

Development of an Information-analytical System for the Support and Maintenance of Licenses at MLIT JINR

E.I. ALEXANDROV, I.N. ALEXANDROV, D.V. BELYAKOV, N.A. DAVYDOVA, L.A. KALMYKOVA,
M.A. LYUBIMOVA, T.F. SAPOZHNIKOVA, T.S. SYRESINA, A.V. YAKOVLEV, P.V. ZRELOV

Joint Institute for Nuclear Research, Dubna, Russia

Abstract. The License Management System (LMS) was developed at the JINR Information Technology Laboratory. The purpose of creating an LMS is to automate the management, acquisition, maintenance and use of licensed software products. The system consists of a network licensing system (NLS), databases and a web interface. NLS provides network license management, collects and transmits to the time series database information about which network license was used by the user and at what time. The features of collecting this type of data are given. This data is used in monitoring implemented on the basis of the Grafana platform. The main LMS database stores data related to corporate, private, and other types of licenses. It also stores the necessary data regarding license users. The database is implemented in PostgreSQL. The system provides the ability to process workflows such as ordering new licenses that users need, ordering to add to the catalog of purchased licenses, and other functions. The LMS web interface is implemented in the development environment of the Electronic Document Management System "EDMS Dubna" using the LegoToolkit web application. A website has been developed and created for LMS users.

Keywords: *license management system, database, web interface.*

DOI: 10.14357/20790279230103

Introduction

The Meshcheryakov Laboratory of Information Technologies (MLIT) of the Joint Institute for Nuclear Research (JINR) is actively working to streamline the licensed software used. There are various aspects of work related to optimizing the use of licenses. For example, choosing the optimal consistent set of licenses for complex open source projects [1]. In our work, we primarily consider the issues of optimizing the use of licensed products in terms of the cost of their acquisition, and for the same purposes, the collection and accounting of information on the use of licenses. The variety of types of purchased licenses creates a potential difficulty in considering and choosing the optimal licensing conditions for the specific needs of the Laboratory. There are many types of licenses, such as personal, local, pre-installed software, full package product, volume licensing, group, network, etc. To select the required types of licenses, it is necessary to have information on both existing licenses and the needs of users. Additional difficulties arise when solving the problem of collecting dynamic information on the use of network licenses. Licenses of this type are pur-

chased for a certain number of users who have the right to simultaneously use the license. The cost of network licenses used in the Laboratory is quite high and varies greatly depending on the number of users it is purchased for. Therefore, detecting the maximum number of concurrent users is an important task for future license renewal. To solve this problem, it is necessary to collect data on the beginning and end of the use of the license by the user. The corresponding license usage and release requests are sent from different user machines, and license managers are third-party products from which it is not always possible to obtain license usage information. Based on this, and also taking into account the high costs of acquiring licensed software products, the obligation to comply with the rules of the licensing policy and the need to plan and optimize the purchase of licensed software, it was proposed to automate the processes of license management. It resulted in the development and most of the implementation of the LMS. The architecture of the system, the database object model schema, the Web Interface design and implementation are presented in this paper below.

1. Architecture of the LMS

The LMS consists of three parts: Network Licensing System (NLS), databases and Web Services.

The general architecture of the LMS is shown in fig. 1.

Important parts of the system are Web services and the associated database implemented in PostgreSQL. The database contains all information on purchased licenses, licensed software product (LSP), LSP owners, main and auxiliary catalogs, LSP development companies, vendors and other necessary information. Based on this information, personal accounts of the user, operator and auditor, the corresponding forms and documents are formed. The monitoring of Network Licenses Usage information is based on data stored in the Influx database (InfluxDB). This data is used directly by the Grafana Server [2] and its plugin implemented within the LMS.

The external, public part of the Web services is a Web site [3] built on the Drupal 9 platform. The site contains up-to-date data on software licenses available in the Laboratory, the terms and conditions of their use, and also provides access to the distributions of server software products. The private part of the Web services is a personal account that allows various categories of users to manage the purchase and distribution of licenses, as well as to make changes to the database.

