Collaboration and the Knowledge Economy: Issues, Applications, Case Studies
Paul Cunningham and Miriam Cunningham (Eds)
IOS Press, 2008 Amsterdam
ISBN 978–1–58603–924-0

Support for Client-Server based License Management Schemes in the Grid

Christian SIMMENDINGER[1], Ottmar KRÄMER-FUHRMANN[2],
Yona RAEKOW[2], Hubert HERENGER[3]
[1]*T-Systems SfR, Pfaffenwaldring 38-40, Stuttgart, 70569, Germany*
[2]*Fraunhofer SCAI, Schloss Birlinghoven, Sankt Augustin, 53754, Germany*
[3]*HLRS, Nobelstrasse 19, Stuttgart, 70550, Germany*

Abstract: License Management is currently not supported in any Grid middleware. Since most small and medium enterprises (SME) use commercial applications with associated licensing issues, this lack of support is becoming a major obstacle for the commercial exploitation of these middleware and the corresponding Grid infrastructure. In order to resolve this obstacle, we have designed and implemented a complete grid-friendly License Management architecture, which supports the currently used client-server license schemes. In this paper we discuss the process of eliciting license management related requirements, the subsequently derived common components and their corresponding design patterns as well as details of implementation patterns.

1. Introduction

Over the last few years, Grid technology has evolved from a technology designed largely for the needs of the High Performance Computing (HPC) community towards an open framework supporting the general business domain. In order to foster the adoption of Grid technology in European business and society BEinGRID [1] has gathered the requirements for a commercial Grid environment from a first wave of 18 business experiments. One of the key elements derived from this elicitation of requirements is support for commercial applications from independent software vendors (ISV) in grid environments. In order to meet the requirements of an on-demand computing scenario a pay-per-use License Management scheme needs to be established.

1.1 Background: ISV simulation applications in industry.

Especially small and medium enterprises (SME) from the engineering community stand to profit from pay-per-use HPC Grid scenarios. Very few of these SMEs however maintain their own simulation applications. Instead commercial applications from independent software vendors (ISV) are commonly used with an associated client-server based licensing. For the latter typically FlexNet available from Acresso [2] (formerly Macrovision) is employed, which is the quasi-standard in this area. The authorization of these client-server based license mechanisms relies on an IP-centric scheme - a client within a specific range of IP-adresses is allowed to access the license server.

1.2 Gap Analysis

Due to this IP-centric authorization, arbitrary users of shared (Grid) resources may access an exposed license server, irrespective of whether or not they are authorized to do so. Secure and authorized access to a local or remote license server in grid environments therefore has not been possible so far. The use of commercial ISV applications in grid

environments therefore was not possible either. The here presented License Management (LM) architecture resolves this problem. Amongst other features the LM architecture also provides a cost-unit based accounting with the possibility to check validity (e.g. in terms of available funding) of license accounts at job submission time. Neither of these features so far existed. The solution is generic and independent of any grid middleware.

1.3 Business Impact

The lack of a grid solution for client-server based license management readily implies that the vast majority of users from industry has not been able to use their ISV applications in grid environments. The here presented license management architecture thus potentially increases the grid market size in the area of on-demand computing by industry by a large factor.

1.4 The Pay-per-Use model

Prices for licenses for ISV simulation applications in the area of HPC typically exceed the cost of correspondingly required resources (CPU, memory, filespace) by more than two orders of magnitude -- a single license can cost up to 100.000 Euros per year. A pay-per-use model therefore seems to be required in a order to provide a satisfactory on-demand computing scenario with a licensed application from an ISV.

Currently licenses are issued on a yearly basis: Customers buy a fixed number of licenses, with associated features and an included support. The generated revenue for the ISV therefore is predictable and stable. This business model also guarantees that the provided simulation applications are always in line with the requirements of the end-users: There is a close dialog between ISV and end-user.

