

Introducing Portable.nsd

A portability framework for OpenACS

John Sequeira

<http://www.pobox.com/~johnseq>

9/2002

OpenACS

OpenACS is a framework for developing community sites using TCL and AOLServer

- It has a sophisticated data model, module packaging framework, an emphasis on high performance, low learning curve, and coding ease
- Supports model-view-controller, pure script or mixed html/script
- It's the best!!!
- But ...

OpenACS & Portability

- OpenACS runs only under AOLServer!
- AOL has de-emphasized cross-platform support
- AOLServer has limited database support
- AOLServer has small-ish user community and it's performance, reliability and ease of use are not widely known.
- Prior attempts to port OpenACS to IIS and Apache have not been maintained
- AOLServer is driven by and for AOL, on their schedule
- “AOLServer, what's that?”

Solution: “portable.nsd”

- Build a deployment container for OpenACS that recreates the AOLServer ns_ API's, so that OpenACS code can run under any web server
- Implement this in pure TCL to minimize development time and maximize portability and database support

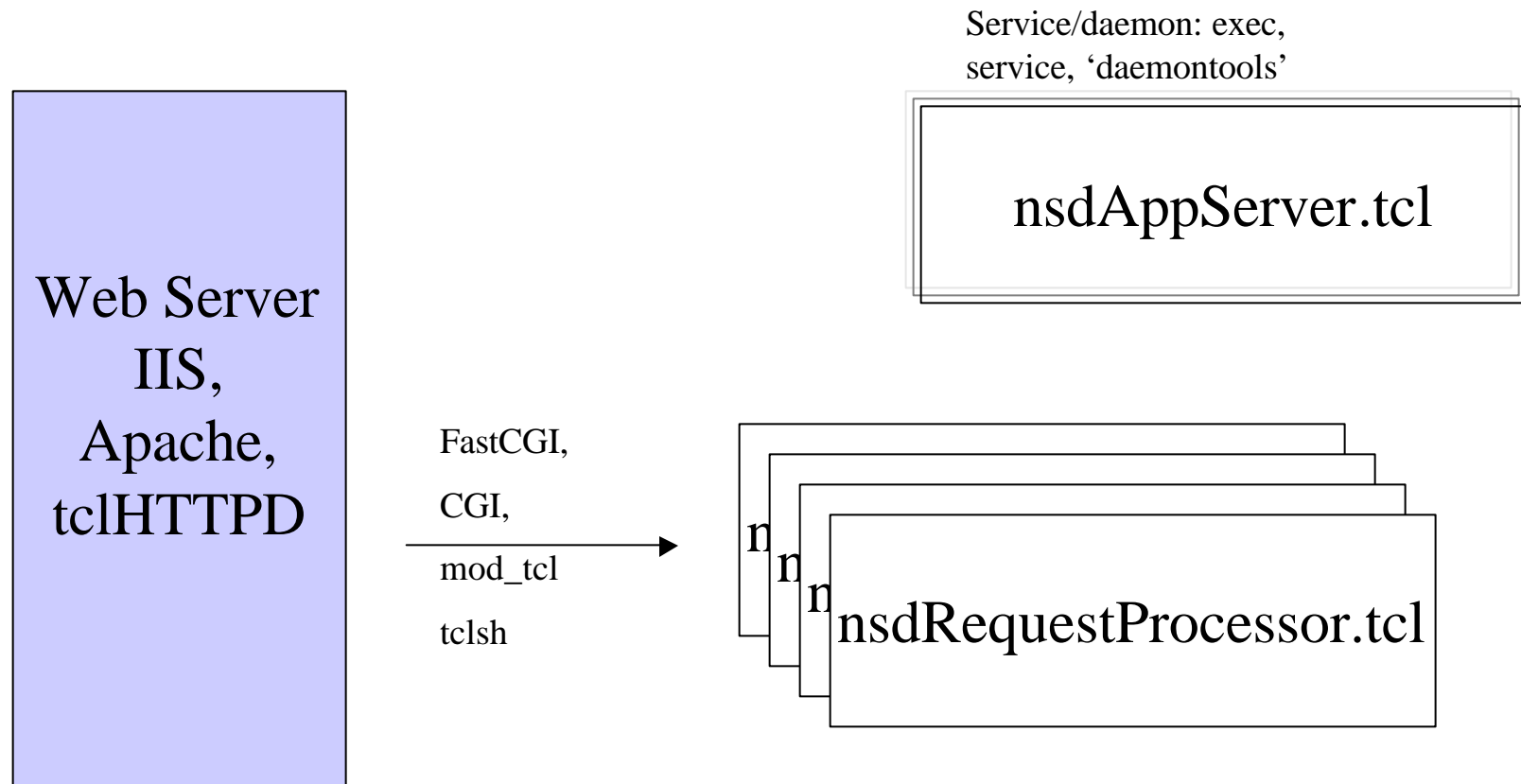
What's not portable about OpenACS?

- ns API in general
- database pooling
- nsv's
- ns_register_proc, ns_register_filter
- startup/shutdown

What is portable.NSD?

- An TCL-only emulation layer that hides the details of the deployment environment.
- Fully backwards compatible with OpenACS code base.
- Based on nstcl, a partial implementation of AOLServer ns api

Portable.NSD Architecture



nsdAppServer.tcl

- Handles Application Server responsibilities for OpenACS
 - Scheduling
 - Global State (nsv arrays) using Tequila
 - Database initialization, installation
 - All “singleton” behavior

nsdRequestProcessor.tcl

- Handles requests for pages, returns HTML
- Bytecode compiles and caches(?) page tcl
- Communicates with AppServer
- Has persistent database connection(s)

Benefits

- Pure tcl means no or little problem integrating broader TCL community efforts
 - ODBC, various database drivers
 - XOTcl
 - JDBC via Jacl, JavaBlend, Bean Scripting Framework
 - TCLScript, ADO, ASP, .NET
 - Parrot?

More Benefits

- Run and debug pages from command line
- Easy IDE debugging
- Possibility of standalone deployment (creating one or two exe's for demo's)
- Possible low-cost ISP deployment
- Vastly fewer reasons for interested users to not try and select it
- Running under tcsh should easily support white-box unit testing

Alright, the last benefit

- If the major web platforms were supported, the community would grow a great deal

Implementation

- Identify unimplemented ns functions
- Stub missing functions with dummy environment
- Write initialization script to load all necessary libraries
- Test against pre-initialized OpenACS database
- Implement request processor dispatch tables for filters and registered proc's

Things to Prototype

- “Tequila” as state server
- Bootstrapper to load all necessary libraries
- Make request processor container independent
- mod_rewrite/FastCGI combination
- Replace ns_xml

Candidate Reference Platforms

- Apache 2.0 w/mod_fastcgi
- Apache 2.0 w/mod_tcl
- Apache 2.0 w/Rivet
- tclHTTPD Standalone

Fun Things to Try

- Standalone tclHTTPD deployment and exe packaging
- Tclblend
- mod_webdav integration
- Running in-process w/IIS using TCLScript