



The CMS Electronic Logbook



G. Bauer (6), B. Beccati (2), U. Behrens (1), K. Biery (5), J. Branson (4), S. Bukowiec* (2), E. Cano (2), H. Cheung (5), M. Ciganek (2), S. Cittolin (2), J. A. Coarasa (2), C. Deldicque (2), S. Erhan (3), D. Gigi (2), F. Glege (2), R. Gomez-Reino (2), D. Hatton (1), Y. L. Hwang (2), C. Loizides (6), F. Ma (6), L. Masetti (2), F. Meijers (2), E. Meschi (2), A. Meyer (1), R. K. Mommsen (5), R. Moser (2), V. O'Dell (5), L. Orsini (2), C. Paus (6), A. Petrucci (a)(4), M. Pieri (4), A. Racz (2), O. Raginel (6), H. Sakulin (2), M. Sani (4), P. Schieferdecker (b)(2), C. Schwick (2), D. Shpakov (5), M. Simon (2), K. Sumorok (6), A. S. Yoon (6)

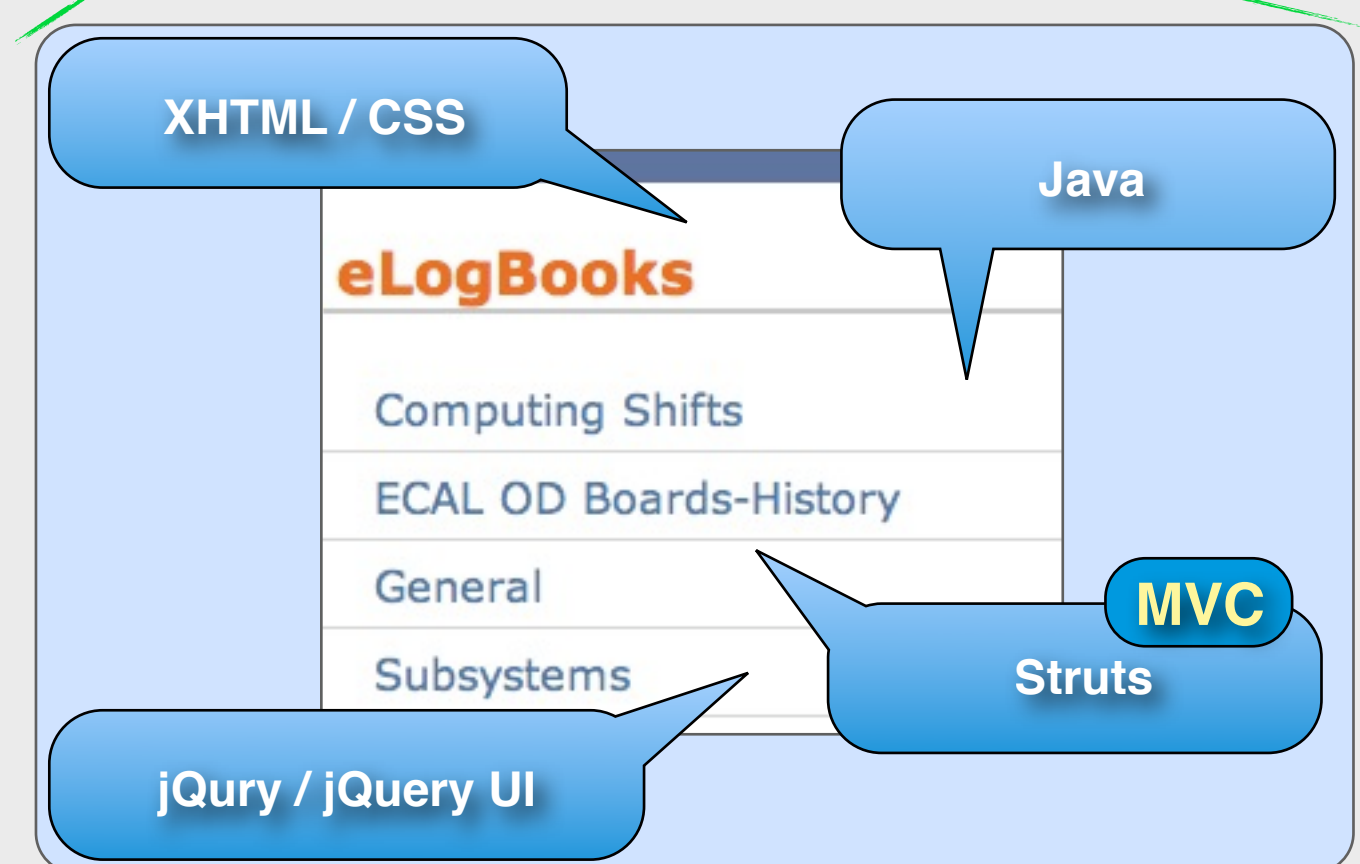
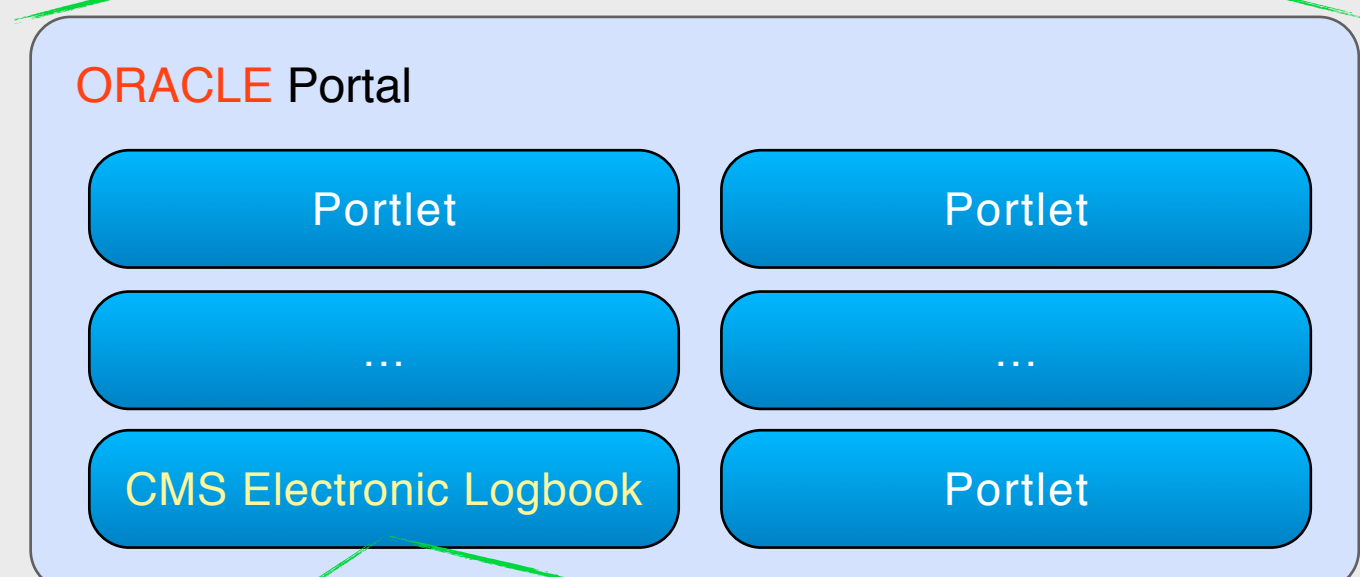
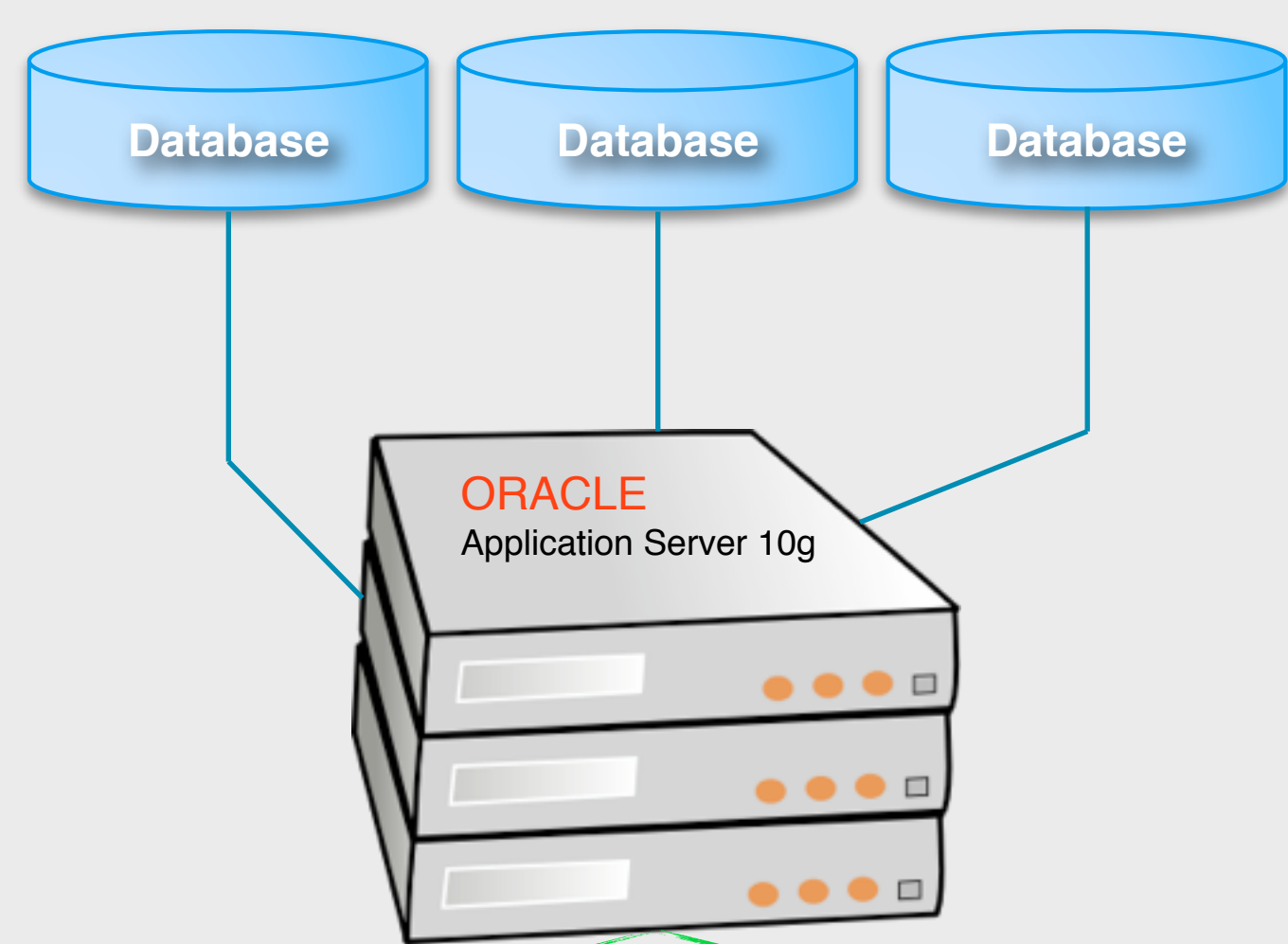
Introduction

