

# Freeware App Server and Opensource ORM Proven in Major Hollywood Studio Application

Author: Tim Stapleton, CEO, Beyond Browsing

**Freeware App Server and Opensource ORM Proven in Major Hollywood Studio Application.....1**  
 Author: Tim Stapleton, CEO, Beyond Browsing.....1

**Table of Contents**

**Production Management System Delivers Agility, Performance Gain with POJOs and ORM, Replaces EJBs.....3**  
 The Previous Architecture.....3  
 An EJB-Based System.....3  
 The Solution.....4  
 New Architecture.....4  
 Migration.....5  
 Results.....5  
 Conclusion.....6  
 Beyond Browsing Overview.....7  
 Versant Overview.....7

**Summary**

Beyond Browsing, Inc. has worked with a large Hollywood Studio to develop a very large animated movie production system. The system includes about 1000 rendering servers and about 300 creative workstations. The newest version of this system uses a POJO based architecture, replacing an EJB-based architecture to deliver greatly increased agility, reducing the compile plus deployment time to 17 seconds from about 8 minutes and reducing the number of classes to about 25% of the original count. At the same time, the new system delivers better performance.

## Production Management System Delivers Agility, Performance Gain with POJOs and ORM, Replaces EJBs

Beyond Browsing, Inc. has worked with a large Hollywood Studio to develop a very large animated movie production system. The newest version of this system uses Versant's Open Access object relational mapping product with Beyond Browsing's OccamJ app server to deliver dramatically improved agility and better performance, replacing EJB-based architecture with a plain old Java object (POJO) architecture.

Animated movie production is a complex operation involving hundreds of artists and production staff and thousands of computers. The production process typically takes two to four years. The computing environment supporting this process is highly complex, employing various software tools for compositing, rendering, and editing, combined with custom process tools typically written in Perl, Python, or C-shell. The production management system tracks and routes the pipeline of production work. Updates to the production management system are triggered by user-input and by inter-process input from the rendering and production tool systems. The production management system manages five or more movies in various stages of production simultaneously and the production of each movie can involve over 250,000 individual tasks.

### The Previous Architecture

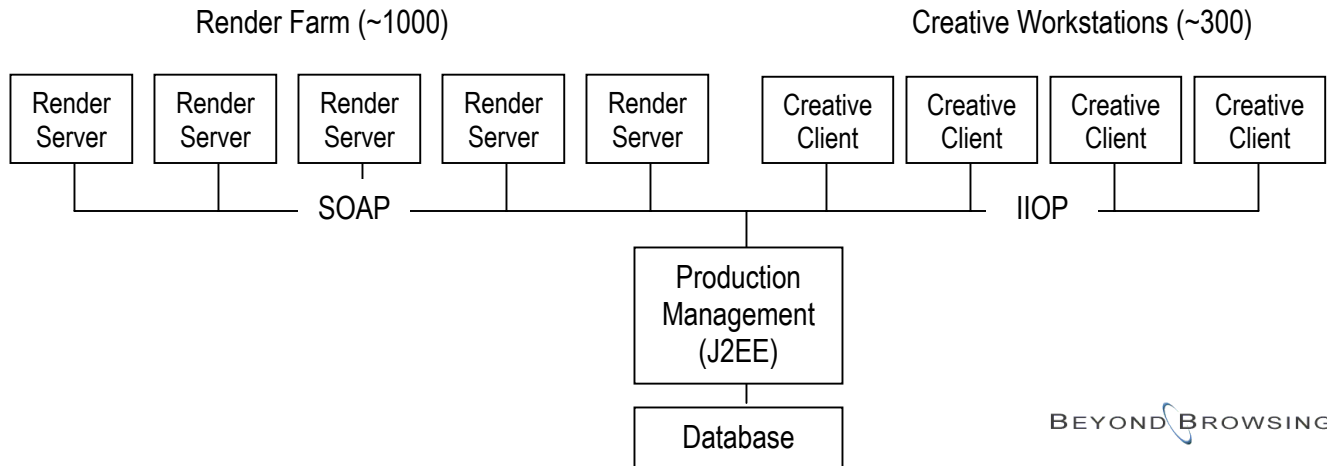
The architecture of the previous production management system was implemented in EJB 2.0 (Enterprise Java Beans) on a well-known J2EE application server. About 1000 rendering servers, which are PCs running Perl, Python, and C++ render tools, are supported by the production management system. Communication with the rendering servers is via SOAP. The creative clients are used by artists and production management staff and run custom applications written in Java with a Swing user interface or Python pipeline tools. Only the user interface and user/application interaction code executes on the creative clients, the business code runs on the application server or in the relational database. Communication with the creative workstations is via RMI-IIOP.