The NLS is a separate subsystem of the LMS. It consists of a separate network segment with private network addresses and a system of access to them based on the general entry into the pool of virtual machines and the transfer of a request to use a license to the corresponding virtual machine via the ports specified inside the request. This is done in order to protect (secure) the

pool of virtual machines with license managers. Each virtual machine contains the manager of the corresponding LSP and the manager's log analyzer for obtaining information on monitoring the use of the LSP. When installing a software package, the user specifies a single entry point as the name of the license server, and, depending on the selected package, the required port. When the software package is launched on the user's computer, the license server is contacted, a connection is established with a proxy server, which, depending on the incoming ports, forwards the connection to the appropriate virtual machine with the required license manager, and the license is granted. Accordingly, at the end of using the software package, the license is returned to the pool of available licenses. Thus, the developed structure of the licensing system allows organizing a single entry point for users, easily updating versions of the license manager, and also provides a sufficient level of protection against external intrusions. The manager usually writes into a log file all activities concerning granting and releasing licenses, including the time of license occupation and release, as well as the user's name who requested the license. The log file can be used to collect information about license usage. MLIT has network licenses, all of which use the FlexNet Publisher license manager [4]. In any log, one can select keywords to search for the username and time when the license was granted, as well as another entry with the username and time when the license was returned. An example log file for Maple can be seen in fig. 2. The above keywords are "license granted" and "license accepted". The output of such a parse is the same for all license managers. The structure of the InfluxDB time series database is in the center of fig. 2.

