

OODT Grid Services over HTTP

Sean Kelly

2005.7.11

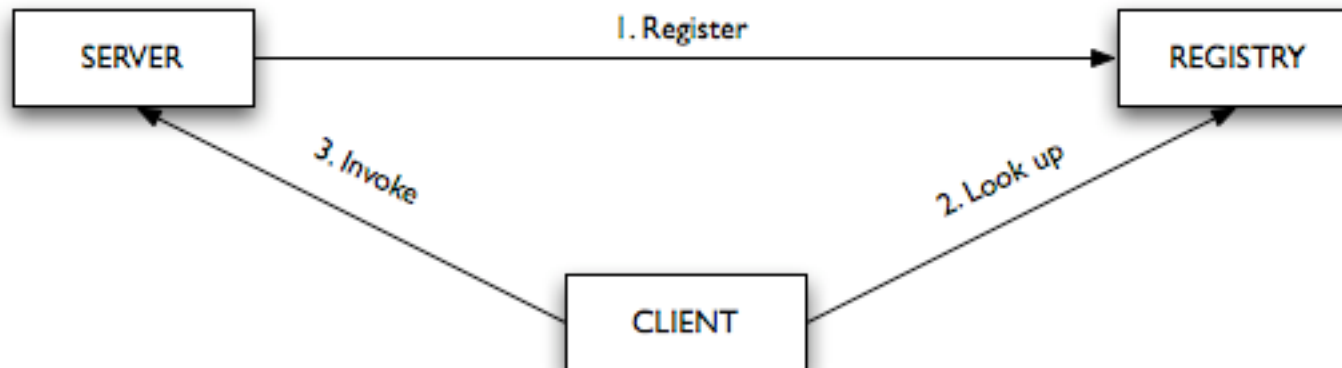
Overview

- Old RPC-based approach
- Web grid
 - Deploying
 - Configuring
 - Using
- Performance

OODT Services

- Historical implementation is *procedural*
- Remote procedure calls
 - CORBA
 - RMI
- Centralized *naming registry*

Procedural Architecture



RPC Benefits

- Familiar programming paradigm:
`byte[] data = service.get(arg);`
 - “get” could be a remote method
 - Looks like a local invocation
- Move servers without affecting clients
 - Server registers under a name
 - Clients check registry for server’s current location

RPC Drawbacks

- Limit to argument/return value sizes
`byte[] data = server.get(arg);`
 - What if returned data is a HiRISE image?
 - What if the argument is large too?
- Arguments/return values must fit in memory
 - Programs can *stream* data over I/O ports
 - But RPC cannot stream

Workaround?

- **Modify the interface:**
`byte[] get(X arg, long offset, int size);`
 - Repeatedly call “get”
 - Use various offsets and size
 - Until you get the whole product
- **Slow**
 - Slower than streaming

The Workaround

- Currently implemented for product service
- Profile service can still fail when large numbers of profiles match
 - And/or large profiles match

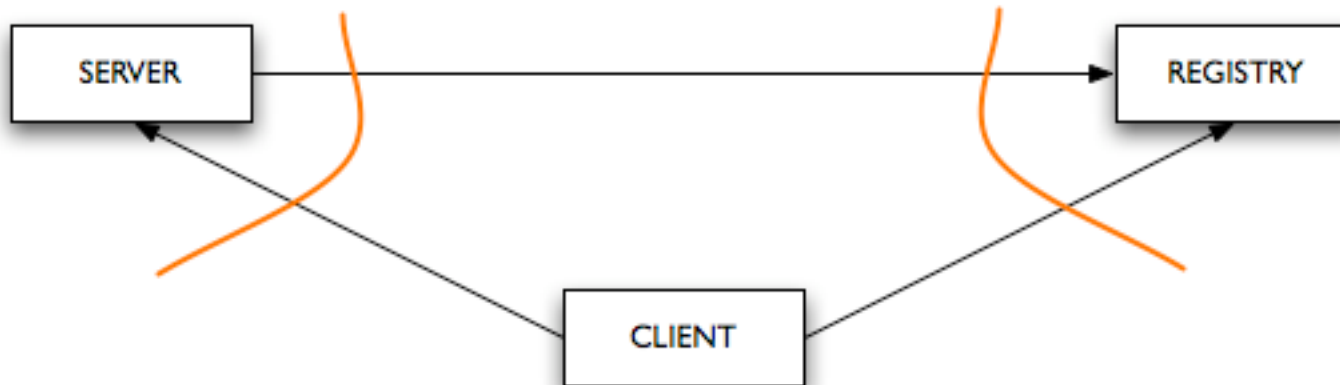
RPC Drawbacks

- Naming registry is a point of failure
 - Even if servers are up and running, clients cannot locate them
- Workaround
 - Set up multiple registries
- Worth it?
 - ERNE servers have moved 3 times in 3 years
 - PDS servers have moved *once* in 3 years

