



Customization

Lukas Renggli

renggli@iam.unibe.ch, University of Bern
renggli@netstyle.ch, netstyle.ch GmbH

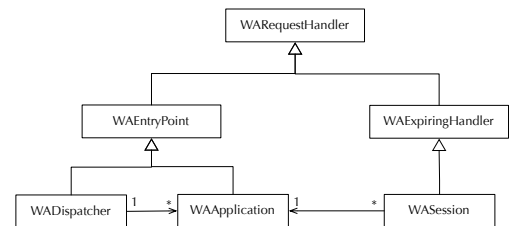
Customization

- ✧ Session Class
- ✧ Error Handler
- ✧ Renderer
- ✧ Apache Serving

Session Class

- ✧ Create a subclass of `WASession`.
- ✧ Register your new subclass in the configuration interface as the *session class* to be used with your application.
- ✧ Access the current session instance from any component by sending `self session` (or by evaluating `WACurrentSession value`).

Application Dispatching



Session Examples

- ✧ Useful to keep an instance of the *current user* and *its permissions*.
- ✧ Useful to keep an open *connection to the database* you are using for persistence .
- ✧ Clean up the session state (e.g. close the database connection) by overriding `#unregistered`.

Error Handler

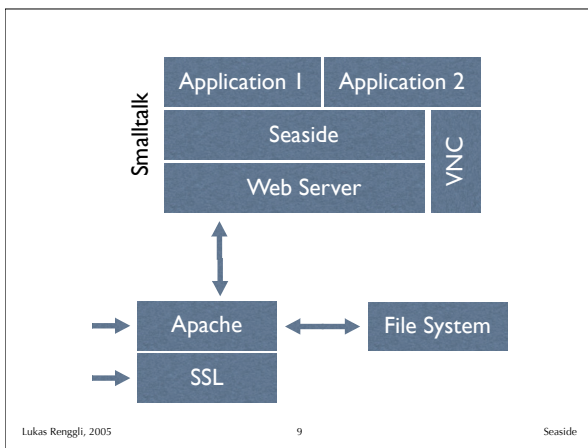
- ✧ Customize the error handling by subclassing `WAEErrorHandler`:
 - ✧ Display a simple error dialog in the design of your application to the user.
 - ✧ Send an e-mail with the stack-trace and the debug link to the developer.
 - ✧ Allow the user to continue the session from the last page.

Renderer

- ✧ Use *class-extension* to add new tags or repetitive layout-constructs to `WAHtmlRenderer`.
- ✧ Override `#rendererClass` in your components to use a *custom renderer* of yours, e.g. a subclass of `WAHtmlRenderer`.

Apache

- ✧ Virtual-hosting
- ✧ Security and SSL
- ✧ Serve huge static files faster
- ✧ Hide long and ugly access-paths
- ✧ Map Seaside applications anywhere into an existing directory tree



Virtual Hosting

```
<VirtualHost myapp.server.com>
    ServerAdmin renggli@iam.unibe.ch
    DocumentRoot /var/www/myapp

    RewriteEngine On
    RewriteCond %{REQUEST_URI} !^/resources/(.*)$
    RewriteRule ^/(.*)$ http://localhost:8080/seaside/myapp/$1 [L,P]
</VirtualHost>
```

Summary

Seaside allows customization and tweaking within all parts of the framework.