hash function. In contrast to the trusted instance described before in section 4.2, the solution depends on a third party the Cloud provider delivering the hash key and as such is not as easy/fast to implement as the previously described approach.

Clearly, the ISV has to allow license delegation. The ISV can do this implicitly through providing an extended API that is able to validate the delegation chain. Any non-enhanced API would reject the tokens that were created based on a delegated license. In case the ISV does not allow license delegation another possibility for providing multiple execution authorisations in the Cloud is the Trusted Instance described in the next section.

4.3 Trusted Instance

The rationale behind the trusted instance is twofold: providing a secure container for tokens that in addition is able to communicate with the API (the applications policy decision point) and to deliver all information to the API required to authorise a user's request to launch an application. As depicted in Figure 4 the same mechanism for binding tokens to the respective Cloud environment can be used as with license delegation.

The trusted instance verifies that the token is valid. Additionally, depending on firewall restrictions, it may provide a secure channel that can be used for communication with the license server located at the user's premises (e.g. to verify the status of a token prior and during the operation of the license-protected application or to cancel a reservation). If renegotiation of the license terms is allowed (which is defined

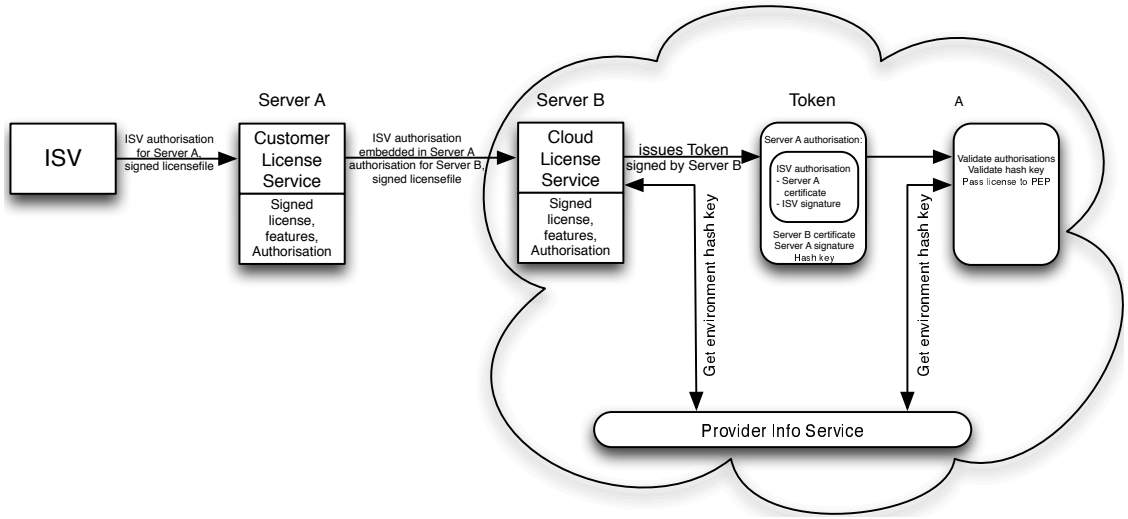

Figure 3: Process of the license delegation.

in the token) and the secure channel is available, renegotiation of license terms could be started in case of shortage of resources, e.g. time running out or additional features needed due to the dynamic nature of the application. The trusted instance is in a VM that can be deployed in the Cloud together with the application VM(s). The OPTIMIS contextualiser is responsible for setting-up and configuring the trusted instance in the VM at deployment. This includes network address configuration, adding an initial set of tokens, etc. The necessary contextualisation information is provided in the service manifest, which is used in OPTIMIS to allow a service provider or user to describe its requirements regarding the Cloud environment and the services provided by the infrastructure provider.

The major benefit of using a trusted instance is that the token does not need to be accessible by the applications API where it potentially may be accessed by a malicious user to replicate it. Rather, the trusted instance provides an assertion to the API, which includes the content of the token. Another benefit in terms of software development effort is that this approach does not require changes in the processing of the authorisation in the API, because the only difference is the source of information, namely the trusted instance instead of a local token file read by the API. It only requires the implementation of a protocol that retrieves the authorisation from the trusted instance.