The CMS ELogbook (ELog) is a collaborative tool, which provides a platform to share and store information about various events or problems occurring in the Compact Muon Solenoid (CMS) [1] experiment at CERN during data taking.

The ELog is based on a Model-View-Controller (MVC) [2] software architectural pattern and uses an ORACLE database [3] to store messages and attachments.

The ELog is developed as a pluggable web component in ORACLE Portal [4] in order to provide better management, monitoring and security.

Architecture



The ELog is a portlet running on ORACLE Portal (see Figure 2) that provides implementation of the portlet container [5][6]. A portlet is a dynamic, reusable, pluggable web component that can draw content from many different sources. As different portlets can be placed on a common page, the user receives a single-source experience. The ORACLE Portal has been chosen because it is scalable, secure and gives the possibility to monitor and manage all the data and users.

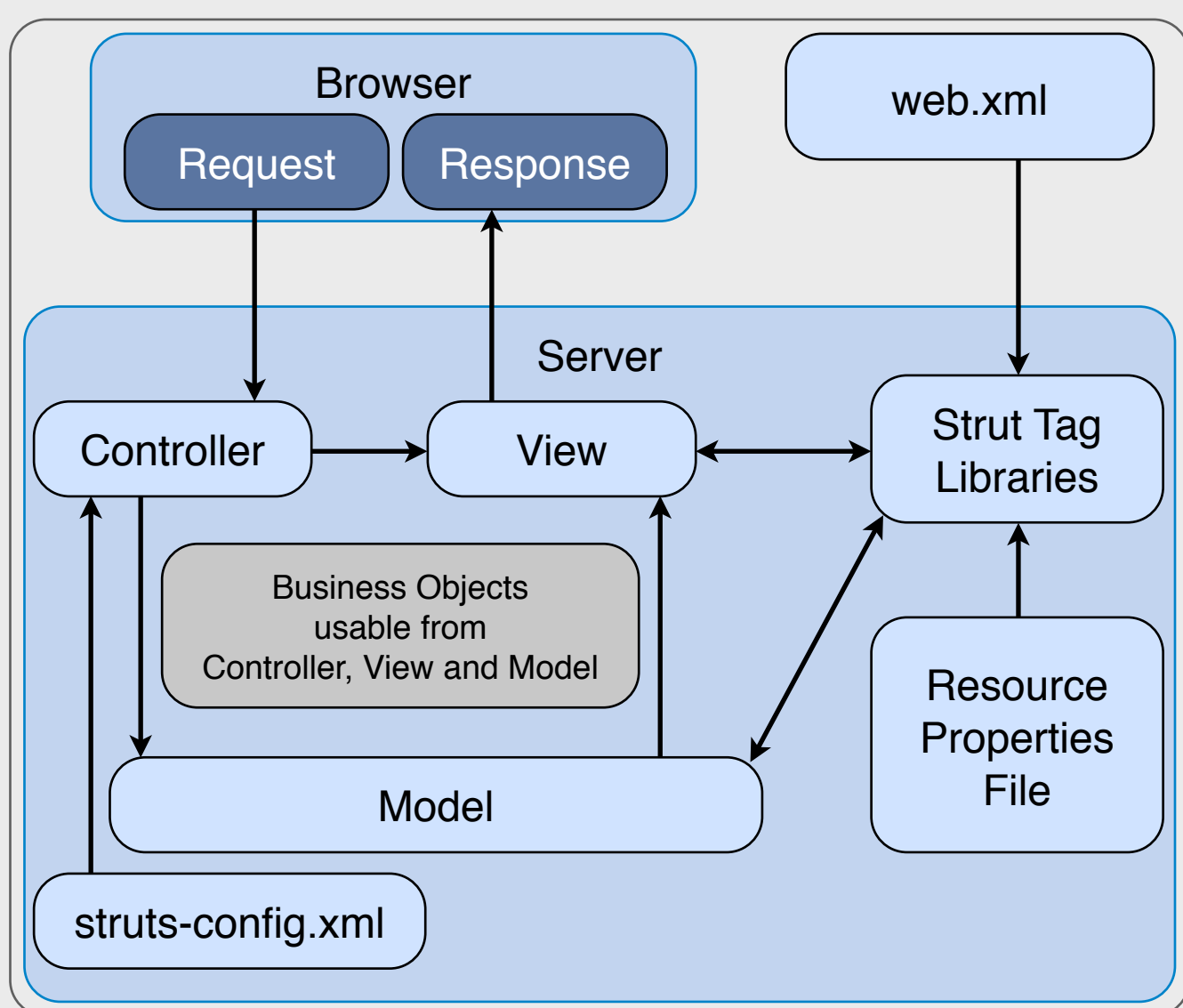


Figure 2. Architecture overview

Figure 3. Struts flow diagram

By applying the Model-View-Controller architecture (see Figure 3) to the ELog, core business model functionality was separated from the presentation and application logic. The Model represents the business or database code, the View represents the page design code and the Controller represents the navigational code. Such separation allows multiple views, gives clarity of the design and efficient modularity that makes developing, testing and maintaining easier.