Contrary to this, in a pay-per-use scenario there is no predictable revenue for the ISV – and unless the ISV is also the license service provider (LSP) – the ISVs would loose the direct contact to their end-users. Moreover the currently established business model (yearly licenses) implies a substantial over-provisioning of licenses: End-users need to buy more licenses than they require on a daily basis in order to satisfy their peak requirements. With a pay-per-use scenario this over-provisioning immediately becomes obsolete with a corresponding loss of revenue for the ISVs. On the other hand a pay-per-use model would create a new source of revenue for ISVs, because SME which so far could not afford to purchase licenses can now access the licenses on a pay-per-use basis. Additionally large customers become able to dramatically increase the number of licenses during peak-demand periods.

These contrasting business models make a direct transition towards a pay-per-use license model on a new technology basis rather unrealistic. Instead ISVs will need to constantly evaluate and refine the evolving new business models in a non-interruptive transition on the basis of currently used technology.

From the perspective of the proposed license management architecture this implies that the architecture needs to be able to support a fluent transition from currently used technology towards a grid-friendly pay per use scheme. The here proposed solution provides a solid basis for that transition.

2. Objectives

The purpose of this paper is to highlight the relevance of license management for the Grid -- where we refer to the introduction -- and to convey the results of the use-case and requirements analysis performed in BEinGRID with respect to license management. From this analysis design patterns and component descriptions for a new license management

architecture have been derived. We also describe implementation patterns for the license management components as well as correspondingly developed components.

3. Methodology

A number of common requirements have been elicited from the initial wave of 18 business experiments in the BEinGRID project. Based upon this analysis design patterns and component descriptions for a client-server based license management scheme have been derived. Implementation patterns are discussed and a reference implementation is presented.

3.1 Requirements

The following license related requirements have been elicited:

1. **Gridification of currently used License Management Systems**: Support for license management was required for four different middlewares: GRIA, GT4, UNICORE and gLite – and all of the four business experiments required support for FlexNet. Since predictably no LSP or ISV is going to support four different middlewares we hence aimed for a generic middleware independent solution, which would support an arbitrary client-server based license scheme including FlexNet. In fact the solution is even usable in a non-Grid LSP context and hence is able to cover the complete spectrum of required scenarios. There are some side-implications with respect to accounting associated with the above requirement: Whereas in the non-Grid scenario the bill already has been paid in advance – and accounting plays a minor role – the pay-per-use model would at least need to support a flexible cost unit based accounting rather than an identity bound accounting: The reason is that usually institutions or research groups own the licenses – not their individual members. Also licenses are typically used in a specific accounting context, namely cost units.
2. **Limited LSP Capability**: The required fluent transition to a pay-per-use model – with respect to the underlying business model – made it likely that at least for a short transitional period (where 'short' can mean: up to a decade) the ISV will also assume the role of the license service provider (LSP) for all its customers: ISVs need to be able to quickly implement and refine the evolving new business models. This immediately implied that scalability of the license service would become an issue. In this transitional period the ISV would have to maintain potentially several thousand simultaneous connections to the FlexNet server. (FlexNet itself is able to handle this amount of connections).
3. **Grid License Models**: The FlexNet scheme (floating licenses from a single – possibly redundant – server) or other Client-Server schemes are limited in their scope and scalability. An extension of such a scheme to a scenario where licenses are offered as standard resources in the Grid therefore seems to be a logical step. This scenario readily implies that licenses have to be scheduled like other resources e.g. networks or CPUs. Licenses are typically an even more precious and expensive resource than compute-power. It therefore is highly important that licenses are not unnecessarily blocked by queued jobs or that jobs, which already have allocated their computing-resources do not get blocked by missing license-keys. Interfaces to e.g. Grid schedulers therefore are needed. An extension of that scenario, where license owners are allowed to re-sell their licenses on a LSP basis should be feasible with respect to technology. This scenario extension would, however, certainly require the cooperation of ISVs. An even bigger step for the ISVs – but maybe a logical one – would be to entirely drop the concept of selling a yearly license and to instead introduce a -- probably token based – system where licenses are accounted and billed on a per job basis.

