An Introduction to Repoze.Zope2

Plumbing Zope 2 Into The WSGI Pipeline

For the PyATL December 2007 Meeting

Chris McDonough, Tres Seaver, Agendaless Consulting

Case Study: repoze.org configuration

- repoze.kiss integrates Zope2 publisher with filesystem-based content
- repoze.mmwsgi drives Mailman inside a WSGI app
- Bug tracker: roundup (WSGI-aware)
- Blog: pyblosxom (WSGI-aware)
- repoze.plone
- Grok applications: AnimalTree, Bookshelf, TodoList
- All themed via a single static HTML page using Deliverance
- Served from Apache + mod_wsgi

Eggs and WSGI

- WSGI standard begun in 2004, eggs about the same time.
- Repoze.Zope2 is about:
  - allowing non-Zope developers to take advantage of Zope technologies via WSGI & eggs.
  - allowing Zope developers to take advantage of WSGI components.

Repoze

Other Top-Level Packages

- repoze.grok -- Allows you to run the Grok application server under the Repoze stack.
- repoze.plone -- Allows you to run Plone 3 under the Repoze stack (via repoze.zope2)

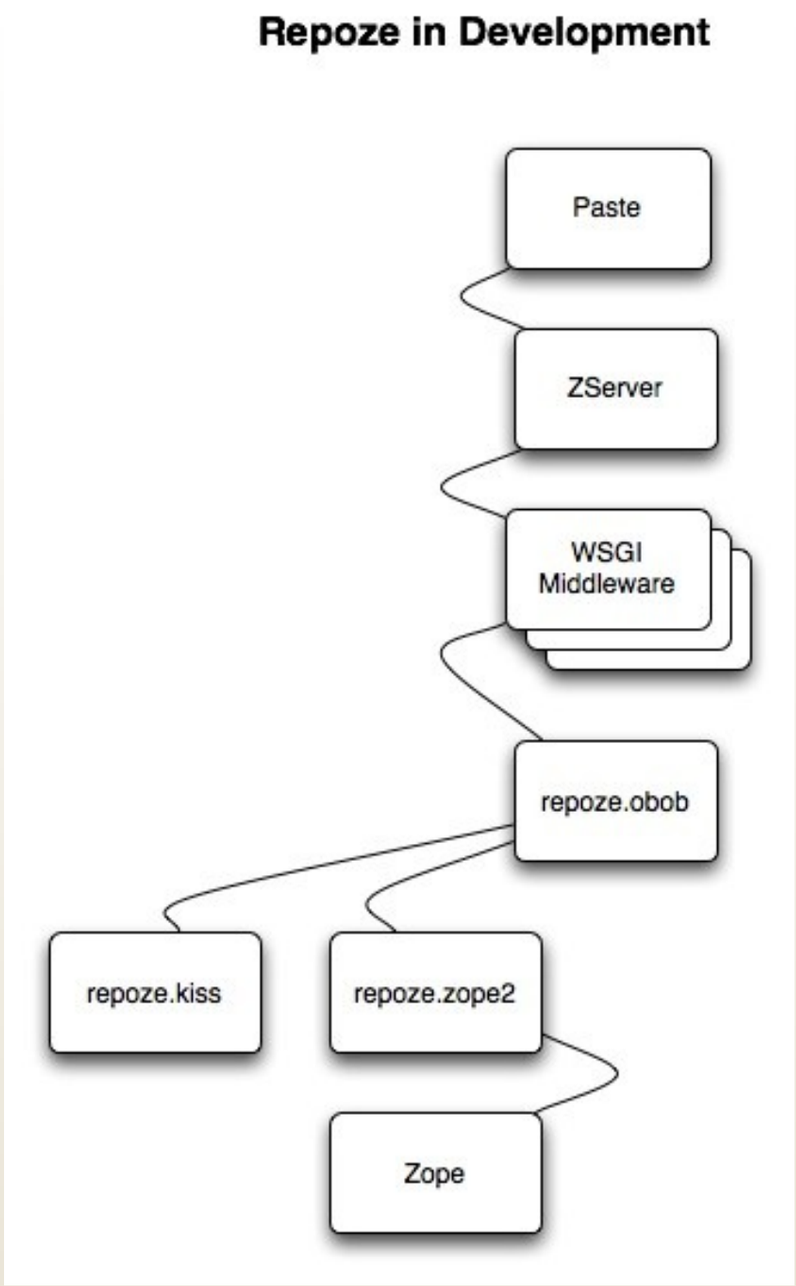
repoze.zope2

- Reimplementation of Zope 2's ZPublisher to fit into a WSGI stack natively.
- Aims to be 100% backwards compatible with all Zope 2 products.
- Installs all software (including Zope) via eggs.

