

Equinox-based Open-Xchange opens Doors for ISPs



Open-Xchange has become a big fan of Eclipse Equinox and OSGi. For years their messaging and collaboration products have been assisting organizations around the world with enterprise-grade client and server tools for email, calendaring, contact, task and project management.

These products have traditionally been delivered as J2EE-based applications that make heavy use of open source projects. Client services were delivered to standard browsers using html rendered on the server, a server intensive approach that works well on dedicated equipment.

Hosted Services

Around 2005 Open-Xchange began work on a hosted version of their collaboration suite. As Juergen Geck, CTO explains, “There are many companies that don’t need and can’t justify an in-house solution. At the same time, ISPs are hungry for value added differentiators that generate substantial income. A hosted version of Open-Xchange fills this gap.” Indeed, they

“We needed to get the low latency and high performance to support thousands of concurrent users in a shared environment. The OSGi framework made this doable.”



expect that Internet providers will be able to offer full-featured collaboration tools that are competitive with commercial in-house solutions such as Microsoft Exchange. Geck expects that ISPs will be able to realize several times traditional per seat earnings with the hosted edition of Open-Xchange, giving them a competitive and lucrative client offering.

But a version of the tool suitable for use in the Software as a Service (SaaS) world had to be substantially different from their existing product architecture. “J2EE and server side rendering of html is a great environment for building in-house server applications running in the back office,” says Geck, “but we knew that a hosted version of Open-Xchange would need a much leaner, faster framework.”

A Partner Ecology

They began this journey by migrating their server side html rendering code to an AJAX based, more client-centric application. This exercise took them about a year and a half, and since then

they have been porting the final product to OSGi. “Equinox just made so much sense,” recounts Geck. “OSGi containers gave us a reusable and extensible structure that we can easily extend with other features.” Just as importantly, Geck further explains, Open-Xchange has many software partners that provide their customers with vertical extensions. For example, a partner may create add-ons to integrate Open-Xchange with Customer Relationship Management (CRM) systems. The Equinox component model fosters this partner ecosystem and the OSGi framework gives them a rock-solid framework and API they can rely on.

Clean Coding

Although work on the Equinox-based solution is ongoing, Open-Xchange has already integrated their administrator tools using Equinox. In this and ongoing work their developers have found productivity gains from the feature rich IDE and debugging tools. The well-

defined OSGi APIs also encourage cleaner coding and dependency management. Geck expects that these characteristics will also make it easier to support their partners.

But perhaps the most important feature of Equinox was the ability to dispense with platform functionality they did not need, and to move much of the remaining functionality to the client.

“Coming from the J2EE world, we really needed to strip the hosted Open-Xchange server down to get the low latency and high performance needed to support thousands of concurrent users in a shared environment,” says Geck. “The OSGi framework and our modular architecture really made this doable.”

The flexibility of Equinox has opened doors for Open-Xchange, allowing them to develop flexible solutions that will allow their customers to compete with established solutions for years to come.

[Ron Stone](#) is a technology writer and content management consultant based in Ottawa, Canada.

March, 2008