### An EJB-Based System

The previous production management system was built with about 80 first-order entity beans using container-managed persistence (CMP). The production management server code was layered using statefull session beans as facades and stateless session beans for business processes and communication services such as email, LDAP, and so on. The production management system ran on two 4-way SMP servers (primary and backup) fronting replicated database servers.

The production management system was functionally satisfactory, however there were significant problems managing and evolving the system. Code compilation times had grown to nearly 8 minutes and compile/deployment times had grown to nearly 10 minutes. The EJB deployment descriptors had grown to 50,000 lines of XML. The high compile times and cumbersome nature of EJBs encouraged developers to use very coarse grained beans. Code quality suffered because of the difficulty in refactoring. There were a number of use cases where inflating entity objects using native database SQL could yield substantial performance improvements, but this was awkward to implement in the EJB CMP architecture.

## Animation System for Major Hollywood Studio (J2EE Based)



The system required multiple JVMs to support containers, lookup services and so on. Occasionally, one process would fail to stop on shutdown and not be noticed. This would then present problems that were difficult to diagnose when the application was restarted.

### The Solution

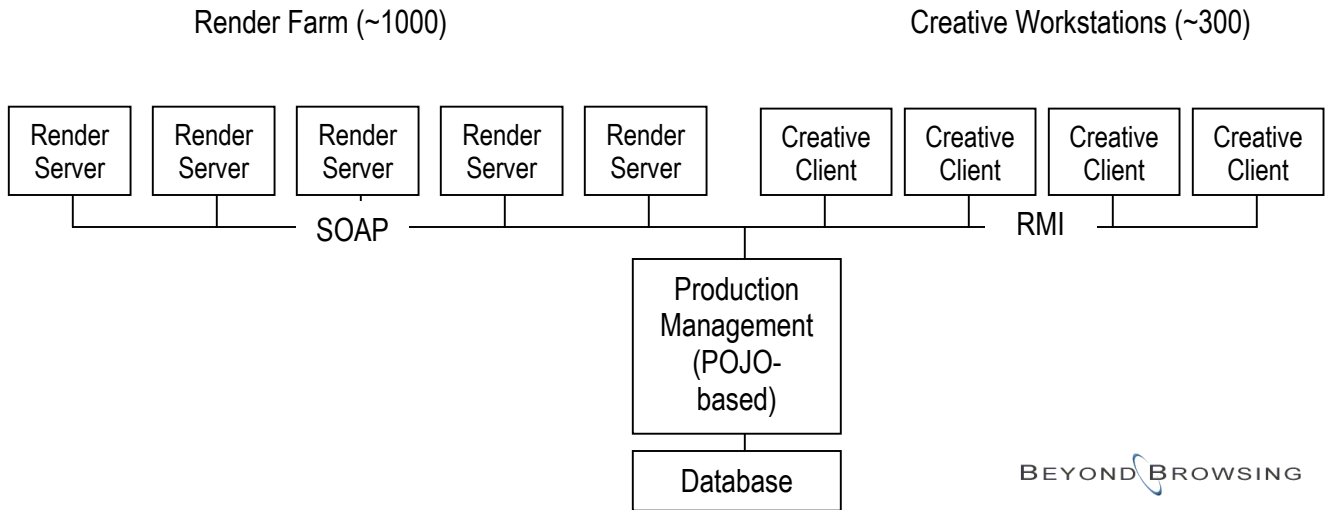
After experience with the initial production management system, and clarification of the essential architectural requirements, a simplified system was developed using plain old Java objects (POJOs) with object-relational mapping for persistence and RMI with dynamic remoting for communication to the creative clients. The architects wanted to make the production environment as clean as possible. Support for the Eclipse and JBuilder IDEs were added to enhance developer productivity.