3.2 Common Capabilities

1. **License Management related Extension of Job Description and Submission**: This Common Capability covers the extension of the job description and its submission with respect to license management. A user needs to provide details about the requested licenses – including authorization – as well as the accounting context. This Common Capability (CC) allows a user to request license resources in a similar manner as currently implemented in Grid middlewares for any other resource (cpu, memory, etc). The resources here can be either own licenses, licenses provided by the service provider or an external LSP.
2. **License Management related Authorization for Job Submission**: This Common Capability covers authorization mechanisms required with respect to License Resource Requests. The sole deviation from current standard authorisation mechanisms is the type of resource, namely whether or not a user is entitled to use a specific license server, specific features of the licensed software or whether limits exist with respect to the number of licenses. In complete analogy possible solutions range from simple locally maintained lists or pin / tan mechanisms up to a full integration into identity management systems like Shibboleth or VOMS with explicit requests to home organizations the of users and/or third party license service providers (LSP) or even requests to License Brokers.

Figure 1:License Management related Authorization

3. **License Management related Scheduler Extension**: This Common Capability covers the extension of a local scheduler. If access to the license servers has been granted, the scheduler needs to schedule the job – according to policies and – possible remote – resources – and to pass the information about the license request details to the resource management system.
4. **License Management related Resource Management Extension:** This Common Capability covers the extension of a local resource management system. In a batch prologue the resource management can dynamically reconfigure the license proxy in order to grant access for a specific userID at the assigned – ideally dedicated – local resources. Correspondingly in a batch epilogue the proxy is reconfigured in order to prohibit non-authorized access. At this point security of access to the license server is transferred from a – e.g. certificate based – authentication/authorization level to the required network level security at which e.g. FlexLM operates. However, this is only required if there is no additional run-time authorization at the license server.
5. **License Proxy:** All external accesses from dedicated local resources are re-routed via to the proxy, which can allow or reject these connections (see License Management related Resource Management Extension) and can re-route this request via a proxy-chain (including run-time authorization) to the remote license server. The proxy and possibly its upstream counterpart both log access time, duration and accounting context.
6. **License Management related Accounting and Billing**: This Common Capability covers the Accounting and Billing of licenses. In order to produce the complete accounting log, information from both the proxy (userID / time-stamp) and the license server (userID / time-stamp, number of licenses, license features) are required. The actual details of billing and accounting not only are depending on middleware, but also

on the underlying business model. Depending on whether licenses are owned by the user, the service provider, an external static LSP or obtained via a Grid broker, the exchange and assembly of the actual accounting and billing information will differ.

7. **Encapsulation of License Server** This Common Capability relates to the integration of existing license servers. It not just addresses a possible encapsulation as a web service but more generally an integration of the license server into the respective Grid middlewares. Remark: In our License Management Architecture Implementation the upstream proxy partly provides this encapsulation. (Run-time authorization)

3.3 License Management Architecture

The License Management Architecture components (LM-Job Description and Submission, , LM-Authorization, LM-Proxy, LM–Monitor, LM-Accounting) are designed to provide a complete License Management for FlexNet (or more generally: client-server license models) based Independent Software Vendor (ISV) applications in Grid environments.