Dependencies

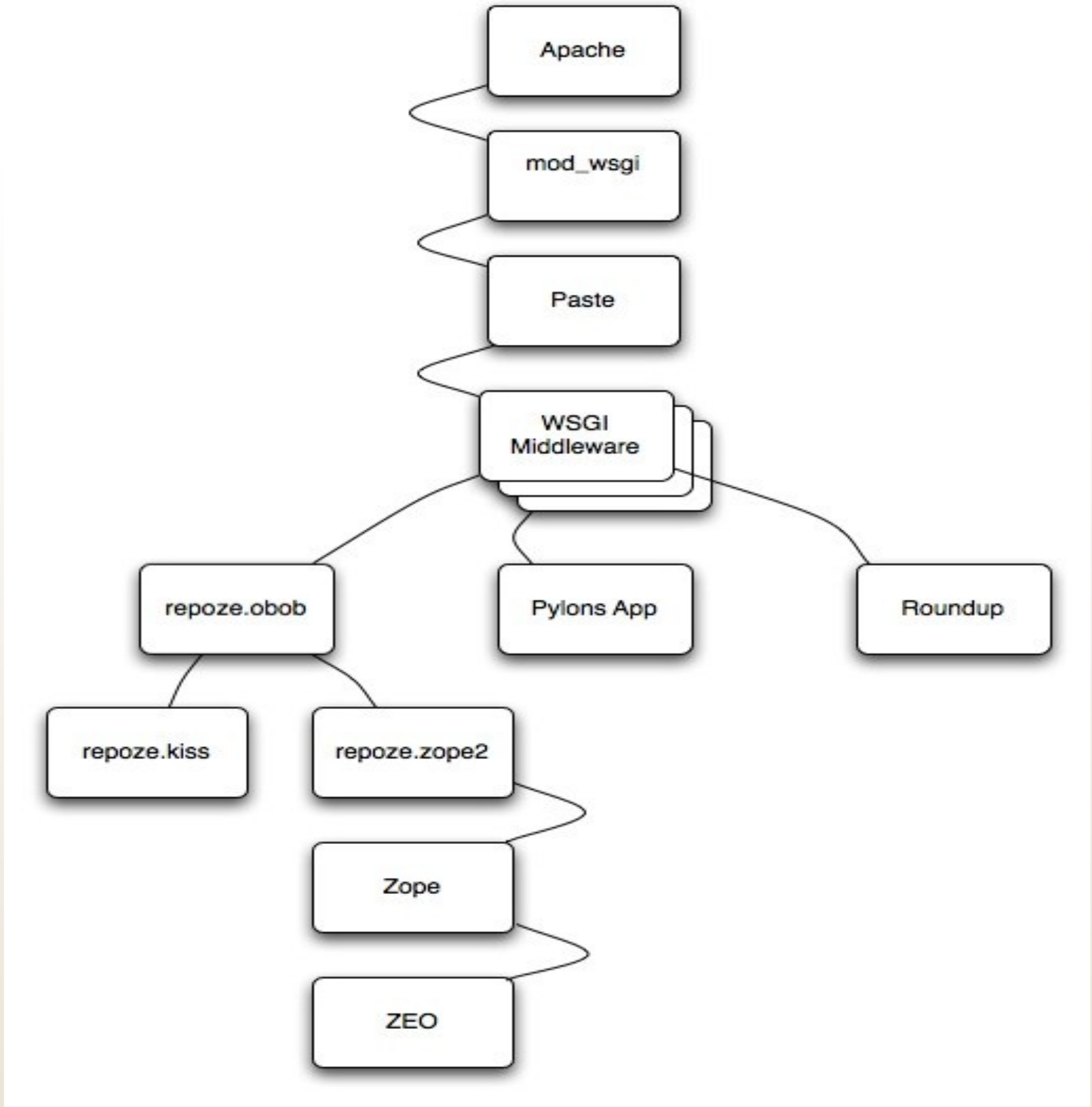
- repoze.zope2 depends heavily on Ian Bicking's **Paste**, particularly **PasteDeploy**, which provides a declarative syntax for configuring WSGI "pipelines".
- Setuptools.
- Obviously, repoze.zope2 depends on Zope 2.