Administrative Overhead

- Naming registry requires security policy changes at *multiple institutions*
 - Servers must make outgoing calls to registry, as well as accept incoming requests

Firewalls



Server Manager

- Myche McAuley, Imaging Node, PDS:

“Look, I trust you, but the Server Manager makes me darn nervous. Can’t you just give me a WAR file?”

Motivation

- Saddening performance
- Constant administrative vigilance
- REST

Web Grid

- Provide profile and product service over HTTP
 - No central naming registry
 - You need to know your service endpoints
 - But they *rarely* change
 - Streaming profiles and products
 - As fast as HTTP can be
 - Yet compatible
 - On both server and client side

Why Web Grid?

- Web (HTTP) accessible
- Grid service
 - We call OODT “grid” after all!

Web Grid

- No Server Manager
 - Remote administration is an *option*
 - Limited to updating profile/product handlers
- Packageable
 - Deploy the raw WAR file
 - Preconfigure the WAR file with selected profile/product handlers
 - Include the WAR file with a lightweight servlet container

Getting Web Grid

- Download it from
`http://oodt.jpl.nasa.gov/web-grid/`
- Build it yourself:
`cvs checkout web-grid`
`cd web-grid`
`maven war`
- Deploy `web-grid-1.1.0.war` to your favorite servlet container
 - Then visit the deployed webapp

Tested Servlet Containers

- Tomcat 5.5.9
 - Tomcat 5.5 is minimum requirement
 - Will not work with 5.0 or 4
- Jetty 5.1.4
 - Not tested with earlier Jetty containers

Making Generic Web Grid

- Compile, make the war, and rename it:
`cd web-grid`
`maven war`
`mv target/web-grid-1.1.0.war target/test.war`
- Why?
 - So that URLs are just “/test” instead of “/web-grid-1.1.0”
 - Yes, you can override this in your servlet container
 - But I’m in a hurry to finish these slides