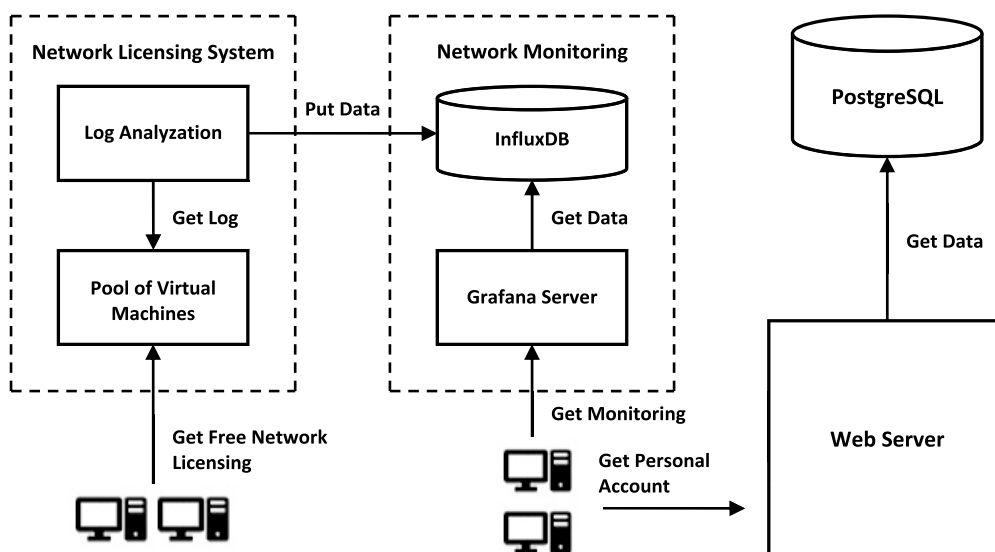


Fig. 1. General architecture of the LMS

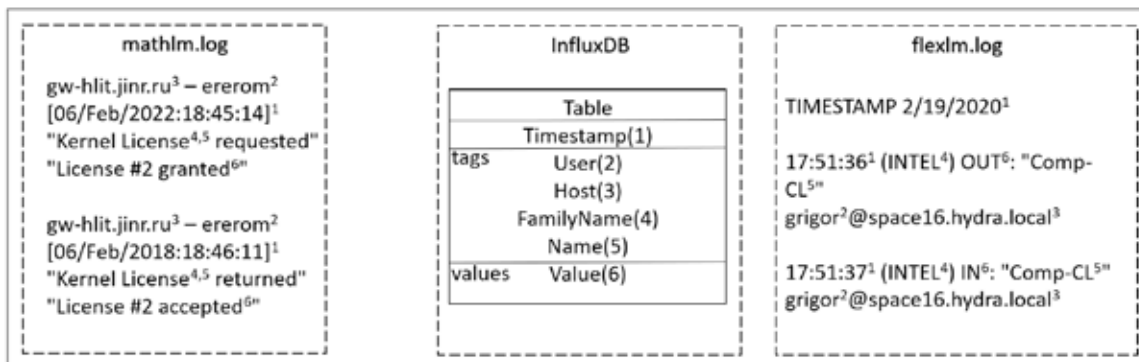


Fig. 2. Structure of an InfluxDB record

InfluxDB records have the following structure: TimeStamp (always required for a time series database), tags, and values. The tag means the string type, and the value means the numeric type of the corresponding parameter. The tag part has the following parameters: User, Host, FamilyName (license family), Name (license name). The value part has only one value: 1 for a license grant operation and 0 for a license release operation. The numbers show the correspondence between the log lines and database entry parameters. The corresponding Log Analyzer runs as a daemon on each virtual machine. It checks the size of the log file, and if the size is changed, it reads the log file, detects the unprocessed part of the log file and parses it finding log records that reflect the user's activity on request or release of licenses. The Log Analyzer writes the parsed data to the InfluxDB. It can parse 2 types of logs: Maple and common flexlm used by Matlab, Comsol, etc. This information is used by the Grafana server to generate network license dashboards at the request of users.

2. Database object model

The developed object model of the LMS is shown in fig. 3. The LMS database keeps permanent data concerning licenses and their usage, users and their roles, software products and companies that produce it, vendors that provide software licenses, as well as other information necessary for the implementation of the project. All information is structured and represented by the corresponding related entities (objects).

The 'license' object keeps parameters that describe a license. The 'license' object stores parameters that describe information about the license, such as the name of the license, vendors contact information, vendor code, license purchase date, who purchase the license, price, and license description. The object also stores important parameters describing the license validity period and other relevant information.

The 'license' object has associations with the 'license_type' object (parameter 'license_type' in use), the 'software' object (parameter 'software_id' in use) and the 'provider' object (parameter 'provider_id' in use). The 'license_type' object has 'license_type_name' parameter that describe the type of the license. There is a set of license types such as Public (information about the license is available to any user of the system), Closed License (license information is available only to those who use it and to special users), Personal License (a license intended for a single user), Local License (a license intended for local installation on a single computer), Group License (a license intended for multiple users or local installation on multiple computers, and the number of computers cannot be limited).

A highly important type is the Network License, which is obtained through a special license server. The network license entitles the user to install the product on several computers, but the number of concurrent users must not exceed the number of purchased licenses. Preinstalled Software (OEM) is also a type of licenses. The user of such software purchases the software together with a personal computer (PC) or a server, and it can only be used on the purchased PC. The last type of licenses used is the Boxes product or Full Package Product. The purchase in this case of one "box" gives permission to use the software product on one computer. Users are allowed to reinstall the product on another PC, but only a certain number of times.

The 'provider' object describes the license provider. The 'software' object represents the licensed software using its name and description. This object has an association with 'company' objects that describe the license owner company. The 'sys_users', 'license_use', 'computer' and 'computer_type' objects describe such essences that are connected with users who use the license and the computer on which it is installed. Only the 'license_use' objects are associated with the 'license' object ('license_id' parameter in use).

