Master Thesis Topic Proposal

Student: Daniel J. Szabo
Student ID: 0939907
Date: 01/27/2010

Design and Implementation of a Role Based Access Control for an Enterprise Wiki built on J2EE Technologies using the Grails Framework as backend and Adobe Flex as Administration frontend

This Master Thesis continues the work on an Open Source Enterprise Wiki, which is maintained by the *University of Applied Sciences Esslingen*[1]. The difference to a common Wiki is that an Enterprise Wiki is used for the documentation of Enterprise projects. Due to the individual requirements for a project and the mixture of different sources of contribution that not always may gain knowledge exceeding their tasks, an Enterprise Wiki has to be much more flexible and configurable in the way of managing the contents. In contrast to all the available Open Source Wiki Software on the market, this bases on different philosophies. The common Open Source Wiki Software, such as the *MediaWiki*[2] project, which is used for the most popular Wiki[3], is about sharing all information without restrictions. Several commercial solutions exist, which provide Enterprise Wiki software. Currently the market leader is *Atlassian* with its product *Confluence*[4]. The Enterprise Wiki has to support the definition of rights for the access to the contents and resources that are saved into the Wiki. A distinction is made between the different projects, so called spaces, and the content of these spaces, the pages of the Wiki. Therefore a Role Based Access Control should be implemented, which provides the possibility to restrict access to the different contents and functionalities of the Enterprise Wiki application. Some default user roles have to be foreseen, such as the default user, who is an anonymous guest. Furthermore different Administrator hierarchies have to be considered. For example Super-Administrators may have all the rights for the application, but Space-Administrators may only have rights on administering the Project, they are responsible for. Additionally to the default roles new roles should be created by the corresponding Administrators. A problem, which is expected to be solved, is that currently available standard security frameworks for Java Enterprise Technologies, such as *Spring Security*[5] or *JSecurity*[6] make a distinction on controlling the access to a certain web

[1] See http://www.hs-esslingen.de
[2] See http://www.mediawiki.org/wiki/MediaWiki
[3] See http://www.wikipedia.org
[4] See http://www.atlassian.com/software/confluence/
[5] See http://static.springsource.org/spring-security/site/index.html

page either allowing all or nothing to be viewed by the corresponding user related to the roles he does impose. This means that a sophisticated distinction on which content or functionality of a web page may be shown to a user is not supported. Since the Enterprise Wiki should support this approach, the standard security frameworks are likely to be insufficient to fulfil these requirements.

The project is still in an early development phase. No Access Control is implemented currently. The Enterprise Wiki is a Java Web Application using a J2EE backend, which is built using the *Grails Framework*[7]. This framework again is built using the scripting language *Groovy*[8], which is implemented on a layer on top of the Java language. Both, Groovy and Grails are considered to be powerful development tools still themselves being developed, but growing in stability, functionality and their users community as well. The backend uses the *Apache Jackrabbit Repository*[9], which is a content repository for Java Technology specified in *JSR 170*[10] and *JSR 283*[11]. The Jackrabbit repository is used for the version control of any contents of the Enterprise Wiki and was evaluated by a preceding diploma thesis.

The administration of the Role Based Access Control, as well as the roles themselves is planned to use the *Adobe Flex Framework*[12] on the client-side of the Web Application. The Flex Framework currently provides the state of the art possibility to create Rich Internet Applications that are executed by Web browsers using Adobes Flash Plugin. Connections to different server backends are supported. At the moment the Grails support for Flex is in an experimental state, therefore analysis has to be done on how to connect the client frontend to the server backend. Though currently different solutions may be considered, it is expected that a stable support by the Grails community is provided in the future. This has to be reviewed in more detail, if this Master Thesis Topic Proposal is accepted to be dealt with.

[6] See http://www.jsecurity.org
[7] See http://www.grails.org
[8] See http://groovy.codehaus.org
[9] See http://jackrabbit.apache.org
[10] See http://jcp.org/en/jsr/detail?id=170
[11] See http://jcp.org/en/jsr/detail?id=283
[12] See http://www.adobe.com/de/products/flex/