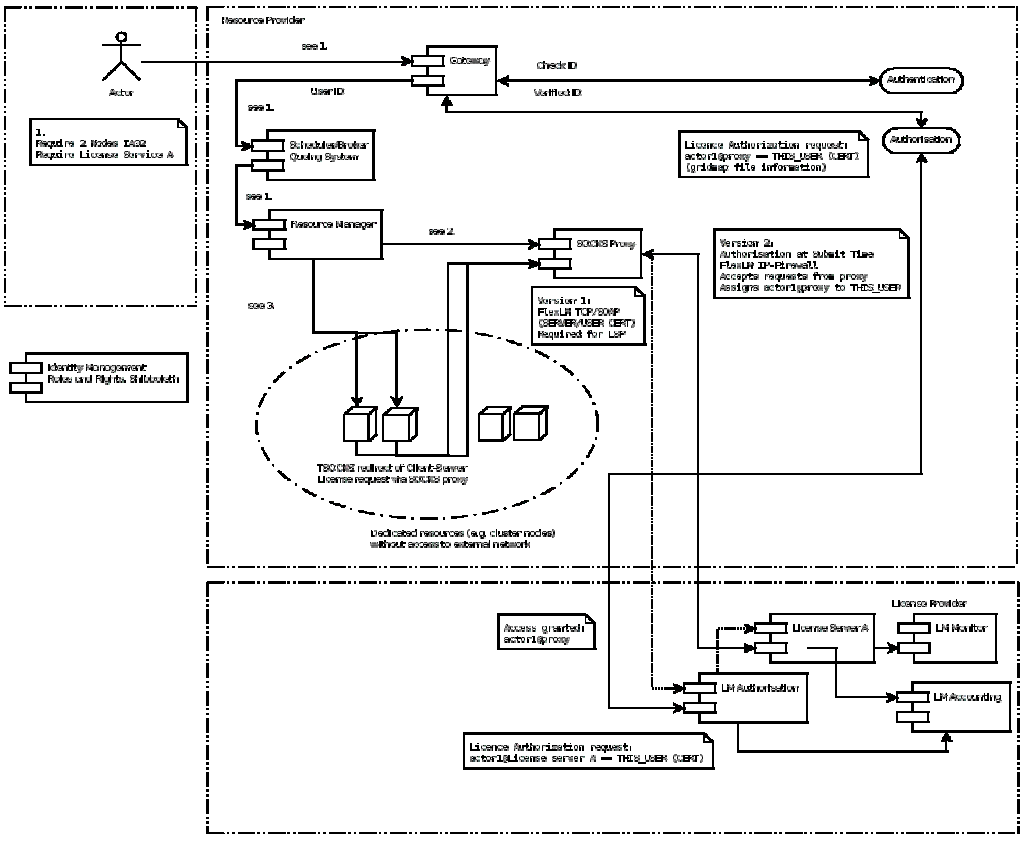

Figure 2:License Management Architecture

Job submission and description are extended with respect to the list of resources (license server) and possibly required authorization credentials. ***License authorization*** then is performed at submit time via a query to a remote service. In a ***first step*** license authorization will be based on a local access list (ACL). Since this latter functionality is frequently required for local usage in a non-Grid context, (e.g. any linux cluster which is shared between different organizations) we will maintain two different versions of license authorization. In a **second step** an optional/alternative run-time authorization (at the remote upstream proxy) via a pin/tan mechanism is implemented. Details of this implementation are presented in Section 4. Usage of licenses is accounted by the ***LM accounting*** module. The ***LM monitor*** monitors the status of available licenses and upon request returns this status to higher-level services like license schedulers/brokers or an SLA monitor.

4. Technology Description

Industrial environments typically rely on commercial applications of ISVs with an associated License Management – usually FlexNet from Acresso [2], which is the de-facto standard in this area. FlexNet has a closed API, is proprietary and based on a simple client-server mechanism. The FlexNet scheme allows 'floating' licenses, which are not bound to a specific host. Rather they are allocated dynamically to arbitrary hosts. Licenses then are checked out at the license server when an application starts and checked in when it ends.

In principle FlexNet therefore is suitable for usage in a Grid environment. There are, however major security and identity issues with respect to the access to the license server in a Grid environment. For example the FlexNet software is able to filter legal and illegal accesses based on the host IP, but is *not* able to grant access on the basis of user/group certificates. This implies that on every Grid site an unauthorized user could check out and use an arbitrary number of licenses once the corresponding license server is exposed.

In order to support this standard (or in general: any client-server license management scheme) we propose to transparently reroute the encrypted socket-based communication between client and server via a SOCKS proxy-chain. The communication from license client then can be transparently forwarded (via a socksified job shell) to a remote upstream proxy and then to the remote license server. The run-time authorization at the upstream proxy is handled via a PIN and associated encrypted one-time passwords. The PIN here represents a license account and can be used to provide the accounting context. License owners (typically institutions) can set up an arbitrary number of these license accounts under a billing account. This mechanism allows institutions or research groups to share access to licenses and to use licenses in a cost unit based accounting context. A self-imposed budget-control for pay-per-use scenarios will be available. The handling of the one-time passwords (generation of tan lists, license accounts and their properties) is implemented as a web service. A design of a portal which enables users to access these web services, conveniently share accounts, automatically extract one-time passwords and submit correspondingly modified jobs is currently in progress.

