

Customer Case: Serena

While it's not hard to develop a simple Ajax framework, maintaining and adding features to a homegrown Ajax framework can eventually become time-consuming and error-prone. When Serena Software got to that point, the company turned to Backbase for help.

About Serena

Serena is the leading global independent software company focused on Business Mashups and Application Lifecycle Management (ALM). More than 15,000 organizations around the world, including 96 of the Fortune 100, rely on Serena solutions to automate the application development process and effectively manage their IT portfolio.

The Challenge

"A number of our products are focused towards enterprise change management," explained Rohit (RJ) Jainendra, Serena's Senior Director of Software Development for Mashup Products. "Our source code management products are used by developers to manage their code and are used from within .NET and Eclipse IDEs as well as from the browser."

"We thought that we could build rendering engines for .NET, Java and the Web so that developers would need to build user interfaces just once and have them render in the appropriate target environment. To meet this goal, we built out a rendering framework to display these interfaces within the browser. We had a 4-5 person team that built out a presentation engine which relied on XML and JavaScript, but we found many problems with performance and cross-browser issues. After about a year, it became apparent that we would either need to increase our resource investment, or look for alternatives that could provide a truly enterprise-class platform."

The Solution

Serena considered several technologies: Adobe Flex, Java applets, ActiveX controls, and Ajax frameworks. One constraint was that many of Serena's government customers required "zero-footprint" clients; another was that the solution needed to work on all major Web browsers. These two requirements winnowed the field to Ajax, and Serena evaluated Dojo, TIBCO General Interface and Microsoft Atlas (now called Microsoft AJAX Library) as well as Backbase Enterprise Ajax.

Given its experience building a presentation engine internally, Serena had a strong preference for a commercially supported product that emphasized declarative XML over procedural JavaScript. "Backbase Enterprise Ajax was a good fit for us for four reasons," said Pavel Kramer, a Senior Software Developer working on this product. "First, it uses declarative XML; second, Backbase provides Enterprise-class support; third, the framework has good performance; and fourth, it's easy to bind Backbase XML



“ Backbase has saved us a number of resources who are now focused on building features for our product rather than building a rendering framework. ”

Rohit (RJ) Jainendra
Senior Director of Software Development for
Mashup Products

“ Backbase Enterprise Ajax was a good fit for us for four reasons: First, it uses declarative XML; second, Backbase provides Enterprise-class support; third, the framework has good performance; and fourth, it's easy to bind Backbase XML controllers to databases ”

Rohit (RJ) Jainendra
Senior Director of Software Development for
Mashup Products

controllers to databases.” While each one of these frameworks individually provided one or more items of interest, Backbase was the only one that supported both a declarative and programmatic approach to UI definition. It could be gradually introduced into existing applications and web pages and was not an “all or nothing” approach. Also, the Backbase framework was cleanly divided into a client and server framework. Since we already had a server architecture, we could just use the Backbase client for client development.” Serena took advantage of onsite training to learn the framework. It took the team about 2-3 weeks to learn Backbase.

The Result

Serena used Backbase Enterprise Ajax to build out the interface for the Mashup Manager application of Serena Business Mashups. The server side consists of a number of J2EE controllers that provide data and event handlers. The client is completely written using Backbase and communicates to the server over HTTP. Data is returned as XML and the Backbase framework renders it into the interface using XPath.

“Since the UI is completely in the client, we have a clean separation of user interface from business logic,” said Jainendra. “Although this is a very well accepted best practice, it is very difficult to achieve with JSP, ASP and other server side UI generation technologies. With Backbase we were able to achieve this 100%!”

“We developed 2-3 times faster with Backbase than we did with our internal framework,” said Kramer.

“The implementation of Mashup Manager went extremely well,” said Jainendra. “We had two developers responsible for the interface and they were able to deliver an extremely robust, modern, Ajax enabled application. Our previous attempt at using the homegrown framework had required about 8 people including folks who worked on the presentation engine. So this was a tremendous resource savings for us.

“We also derived another benefit from using Backbase. Because the client was cleanly separated from the server, our UI developers would build out early prototypes using Backbase and render mock data. This helped the entire team visualize the interface and agree on behaviors before any server side code was written, avoiding rework on the server side.

“We will be using Backbase in latter half of 2009 to replace a Win32 admin client for the Serena Business Mashups product. Backbase has saved us a number of resources who are now focused on building features for our product rather than building a rendering framework. We are able to deliver additional features for our customers thanks to Backbase.”



“ We developed 2-3 times faster with Backbase than we did with our internal framework. ”

Pavel Kramer
Senior Software Developer