Finally, multiple tokens can be generated in advance according to the requirements of the service to be run in the Cloud and deployed with the trusted instance during the contextualisation of the virtual machines for a service deployment. This reduces the communication requirements during the service execution time and re-contextualisation caused by missing tokens.

5 EVALUATION

Evaluation was organised in two phases. First, we asked the commercial UK Cloud provider Flexiant (a partner of the OPTIMIS project) to run a number of test cases in their production infrastructure to test and evaluate the implementation. The tests comprised the license delegation (7 test cases), the trusted instance (12 test cases). including integration tests and tests that simulate users that on purpose or accidentally messed around with license tokens. We also did some measurements regarding the overhead in terms of additional time needed (results are discussed below). Second, the software was used in the Fortissimo More.Cloud experiment[1] to realise a one-stop-shop solution for simulations provided by an ISV. Here we focused on the license delegation.

Summary of Results: The test cases passed with the expected outcome with the exception of one integration test, which was only partly satisfied. The reason for the failure was identified as an invalid configuration file which was fixed afterwards. The License-Token validation checks successfully passed. Multiple cycles of the test cases were performed, each yielding similar results and little deviation to the test case result pattern. Overall, the test all cases passed as expected.

[1]Main Routing Architecture Optimisation Research Experiment. MORE is aimed at reducing weight and cost for wiring systems for complex products such as aircraft.

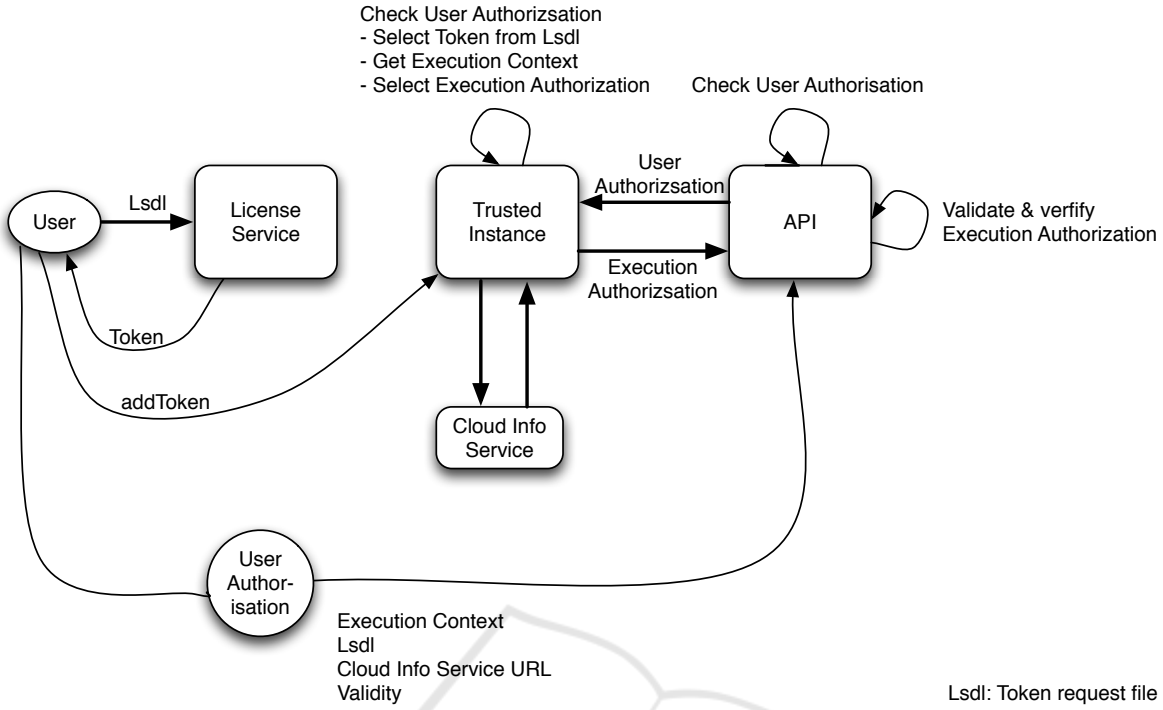

Figure 4: Deployment of the Trusted Instance in a Cloud.