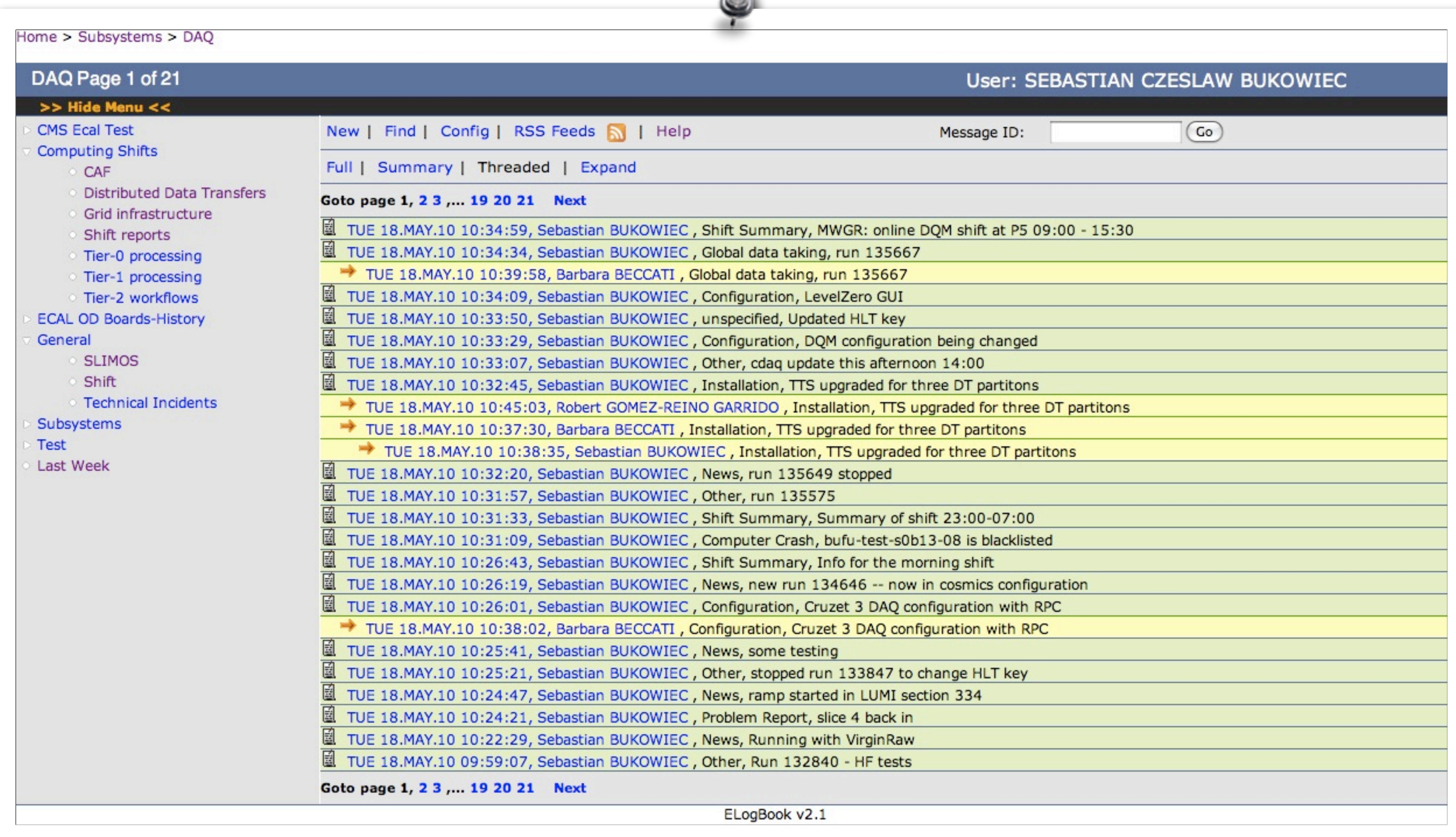


Figure 1. The CMS ELogbook, threaded view

Features

The ELog organizes messages into categories, where each category may have unlimited levels of other categories and subcategories. Each subcategory has its own page where entries are displayed as a list. Four views are available for browsing: Full, Summary, Threaded and Expanded (see Figure 1). The user can navigate page by page through the messages in the current subcategory or display together in a special view all chosen subcategories in the selected time period (day, week, month), see Figure 5. This view makes use of a special personalization mechanism provided by ORACLE Portal allowing chosen subcategories to be saved in the profile of the user. The mechanism is similar to the cookie mechanism in web browsers with the difference that information is kept on the server side and is available for all browsers to which the user is logged in. The user can create a new message that can have an attachments with optional description. Two editors are available: rich and plain text editor.

The editing functionality allows only the author to edit the existing messages, while the replay functionality gives users the possibility to reply to any existing message inside a thread. The ELog also provides the option to subscribe to chosen subcategories. Any update (new or edited message) in the subcategories is sent to the subscribers via email. It also features a search mechanism, where messages from different subcategories can be retrieved on the base of given filters. Additionally the ELog contains RSS Feeds and is open for future extensions.

