SHORT OVERVIEW ABOUT PORTLET TECHNOLOGY FOR REALIZATION OF INTERACTIVE BROADCAST PLATFORMS

Axel Doussier

IT- Departrment
LTU Lufttransport-Unternehmen GmbH,
Duesseldorf, Germany
Email: axel.doussier@arcor.de

KEYWORDS: Open Source, Liferay, Hibernate, Model View Controller, Apache Struts

ABSTRACT

In this paper a rough overview about the main realization aspects of an interactive WEB 2.0 media platform is given. The main issues related to the technical objectives, the user demands as well as the chosen open source environment are shortly overviewed.

INTRODUCTION

Establishing new media platforms needs the requirements definition to finally design the targeted platform. In the first step the services to be applied and their functionality have to be defined on a conceptional level.

The most effective way to create media platforms is to set up on standard WEB 2.0 products /O'Reillynet-2008/. In this case the software product needs to cover most of the technical and functional requirements.

Media Broadcast platforms are not just streaming platforms. Effective media broadcast platforms need to integrate personal functionality improving the customer loyalty as well. Accordingly to this the customer binding has to be realized by offering comprehensive services and support functionalities. To fulfill this demand the broadcast platform is splitted into different areas with different user rights:

- **Global functionality**:
  any user is able to use services like viewing online TV, gathering general Information , wikis, blogs or buying products advertised on the pages

- **Community functionality**:
  registered users will be able to use some social services like dashboards, polls, personalized image galleries and/or public video pools

- **Personal functionality**:
  Special features (fee required) like access to personal video pools, video on demand or special ratings in case of purchasing acts

Most of these Portal services will be offered as global functionality.

PLATFORM INFRASTRUCTURE

Based on the conceptional definition of the requirements (e.g. given in /O'Reillynet-2008/) the decision is made about the infrastructure used. The technical requirements for realizing a media broadcast characterized by the objectives given above should focus the view to the following main topics:

- Data sources
- Services (SOA)
- Platform scalability
- Numerousness services ready to use

In fact the Liferay portal /Liferay-2008/ gives inherently a comprehensive fulfillment of these topics. In the Liferay Portal several services are implemented, like:

- Blogs
- Blogs Aggregator
- Chat
- Message Boards
- Polls
- Wiki
- Page Comments
- Page Ratings
- Bookmarks

For realization purposes the Liferay portal software package has been chosen due to the following reasons:

Liferay Portal /Liferay-2008/ supports all major application servers, databases and operating systems.

- „Liferay Portal complies with key industry standards
- „A highly granular permissioning system allows the user customization considering the user experience at the organizational and personal level.

Several aspects concerning the Enterprise Architecture are fulfilled /Liferay-2008-2/:

- Service Oriented Architecture (SOA) - Liferay uses SOA design principles throughout and provides the tools and framework to extend SOA to other enterprise applications
- The ServiceMix enterprise service bus (ESB) allows applications and 3services to be added quickly to an enterprise's infrastructure.

- Support for Web Services makes it easy for different applications to communicate with each other. Java, .NET, and proprietary applications can work together easily because Web Services use XML standards2 .
- Support for REST-style JSON Web Services for lightweight, maintainable code and to support AJAX-based user interfaces
- Single Sign On – Liferay offers customizable single sign-on with Yale CAS, JAAS, LDAP, Netegrity, Microsoft Exchange, and more. Yale CAS integration is offered out of the box2 .
- High Availability – Maintain zero down time for business critical applications with Hardware/Software Load Balancing, HTTP Failover, Session Replication, and Distributed Cache (using Lightweight Multicast Protocol)2 .
- Dynamic Virtual Hosting – Granting individual community members their own page with a user-defined friendly URL2 .

