PHP Intranet Services

DC PHP Conference 2006

Mike Naberezny

About Me

- Coauthored Zend PHP 5 Certification
- Professional software engineer
- Dynamic language advocate
- Open source contributor

Agenda

- Talk about “intranet” and buzzwords like SOA
- Discuss why PHP fits on the intranet and how this is different from the internet
- Look at some different technologies as they relate to the intranet

Agenda

- Intranet
- PHP
- Web Services: REST, SOAP, XML-RPC
- Java Integration
- Final Thoughts
- Q&A

Intranet

Intranet

- Not accessible via the Internet
- Behind a corporate firewall
- Probably part of some kind of "enterprise"
- May have many differences in technology and culture from the internet at large

Enterprise(-y)

- Has several departments and an "intranet"
- Needs new apps, but also integration services, quickly
- Core business probably isn't internet related
- Full of old, weird, fragile, legacy stuff* that doesn't quite fit together but needs to
- Participates in the "SOA movement" or probably will soon

* Possibly the defining characteristic

SOA Movement

- No two CTOs agree on what SOA means
- Disparate technologies, performing discrete business processes, loosely coupled by contract through service interfaces
- Almost always begins with “service enablement”

Service Enablement

- Most believe SOA begins with creating new internal web services or decoupling existing business processes through web services
- Occurs on all levels of an organization but the actual programming to do this is usually departmental

PHP

PHP

On the Intranet

- A huge number of intranet apps, especially at the department level, are quick little apps
- PHP might be positioned to be the new VB and MySQL is already the new MS Access
- Deployment is somewhat easier than with many competitive languages and it’s possible to build servers requiring little maintenance

PHP

On the Intranet

- PHP can do more than quick little apps. It can provide services, wrap existing processes with services, and be a gateway or filter between services
- PHP's large number of existing extensions and libraries make it a suitable glue language for intranet services

PHP 5

- Most new development is being done in PHP 5 for good reason
- PHP 5 extensions for SOAP and especially XML handling make it ideal for both consuming and providing services
- Object overloading support makes writing clients for RPC services very attractive
- The best new PHP libraries for working with services are being written for PHP 5

Web Services

REST

(The Good)

- A new paradigm in thinking about web services
- Uses HTTP verbs, URLs as targets (think resources), XML payloads as data for CRUD
- Very popular on the internet
- Lightweight payloads faster on the wire
- Less processing overhead involved due to less object marshaling and payload serialization

REST

(The Not So Good)

- Conceptually more difficult for non-web developers, of which intranets are full
- Difficult to describe and document with traditional API documentation
- Bound to the HTTP protocol specifically and request/response more generally

REST

Libraries

- No special libraries beyond an HTTP client and XML handling are required for consuming REST-based services
- SimpleXML and pecl/http
- Some frameworks are emerging for REST servers but aren’t necessary in my cases
- Ruby on Rails’ ActiveResource is one of the most interesting REST frameworks emerging

SOAP

(The Good)

- Serializes method calls, parameter types, and faults that map close to traditional class-based APIs
- Endorsed by the W3C
- Many vendors have standardized on SOAP

SOAP

(The Bad)

- Some of the heaviest payloads you will encounter in web services
- Very difficult to troubleshoot on the wire
- Specification is very large and difficult to understand
- Many partial implementations exist and this hinders interoperability

SOAP

Libraries

- SOAP extension is built into PHP 5
- SOAP4Py (Python)
- All popular Java platforms support SOAP
- Partial implementations available for most languages

XML-RPC

(The Good)

- Specification is considerably easier to understand than SOAP
- Payloads are significantly smaller than SOAP
- Like SOAP, it also has wide adoption. XML-RPC powers most blogging APIs

XML-RPC

(The Bad)

- Not an “official” standard anywhere
- Specification is nebulous about some issues like character encoding
- De facto standards for introspection and method boxcars are not universally supported
- No longer fashionable on the internet

XML-RPC

Libraries

- PEAR's XML_RPC2
- Zend Framework's Zend_XmlRpc
- xmlrpclib (Python)
- Apache XML-RPC (Java)
  If the remote server is Apache XML-RPC, be careful it doesn't use the extended data types.
- Complete implementations are available for just about every language

Java Integration

Java Integration

- Most intranets use Java in some way and many are very heavily invested in it
- It’s not always practical to expose parts of a Java system through web services
- Sometimes tighter integration with the Java platform is necessary

Java Integration

- Extending PHP to communicate with a JVM
  C PHP interpreter running under Apache or similar web server as usual, bridged to a JVM through an extension.

- Integration with Java Application Server
  Rewrite the PHP interpreter in Java and run PHP directly inside the application server.

Zend Platform

- Features include an opcode cache, server management and intelligence capabilities, and a Java bridge
- Hosts its own JVM instance -- a single instance for all instances of PHP
- Allows Java objects inside that JVM instance to be directly instantiated from PHP.
- Doesn’t directly integrate with an application server but allows sharing of Java libraries useful for accessing persistence, messaging, etc.

Caucho Quercus

- Complete re-implementation of the PHP interpreter in Java, running inside their Resin application server
- Complete enough to run many popular PHP applications including MediaWiki, Drupal, Mantis, and phpMyAdmin
- Performance is surprisingly good but varies. In most cases, it is comparable to C PHP, and in some cases it is even faster

Caucho Quercus

- Running inside the application server means separate PHP servers aren’t needed
- Allows for PHP scripts to be used as servlets where JSP might have been used
- Most direct access to the Java environment from inside PHP
- PHP extensions can be written in Java

Final Thoughts

Final Thoughts

Service Enablement

- Don’t be tempted to do things like dig through another department’s databases, even if the PHP connectivity exists
- Loosely couple your business processes through web services as you go
- Future interoperability will be improved and processes can be cleanly replaced later

Final Thoughts

REST vs. SOAP and XML-RPC

- REST is easier to troubleshoot (use cURL)
- For simple data, REST can be significantly lighter
- Exposing class-based APIs means you can auto-generate documentation with tools like phpDocumentor.
- Unit testing class-based APIs is more convenient

Final Thoughts

Integration Beyond HTTP

- PHP extensions can glue together data from many different sources not typically used on the internet, such as Windows COM.
- SOAP and XML-RPC method calls can easily be put in message queues, the file system, or even email. REST is bound to HTTP.
- PHP is generally not suitable for writing long running processes or daemons. Python and Ruby are more suited and provide similar capabilities for most types of integration.

Final Thoughts

- If you have the luxury of designing your services, consider your audience carefully when choosing what technologies to use
- Avoid premature optimization. Focus on writing maintainable code and optimize when you find problems
- Version your APIs
- Practice unit testing and continuous integration

Questions?

Thank You

http://mikenaberezny.com

mike@naberezny.com