Repoze In Development



Repoze

Repoze in a Deployment



Repoze

WSGI: Servers, Middleware, and Applications

- Servers accept requests from browsers/clients and pass request data along to applications.
- Servers respond to requests using data returned by the application.
- Applications return responses.
- Middleware is an application that calls "the next" application; functional composition forms a "pipeline".

Repoze

Middleware Is Cool

- Examples of middleware:
  - Deliverance, HTML/XML output transformation. Repoze.org.
  - evalerror, catch exceptions and query information about them via a browser.
  - repoze.tm, move transaction management out of the publisher and into middleware.

Composition via Middleware

- Previously hardcoded bits are now configurable (and disableable), e.g. ConflictError retry policy.
- Different pipelines expected for development vs. deployment (e.g. "evalerror" is useless and dangerous for deployment).
- Non-Zope people can use the middleware.

Repoze

What's Different

- No effort to make work on Windows (but not actively preventing it).
- standard_error_message doesn't do anything (a generic 500/404 error handler in "httpexceptions" replaces it).
- The Zope 2 "error_log" object doesn't log exceptions; a replacement at '/__error_log__' is provided (it's middleware).
- "zopectl" is gone, replaced with single-purpose scripts.

Repoze

Installation

- See http://repoze.org/quickstart.html
- Essentially two commands from an internet-connected UNIX system.
- Eggs are downloaded, a "virtualenv" is created that serves as the "sandbox", aka "instance home" for the Zope 2 installation.
- Virtualenv is cool and important.

Starting Repoze.Zope2

- bin/paster serve etc/zope2.ini
- Show it.
- [pipeline:main] is the most interesting part.

A Tour of the Paste Config File (Middleware)

- cgitb -- top-level exception catcher
- httpexceptions -- turns special exceptions into HTTP responses
- retry -- retries ConflictErrors
- tm -- performs transaction management
- vhm_xheaders -- changes path-related info in environment to do virtual hosting.
- errorlog -- replacement for Zope2's "error_log"

A Tour of the Paste Config File (Server)

- egg:repoze.zope2#zserver is Zope 3's WSGI server.
- Can replace with a reference to any WSGI-compliant server implementation.
- This stanza not used when run under Apache via mod_wsgi (need a .wsgi file).

Scripts In A Repoze.Zope2 Sandbox

- addzope2user -- equivalent to "zopectl adduser".
- debugzope2 -- equivalent to "zopectl debug"
- runzope2script -- equivalent to "zopectl run"
- Other scripts for other packages are also in "bin" (mkzeoinst, virtualenv, etc).

Sandbox Directories

- Products
- bin
- etc
- import
- include
- lib
- var

Repoze

Configuration, Log, and Data Files

- Config files are in sandbox etc dir (zope.conf, zope2.ini, site.zcml).
- Log files are not written to disk (they're output to the console by default); change this either in zope.conf or in errorlog configuration.
- Can't put Python packages into "lib/python" in sandbox like old Zope instance home; need to go into site-packages of the sandbox (easy_install).
- Data files (Data.fs) live in var directory of sandbox.

Evalerror Demonstration

- A demonstration of replacing cgitb with evalerror.

Repoze Resources

- Home page: http://repoze.org/
- Subversion: http://repoze.org/viewcvs
- Bug tracker: http://bugs.repoze.org/
- Mailing lists: http://lists.repoze.org/
- IRC: irc://irc.freenode.net/#repoze

End

- Full-stop.
-
- Chris McDonough, chrism@agendaless.com
- Tres Seaver, tseaver@agendaless.com

Repoze