Incurred Overhead: Adding license tokens to a VM is part of the contextualisation process that happens anyway. Thus there is no additional overhead. If multiple license tokens are required (i) or if license tokens need to be created dynamically depending on the outcome of applications in a workflow (ii) then for (i) a VM with a trusted instance needs to be deployed or for (ii) a license needs to be split at the license server running at the user's site, included through contextualisation into a VM with a license server that is then deployed to the Cloud as any other VM. The overhead is in the range of about 3-5 minutes which is negligible since is happens only once and many (short) jobs can be run afterwards using the license or license tokens deployed.

Risk of Fraud: The use of a trusted CA for creating the required certificates is essential to avoid attacks on the level of the signatures. Of course, once the code responsible for evaluating the validity of a token (the policy decision point, PDP) in the API of the application, or the policy enforcement point in the application that used the outcome of the PDP is hacked the application may be used without limitation just as with any other software licensing solution.

6 CONCLUSION

In this paper we described the work on software licenses and software license management done in the European OPTIMIS project which is now used in the European project Fortissimo. We introduced the baseline technology taken from the European project SmartLM which developed a licensing solution for Grids. Based on this baseline two extensions have been developed and implemented in OPTIMIS: the license delegation and the trusted instance. These extensions allow execution of license-protected applications in the Cloud without reducing the level of protection of the ISVs IPR. Furthermore, the token based license management allows new business models like pay-per-use where each token can be used as a secure off-line authorisation for running an application and account for exactly this execution without the need to have an on-line connection to the license server at run-time.

As a next step towards commercialisation of elasticLM (elasticLM, 2015) we used it in a productive environment for simulations of aircraft wirings in the European funded project Fortissimo (Fortissimo, 2016). A one-stop-shop solution for ISVs has been developed allowing an ISV to offer its SME customers the a customised infrastructure needed for simulations in a Cloud environment. The offering

is based on a pay-as-you-go model that requests the SME only to pay for resources (including temporary licenses) actually required to perform the simulation work needed for its business. The solution developed in Fortissimo includes access to the Cloud resources determined by the size of the problem, access to the simulation application deployed in the Cloud, and dynamic access to the necessary software licenses to run the simulation application. Currently elasticLM is used in another productive environment for simulations of metal sheet forming processes in the automotive industry within the Fortissimo project.

The experiments in Fortissimo have proved that license server can easily installed in a virtual machine, furnished with the licenses required for running software in a contextualisation step, and deployed into an arbitrary Cloud infrastructure: public, private, hybrid. The software framework will now be made available for download in the Fortissimo marketplace (FortissimoMarketplace, 2015) and later - based on the experience in the Fortissimo Marketplace - also in the AWS Marketplace (AWSmarketplace, 2016).

ACKNOWLEDGEMENTS

Some of the work reported in this paper has been funded by the European Commissions ICT programme in the FP7 project SmartLM under grant #216759 and in the FP7 project OPTIMIS under grant #257115.

REFERENCES

Armstrong, D., Djemame, K., Nair, S., Tordsson, J., and Ziegler, W. (2011). Towards a Contextualization Solution for Cloud Platform Services. In *2011 IEEE 3rd International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 328–331. IEEE.

AWSmarketplace (2016). Amazon Webs Services marketplace. Website. Online at https://aws.amazon.com/marketplace, visited 31 January 2016.

BYOSL (2013). IBM Licensing for Amazon Cloud web site. Website. Online at http://www-01.ibm.com/software/passportadvantage/eligible_public_cloud_BYOSL_ policy.html, visited 10 November 2015.

Dalheimer, M. and Pfreundt, F.-J. (2009). GenLM: License Management for Grid and Cloud Computing Environments. In *Proceedings of the CCGrid conference 2009*.

Dong, X., Wang, Y., Zheng, F., Guo, H., Yang, S., and Wu, W. (2005). Floating license sharing system in grid environment. In *SKG*, page 96.