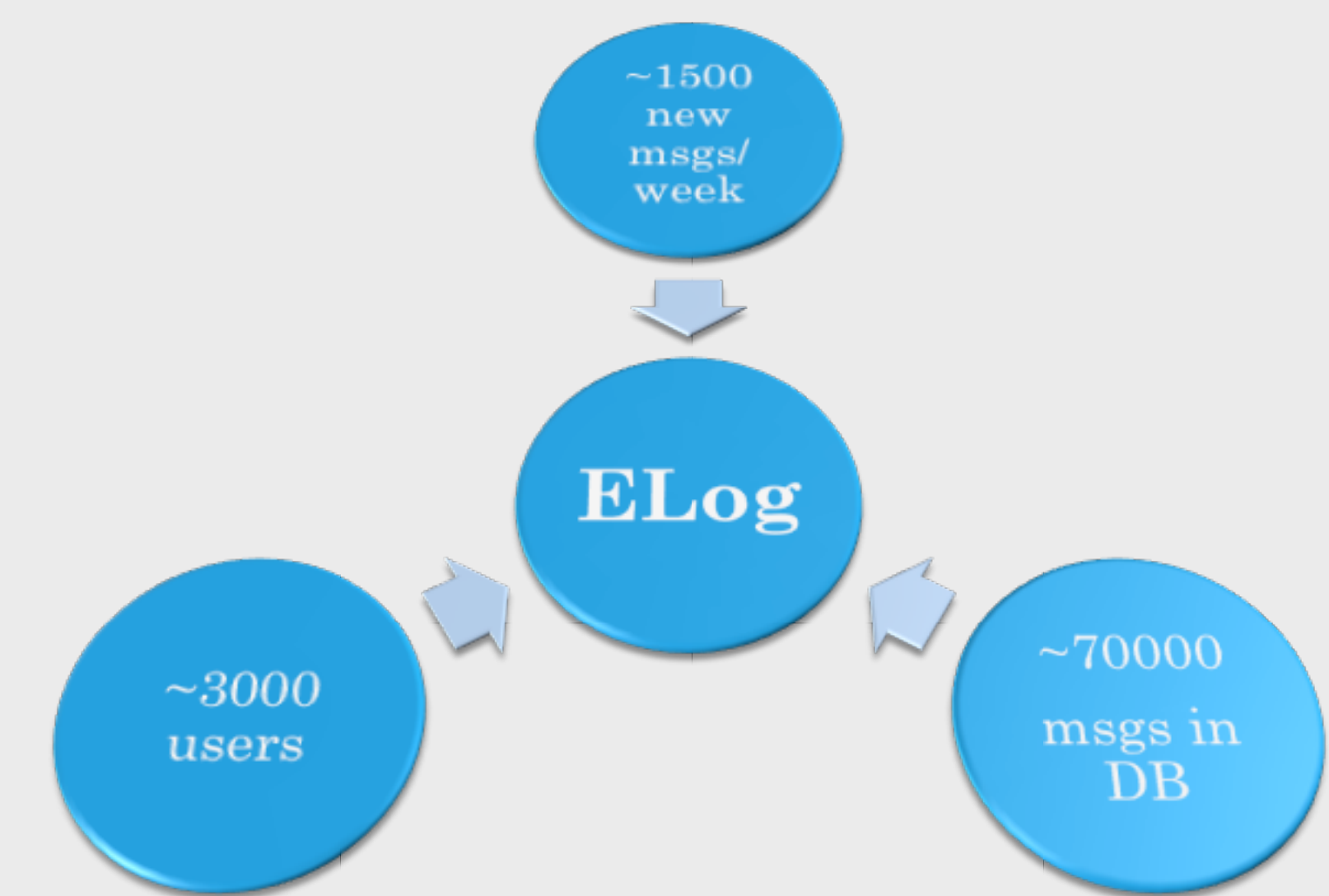


Figure 4. The CMS Elogbook usage statistics

Conclusions

The ELog is extensively used by the entire CMS collaboration (see Figure 4), more than 3000 scientists and engineers from 38 countries, as a tool for communicating and archiving information related to the commissioning and running of the experiment. The tool is continually being improved in order to meet the dynamically changing needs of users. The main goal of developers is to prepare a reliable tool with a modern interface that allows users to trivially find stored information based on a wide range of search options.

References

1. CMS, <http://cms.cern.ch>
2. Struts, MVC architecture, <http://struts.apache.org>
3. Oracle Database 10g, <http://www.oracle.com/technology/products/database/oracle10g/index.html>
4. OracleAS Portal 10g, <http://www.oracle.com/technology/products/ias/portal/10g1014.html>
5. Java Portlet Specification (JSR 168), <http://www.jcp.org/en/jsr/detail?id=168>
6. OracleAS Portal 10g R2: Build Portlets with Java, Volume 1, Student Guide
7. ELOG Project, Stefan Ritt, <https://midas.psi.ch/elog>