The local SOCKS proxy (at the service provider site) additionally can be re-configured on a per-job basis by the local resource management systems, thus providing an additional layer of security. The license management architecture is complemented with extended functionalities for job submission and description, license accounting, billing and a License Monitor. Information from FlexNet itself here needs to be synchronized with information from the authorization module in order to provide a complete cost unit based accounting.

5. Developments

5.1 Development

A reference implementation of the license management architecture is currently developed..The implementation of this architecture provides an encapsulation of a standard FlexNet license manager and includes functionalities like authorisation, monitoring and accounting. The technologies that have been used to develop these components are standard Web Services and SOCKS SS5[3] proxies (both local and upstream). The authentication PIN/TAN license scheme is based on OTP Password [4].

5.2 Availability

The "LM-Proxy Standalone" component and the components "LM Job description and Submission for GT4 and Unicore 6" have been released and are available under [5]. The complementing parts of the license management architecture (LM Authorization, LM

Monitor, LM Accounting) will be available by the end of August 2008. All components will be licensed under GPL and will be accessible through [5].

6. Results

A few selected components of this architecture like the stand-alone proxy and the modified job submission for GT4 and Unicore 6 have been implemented. They have been validated and successfully used by business experiments in BEinGRID. The remaining components of the license management architecture (authorization, monitoring and accounting) will permit secure authorization without the need for out-of-band agreements with potential resource providers. It will allow license providers to map users (or rather: their shared accounts) to the license usage and therefore enable license providers to create the actual bills with a detailed cost-unit based accounting. It will allow end-users to split their license budget in terms of cost-units and to restrict the corresponding usage (Budget control).

7. Business Benefits

The license management architecture will allow the use of commercial ISV applications in the Grid. The solution is not bound to any specific middleware. Since it also is a non-interruptive solution, which merely complements currently exiting license mechanisms, it will allow the ISVs to extend their business models with respect to pay-per-use in a fluent transition. This eventually established pay-per-use model then in turn will permit all customers who need to use licensed ISV applications to make cost-efficient use of on-demand Grid infrastructures in order to dramatically speed up their design phase and/or quality assurance, to minimize the risks and/or to improve the quality of their products.

A good example for this need to use commercial ISV applications in computing on-demand grid environments are small engineering companies which occasionally need to satisfy peak demands in the area of simulation.

8. Conclusions

We have succeeded in designing and implementing a novel license management architecture which supports the entire class of client-server based license mechanisms in grid environments. This includes the proprietary FlexNet solution of Acresso[4]. Since this support is a pre-requisite for the use of commercial ISV applications in grid environments, the solution will substantially enlarge the potential grid market size in the area of on-demand computing by industry. The license management architecture also supports the required non-interruptive transition towards a pay-per-use business model for licenses.

The architecture is centered around a dynamically re-configurable SOCKS proxy infrastructure. It is designed to provide excellent scalability. The solution is generic and – with minor modifications (job description and submission) – compatible with all current middlewares. It is also suitable for License Service Provisioning (LSP) in a non-Grid context. Since it provides a very high level of security we think that this solution will play a very significant part in paving the way towards a pay-per-use license model.

References

[1] BEinGRID – Business Experiments in Grid, www.beingrid.eu (as of 08.05.2008)
[2] Acresso Software, www.acresso.com (as of 08.05.2008)
[3] SOCKS SS5, ss5.sourceforge.net (as of 08.05.2008)
[4] PAM_SOTP, www.cavecanen.org/cs/projects/pam_sotp/ (as of 08.05.2008)
[5] LM Architecture, gforge.beingrid.eu (as of 08.05.2008)