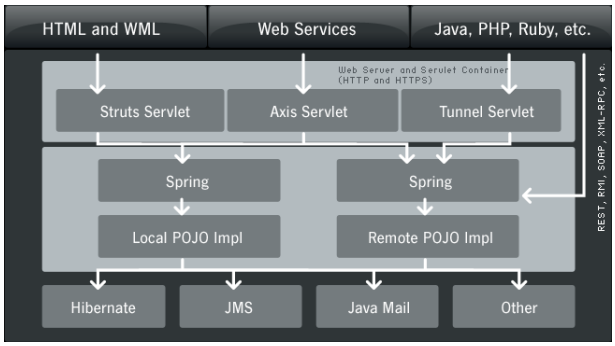

Figure 1: Liferay architecture

The Liferay Portal also includes a lot of technologies like „Hibernate“, „Struts“. In fact this an enormous advantage, because no proprietary technology is used.

Hibernate is an object-relational mapping (ORM) library for the Java language, providing a framework for mapping an object-oriented domain model to a traditional relational database. Hibernate solves Object-Relational impedance mismatch problems by replacing direct persistence-related database accesses with high-level object handling functions.

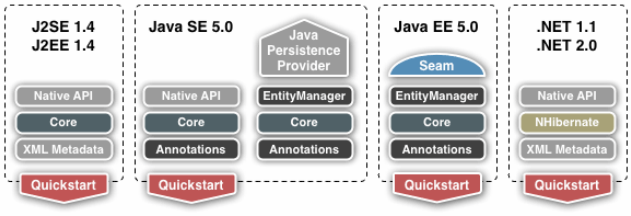

Figure 2: Hibernate modules

Apache Struts is an open-source web application framework for developing Java EE web applications. It uses and extends the Java Servlet API to encourage developers to adopt a model-view-controller (MVC) architecture. It was originally created by Craig McClanahan and donated to the Apache Foundation in May, 2000. Formerly located under the Apache Jakarta Project and known as Jakarta Struts, it became a top level Apache project in 2005.

It is common to split an application into separate layers: presentation (UI), domain logic, and data access. In MVC the presentation layer is further separated into view and controller. MVC encompasses more of the architecture of an application than is typical for a design pattern.

Within the ongoing developments the paradigma of a Model-view-controller is used, devided into a Model and a View:

Model: The domain-specific representation of the information on which the application operates. Domain logic adds meaning to raw data (e.g., shipping charges for shopping cart items).Many applications use a persistent storage mechanism (such as a database) to store data. MVC does not specifically mention the data access layer because it is understood to be underneath or encapsulated by the Model.

View: Renders the model into a form suitable for interaction, typically a user interface element. Multiple views can exist for a single model for different purposes.

Controller: Processes and responds to events, typically user actions, and may invoke changes on the model.

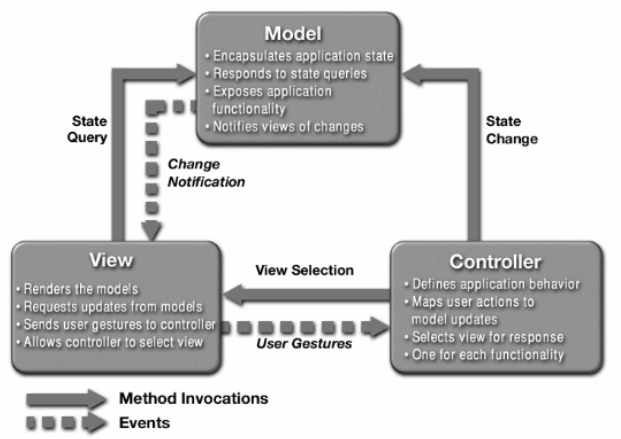

Figure 3: The model view controller paradigm used within this development /Rebag-2008/

REFERENCES

/O'Reillynet-2008/ http://www.oreillynet.com/pub/a/oreilly/tim/news/2005/09/30/what-is-web-20.html, accessed March 2008

/Liferay-2008-1/ http://www.liferay.com, accessed March 2008,

/Liferay-2008-2/ http://www.liferay.com/web/guest/community/tech_specs , accessed March 2008

/Rebag-2008/ http://www.rebag.it/rebag/index.php/Rebag_Ware#J2EE accessed March 2008