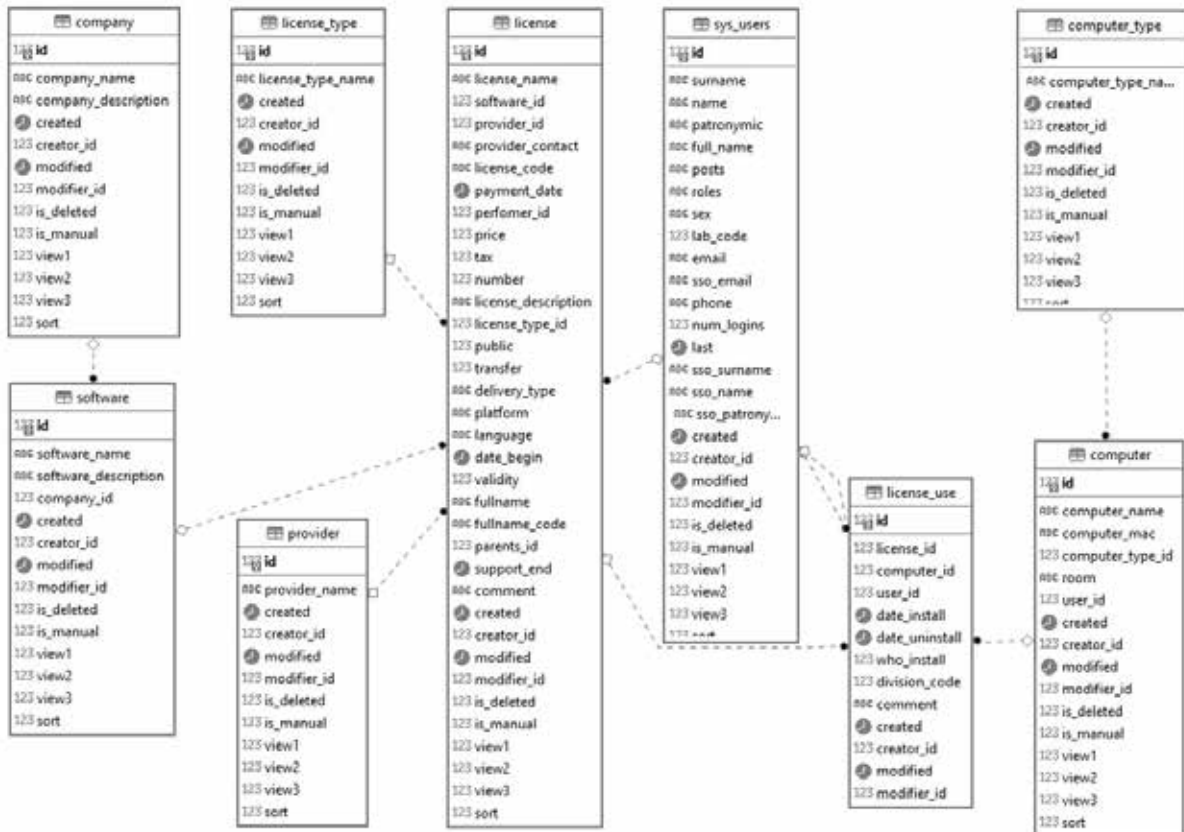


Fig. 3. Object model of the LMS database

The LMS database schema also contains other objects not shown in fig. 3. They provide the functioning of various subsystems of the LMS:

- storage of parameters describing information about users, their roles and their registration in the system;
- functioning of the subsystem “Reference books”;
- integration with other JINR information services, in particular, with “EDMS Dubna” [5].

The main database is implemented on the PostgreSQL platform.

The database that keeps monitoring data is a separate database. This is not an SQL database, and it keeps such data as time series with data in the InfluxDB for Grafana.

3. Web interface design and implementation

The Web interface is designed as a set of personal accounts (PA), which differ depending on user rights or roles. The JINR single sign-on (SSO) is used to authenticate the user and his role when entering the personal account of the LMS.

The LMS defines four roles: user, auditor, operator and administrator. The personal account view depends on the role of the client. The user with

the user role can see information about all public licenses, any information about own licenses and their expiration period, as well as receive notifications about new licenses. The user can request a new license from the list of existing licenses or a renewal of the old one. The user may request that a new license be added to the catalog of available software for purchasing licenses. The operator has all user rights and can additionally enter into the DB all data concerning licenses, their owners and other information that should be presented in the LMS database. The auditor, additionally to the operator’s rights, can also see any information concerning any licenses, including their cost and any analytical information. The administrator performs any actions available in the LMS database. The administrator has the same rights as the auditor, but with additional purely administrative capabilities. The view of the auditor’s personal account is presented in fig. 4. The user account has a set of TABs at the top of the Web page, and any TAB can also have a set of SUBTABs below the corresponding TAB.