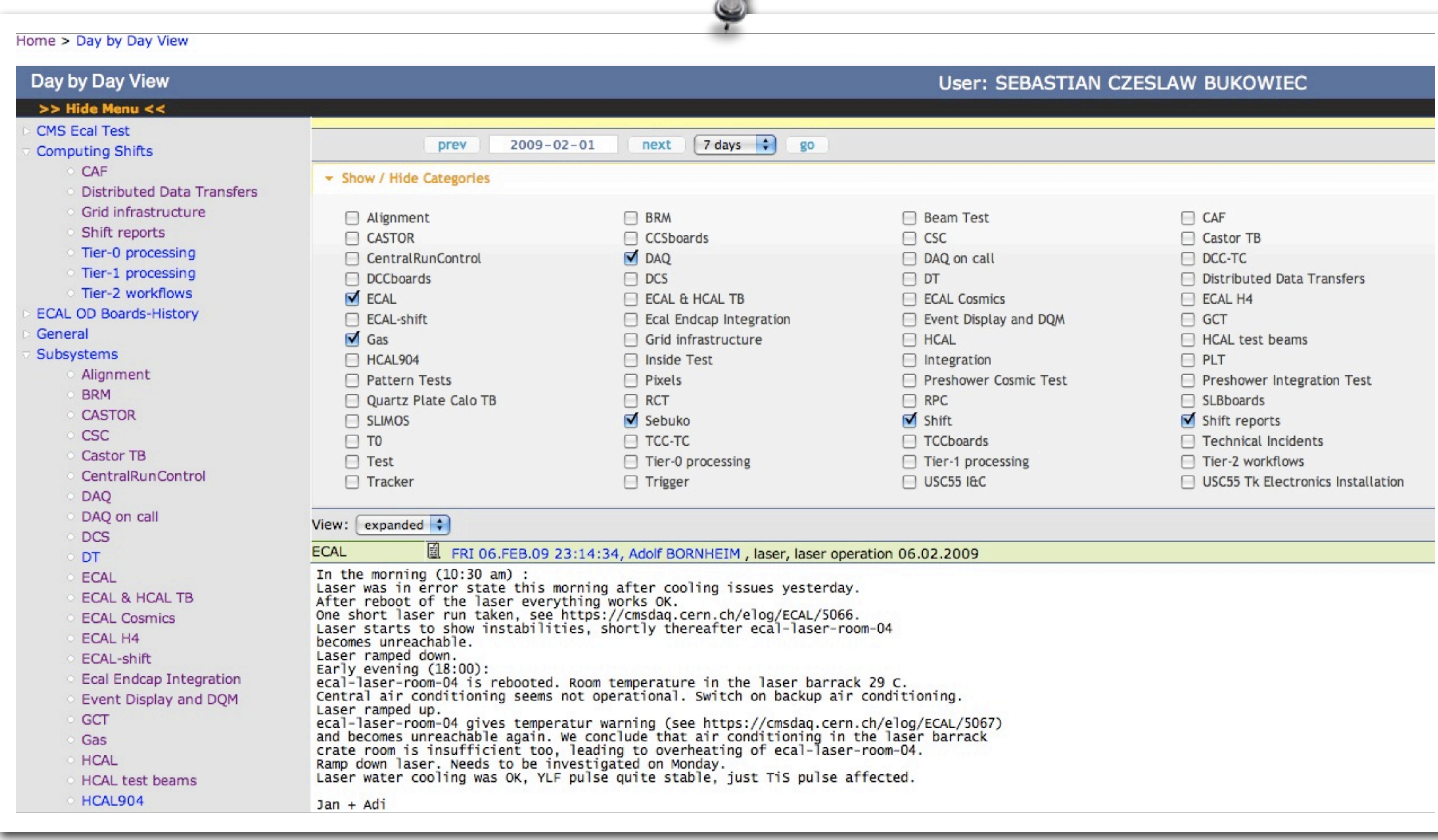


Figure 5. The CMS ELogbook, Day by Day view with expanded messages and menu for choosing subcategories to display

* Contact: Sebastian Bukowiec, Sebastian.Bukowiec@cern.ch, Tel: +41227678086, Fax: +41227668124

This research project has been supported by a Marie Curie Early Initial Training Network Fellowship of the European Community's Seventh Framework Programme under contract number (PITN-GA-2008-211801-ACEOLE). This work was supported in part by the DOE and NSF (USA) and the Marie Curie Program.

1. DESY, Hamburg, Germany; 2. CERN, Geneva, Switzerland; 3. University of California, Los Angeles, Los Angeles, California, USA; 4. University of California, San Diego, San Diego, California, USA; 5. FNAL, Chicago, Illinois, USA; 6. Massachusetts Institute of Technology, Cambridge, Massachusetts, USA; a) Now at CERN; b) Now at Universitaet Karlsruhe