The result of this effort is dubbed the OccamJ application server, which is now available for free download from [www.occamj.org](http://www.occamj.org)

### New Architecture

A new architecture using POJO object-relational mapping was chosen for the production management system. It provides much smaller deployment descriptors than CMP and provides multiple choices of ORM vendor. For this application, the Versant Open Access object relational mapping product was chosen. Versant Open Access is now the open source object relational mapping runtime project being led by Versant. For more information, visit <http://www.versant.com/opensource>.

## Animation System for Major Hollywood Studio (POJO Based)



BEYOND BROWSING

The new production management system uses RMI as the communications protocol to the creative clients. The creative clients are Java, so the IIOP is not needed, saving over a minute of Java2IIOP compile time. Façade objects in the production management system are served using dynamic proxies. Dynamic remoting means that any Java object (POJO) can be served without any framework dependencies.

### Migration

The migration of the production management system took 2 developers approximately 6 weeks duration. The migration involved removing EJB framework dependencies and re-mapping of the entities and relationships with the design-time tooling supplied with the open source object relational mapping project. (Similar capability is now part of the Eclipse JSR220-ORM project.) The migration was easier than it might have been because in the original production management system, EJB specific code such as context lookups, finders, and the like were factored out into helper classes and consistent code patterns were used throughout the application. Many of the changes were made through text search-and-replace techniques.

### Results

A clean architecture providing good support for testing and refactoring, and has much faster compile times. Performance of the overall production management system has improved by 15%, however overall system throughput was increased by a lot more because of the use of native threading rather than clustering across the multi-CPU servers. Entity loading using native SQL in the use cases where warranted led to much greater performance improvements in these areas.

Benefits	With EJBs	With POJOs
Rapid Testing Supported	No	Yes
Refactoring Supported	No	Yes
Number of Classes	3000	800
Compile + Deployment Time	7 minutes 40 seconds	17 seconds
Testing in the IDE Supported	Yes	Yes
Performance	Reference	15% improvement (Throughput significantly more than this)

## Conclusion

Moving from an EJB centered architecture to a POJO architecture has produced a huge difference in the agility and maintainability of the production management system. As a rough judgement, new features or revisions can be accomplished with 20% of the effort required previously. This hasn't reduced the development team by a factor of 5, but instead has enabled the development team to keep up with the requirements of the business and make critical business processes as effective as possible, instead of having the business constrained by the backlog of improvements to the application.

The OccamJ application server and open source ORM runtime have proven themselves in a business-critical, high-volume application, serving a user base of 300 and utilizing a production render farm of 1,000 machines. We believe that this architecture can be applied with similar benefits in many other business situations.

## BEYOND BROWSING

Beyond Browsing is a custom software application development company specializing in the full life-cycle development of enterprise systems and commercial web solutions using Java/J2EE technology. Beyond Browsing recently completed the development and implementation of an enterprise production management system for one of the major Hollywood studios. In addition to the company's US office in Long Beach, California, Beyond Browsing operates a highly skilled offshore development facility in Bulgaria to reduce overall development costs. For more information, contact Tim Stapleton at 562-400 2920 or visit [www.beyondbrowsing.com](http://www.beyondbrowsing.com).

### **Versant Overview**

Versant Corporation (NASDAQ: VSNT) is an industry leader in specialized data management and open data access software. Using Versant's solutions, customers cut hardware costs, speed and simplify development, significantly reduce administration costs and deliver products with a strong competitive edge. Versant's solutions are deployed in a wide array of industries including telecommunications, financial services, transportation, manufacturing, and defense. With over 1 million installations, Versant has been a highly reliable partner for over 15 years for Global 2000 companies such as British Airways, US Government, Financial Times, IBM, and MCI.

**VERSANT Corporation**  
**Worldwide Headquarters**  
6539 Dumbarton Circle  
Fremont, CA 94555 USA  
Main: +1 510 789 1500  
Fax: +1 510 789 1515

**VERSANT GmbH**  
Wiesenkamp 22b  
D-22359 Hamburg  
Germany  
Tel: +49 (0) 40 60990-0  
Fax: +49 (0) 40 60990-113

**1-800-VERSANT**  
**[www.versant.com](http://www.versant.com)**

© 2005 Versant Corporation.  
All products are trademarks or registered trademarks of their respective companies in the United States and other countries.  
The information contained in this document is a summary only.  
For more information about Versant Corporation and its products and services, please contact Versant Worldwide or European Headquarters.