The implementation of the context of tabs and subtabs for different types of user roles is the same with rare exceptions (nested tabs may differ in the

presence of some buttons), thus, the pages differ mainly only in the presence of certain tabs and subtabs.

The view of user personal licenses in the user's personal account page is shown in fig. 5 as an example of the subtab context.

Work to exchange information with other JINR information services, such as "EDMS Dubna", is also underway.

The LMS Web interface is implemented in the development environment of the electronic document

management system "EDMS Dubna" using the entire Web Application LegoToolkit (WALT) [6]. This toolkit was developed at MLIT JINR and has been successfully used in other systems for a long time. WALT is a template-oriented platform designed for the development of Web applications of various degrees of complexity, and the main idea of WALT is to provide transparent, extensible, and modifiable tools for solving some specific problems that arise when developing Web applications. The use of WALT for the LMS proved to be highly effective.

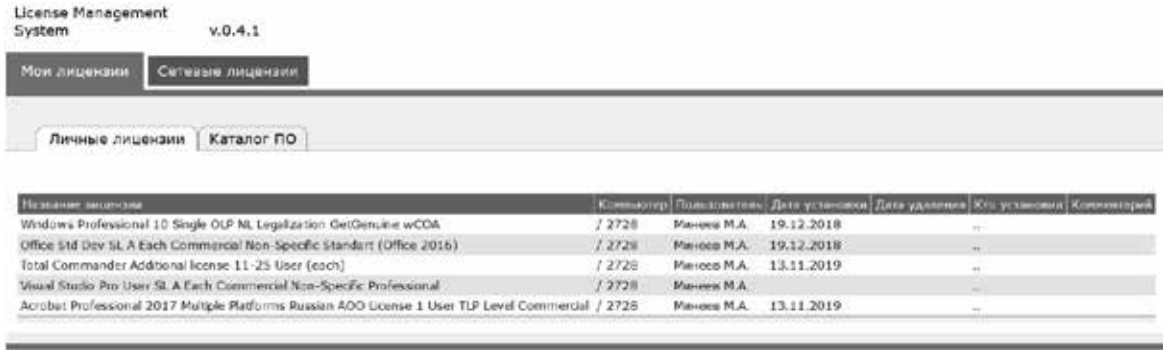


Fig. 4. View of the page of the auditor's personal account



Fig. 5. View of user personal licenses in the user's personal account

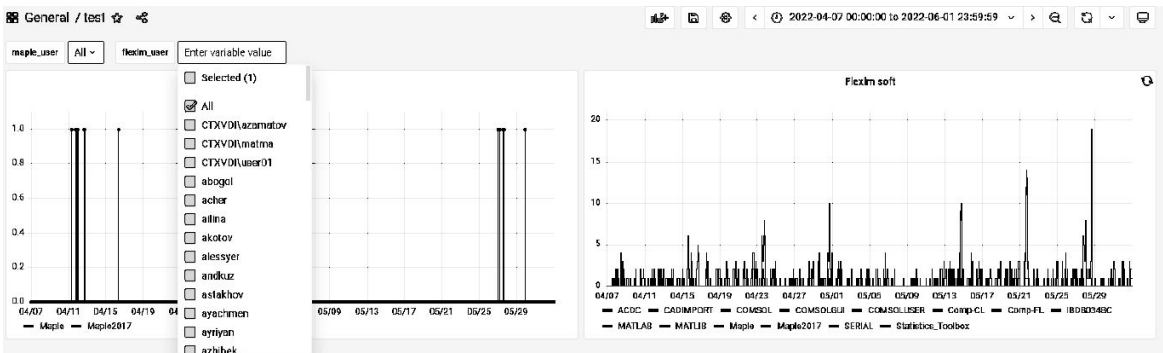


Fig. 6. View of the network license monitoring page

Monitoring the use of network licenses is the basis for analyzing the use of network licenses and the possibility of obtaining a cheaper set of network licenses in the future. The view of the monitoring page is illustrated in fig. 6. The left graph in fig. 6 shows the number of network licenses that users use for Maple. The use of Combobox allows the viewer to set a list of watched users. The right graph shows the number of network licenses that users use for Comsol and Matlab. These programs have several components with separate licenses for each.

