Spring MVC vs. WebWork Smackdown

Presented by Matt Raible and Matthew Porter

Introductions

- Java/J2EE Experience?
- Open Source Experience?
- Web Framework of choice?
- Preferred Servlet Engine?
- Operating System of choice?
- Preferred IDE?

Matt vs. Matthew

The Match

WebWork Rocks

- Designed for Web
- Action is Model *and* Controller
- Tag Libraries write boiler-plate code for you
- Interceptors
- Easy to Test
- Supports JSP, Velocity, FreeMarker, JasperReports, and XSLT
- SiteMesh

Spring MVC is Sweet

- Different Controllers for different operations
- Dependency Injection
- Easy to configure
- Supports JSP, Velocity, FreeMarker, JasperReports, XSLT **and** PDF, Excel
- SiteMesh, but also supports Tiles
- Spring Web Flow

Why Spring MVC Sucks

- XML is too verbose
- Controllers are tied to the Servlet API, can't test out of container
- Model and CommandClass - too much like Struts
- Too many Controller choices
- JSP <spring:bind> tag too verbose
- No client-side validation
- No support for Ajax Technologies

VIRTUAS

Why WebWork Stinks

- Documentation
- No client-side validation
- JSP Tags do too much
- No Professional Support
- No page composition support (like Tiles)
- Spring integration too verbose
- Requires i18n bundle per action
- Security issues with ModelDriven

Why they both suck

- Request-based MVC Frameworks are dying
- Component-based frameworks are the future
- Why all the XML?
- Where are the tools?

Ajax Support

- DWR Project for Spring
- Client-side validation in WebWork 2.2
- Fancy JavaScript: Dojo, Prototype and script.aculo.us
- Create your own with Velocity/FreeMarker Macros and JSP 2.0 Tags

Tips and Tricks

Don't eat yellow snow.

How do I choose?

Questions?

mraible@virtuas.com
matthew.porter@contegix.com

VIRTUAS