Dong, X., Wang, Y., Zheng, F., Qin, Z., Guo, H., and Feng, G. (2006). Key techniques of software sharing for on demand service-oriented computing. In *GPC*, pages 557–566.

elasticLM (2015). elasticLM - License as a Service (LaaS) . Website. Online at http://www.elasticlm.com, visited 10 November 2015.

FLEXERA (2014). FlexNet Licensing. Website. Online at http://www.flexerasoftware.com/products/flexnet-licensing.htm, visited 10 November 2015.

FlexnetManager (2015). Flexera web site. Website. Online at http://www.flexerasoftware.com/products/flexnet-manager.htm, visited 10 November 2015.

Fortissimo (2016). Fortissimo - Enabling European Small and Medium Enterprises (SMEs) in the manufacturing sector to benefit from high-performance digital simulation and modelling. Website. Online at http://www.fortissimo-project.eu/index.html, visited 10 January 2016.

FortissimoMarketplace (2015). Fortissimo Marketplace. Website. Online at https://www.fortissimo-marketplace.com/infopage/, visited 31 January 2016.

FU, W., XIAO, N., and LU, X. (2007). Sharing software resources with floating license in grid environment. In *NPC '07: Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing Workshops*, pages 288–294, Washington, DC, USA. IEEE Computer Society.

Guofu, F., Yinfeng, W., Hua, G., and Xiaoshe, D. (2006). Research on software license manager and sharing system in grid. In *GCCW '06: Proceedings of the Fifth International Conference on Grid and Cooperative Computing Workshops*, pages 35–38, Washington, DC, USA. IEEE Computer Society.

ITWM (2010). ITWM S4Cloud project. Website. Online at http://www.itwm.fraunhofer.de/presse-und-publikationen/pressearchiv/pressearchiv-2010/04032010-itwm-mit-wibu-systems-auf-der-cebit.html, visited 10 May 2015.

Katsaros, Gregory, A., Savvas, K., Dimosthenis, and Varvarigou, T. (2009). Service Oriented License Providing. In *Proceedings of IEEE International Conference on Service-Oriented Computing and Applications (SOCA)*.

Kwok, S. H. and Lui, S. M. (2002). A license management model for peer-to-peer music sharing. *International Journal of Information Technology and Decision Making*, 1(3):541–558.

Li, J., Wädrich, O., and Ziegler, W. (2008). Towards sla-based software licenses. pages 139–152.

Liu, Y., Yuan, C., and Zhong, Y. (2007). Implementing digital right management in p2p content sharing system. In *ICA3PP*, pages 348–355.

MATLAB (2009). The MathWorks Enables MATLAB Parallel Computing Tools to Run on the EGEE Grid. Website. Online at http:// www.mathworks.com/matlabcentral/fileexchange/21426-enhancing-e-infrastructures-with-advanced-technical-computing–parallel-matlab%C2%AE-on-the-grid, visited 10 November 2015.

OPTIMIS (2013). OPTIMIS - Optimised Infrastructure Services. Website. Online at http://www.optimis-project.eu, visited 10 November 2015.

Raekow, Y., Simmendinger, C., and Krämer-Fuhrmann, O. (2009). License management in grid and high performance computing. *Computer Science, Research + Development*, 23(3-4):275–281.

S4Cloud (2015). Wibu S4Cloud project. Website. Online at http://www.wibu.com/data-security-research/s4cloud.html, visited 10 November 2015.

Sentinel (2014). SafeNet Sentinel Cloud Services. Website. Online at http://www.safenet-inc.com/software-monetization/sentinel-cloud-services-overview/, visited 10 November 2015.

SmartLM (2011). SmartLM - Grid-friendly software licensing for location independent application execution . Website. Online at http://www.smartlm.eu, visited 10 November 2015.

Sullivan, F. . (2010). World Software Digital Rights Management Market. Technical Report N671-70.

vmware (2014). Flexera-VMWare cooperation. Website. Online at http://www.flexerasoftware.com/partners/licensing-entitlement-management-alliance-partners.htm, visited 10 September 2015.