The monitoring page can be used as a separate Web page and as the content of a tab of any user's personal account.

The initial version of the LMS release was implemented and is presented above. The current version implementation does not include some workflows, such as ordering for supplementing the catalog of purchased licenses or notification concerning license expiration. Work to develop the system according to the design is underway.

Conclusion

The architecture and object model of the LMS database were developed. The design of the Web interface of the system was completed. The initial release of the LMS was implemented. The system consists of the NLS, Web services and databases. The NLS is a separate LMS subsystem responsible for automatically issuing and releasing network licenses, as well as is a pool of virtual machines, each of which runs the appropriate license manager and log analyzer, supplying monitoring information to the Influx database. Monitoring data is visualized using a plugin for the Grafana server. The Web services consist of the external public part, which contains up-to-date data on software licenses and other corresponding data, and the private part of the services containing personal user accounts. The next version of the implementation is under construction.

References

1. *Pogrebnoy, Dmitry, Ivan Kuznetsov, Yaroslav Golubev, Vladislav Tankov, Timofey Bryksin.* Sorrel: an IDE plugin for managing licenses and detecting license incompatibilities // 2021 IEEE International Conference on Software Maintenance and Evolution (ICSME) [Rules for the citing of sources]. Available at: <https://arxiv.org/pdf/2107.13315.pdf>.
2. Grafana Home page [Rules for the citing of sources]. Available at: <https://grafana.com/>.
3. Licensed software in MLIT [Rules for the citing of sources]. Available at: <http://soft-lit.jinr.ru>.
4. FlexNET Publisher [Rules for the citing of sources]. Available at: https://docs.revenera.com/fnp/2021r2/pdf/fnp_LicAdmin.pdf.
5. *Alexandrov, I.N., O.V. Belyakova, V.V. Korenkov, S.V. Kuniaev, L.N. Pechnikova, M.S. Plyashkevich, S.V. Semashko, G.V. Trubnikov, P.V. Ustenko, S.N. Chikhalina, A.V. Yakovlev.* Development and implementation of electronic document management system "EDMS Dubna" at JINR // CEUR Workshop Proceedings. 2016. Vol. 1787. 97-102 [Rules for the citing of sources]. Available at: <http://ceur-ws.org/Vol-1787/97-102-paper-15.pdf>.
6. *Korenkov, V.V., S.V. Kuniaev, S.V. Semashko, I.A. Sokolov.* WALT platform for Web Application development // Proceedings of the 9th International Conference "Distributed Computing and Grid Technologies in Science and Education" (GRID'2021), Dubna, Russia, July 5-9, 2021. 2021. Vol. 3041. 387-392 [Rules for the citing of sources]. Available at: <http://ceur-ws.org/Vol-3041/387-392-paper-71.pdf>.

Alexandrov E.I. Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: aleksand@jinr.ru.

Alexandrov I.N. Phd, Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: alexand@jinr.ru.

Belyakov D.V. Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: dmitry@jinr.ru.

Davydova N.A. Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: luna@jinr.ru

Kalmykova L.A. Junior Researcher, Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: lidak@jinr.ru (correspondent author)

Lyubimova M.A. Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: lma@jinr.ru

Sapozhnikova T.F. PhD, Junior Researcher, Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: tsap@jinr.ru

Syresina T.S. Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: syresina@jinr.ru

Yakovlev A.V. Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: yakovleva@jinr.ru

Zrelov P.V. PhD, Joint Institute for Nuclear Research (JINR), 6 Joliot-Curie St, Dubna, Moscow Region, 141980, Russia, e-mail: zrelov@jinr.ru