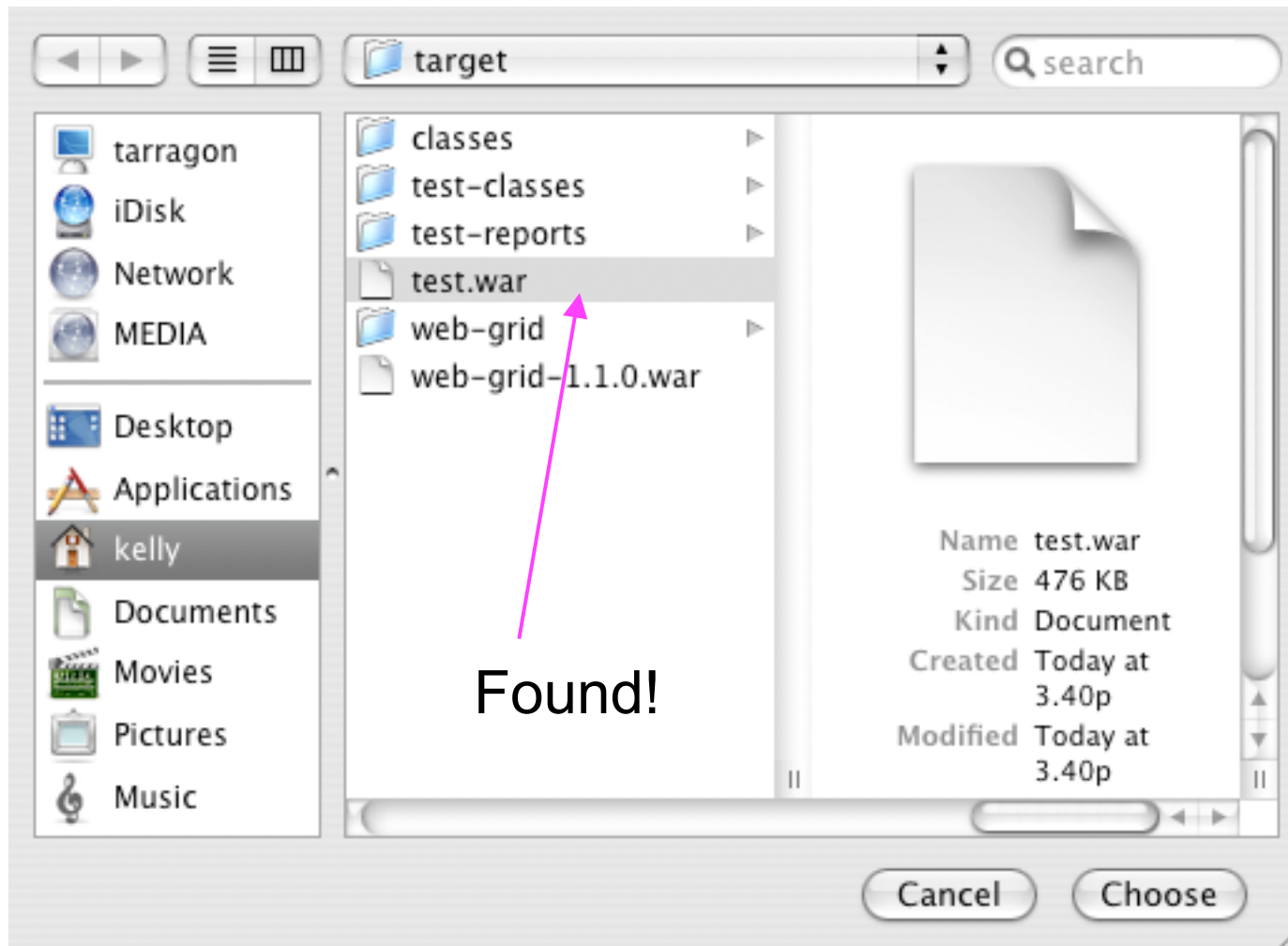
Deploy the Webapp

The screenshot shows a web browser window with the address bar containing `http://tarragon:8080/manager/html`. The page title is `/manager`. The main content area is titled **Deploy** and is divided into two sections:

- Deploy directory or WAR file located on server**: This section contains three input fields: "Context Path (optional)", "XML Configuration file URL", and "WAR or Directory URL". Below these fields is a "Deploy" button.
- WAR file to deploy**: This section contains the text "Select WAR file to upload" followed by a "Choose File" button and the text "no file selected". Below this is another "Deploy" button.

A pink arrow points from the text "Click!" to the "WAR file to deploy" section.

Find the WAR file



Click Deploy

WAR or Directory URL:

Deploy

WAR file to deploy

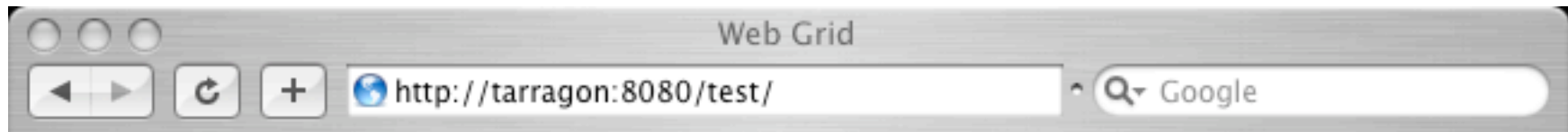
Select WAR file to upload test.war

← Click!

Server Information					
Tomcat Version	JVM Version	JVM Vendor	OS Name	OS Version	OS Architecture
Apache Tomcat/5.5.9	1.4.2_07-215	"Apple Computer, Inc."	Mac OS X	10.4.1	ppc

Copyright © 1999-2005, Apache Software Foundation

Visit the Webapp



Web Grid

Click!

- [Configure](#) the system
- [Learn more](#) about OODT web-based grid technologies

Copyright 2005 California Institute of Technology. ALL RIGHTS RESERVED. U.S. Government Sponsorship acknowledged.

Log In



Web Grid

Log In

Administrator password:

Passwords are case sensitve; check your CAPS LOCK key, if necessary.

Log In

Logging In

- Default password is “hanalei”
 - You can customize that for pre-packaged installations
 - As well as every other setting
- All settings are saved in WEB-INF/config.xml
 - That means that the webapp must be deployed *unpacked!*

Changing the Password

Web Grid Configuration

http://tarragon:8080/test/login

Web Grid Configuration

Administrative Settings

New Administrator Password
Leave blank to leave password unchanged.

.....

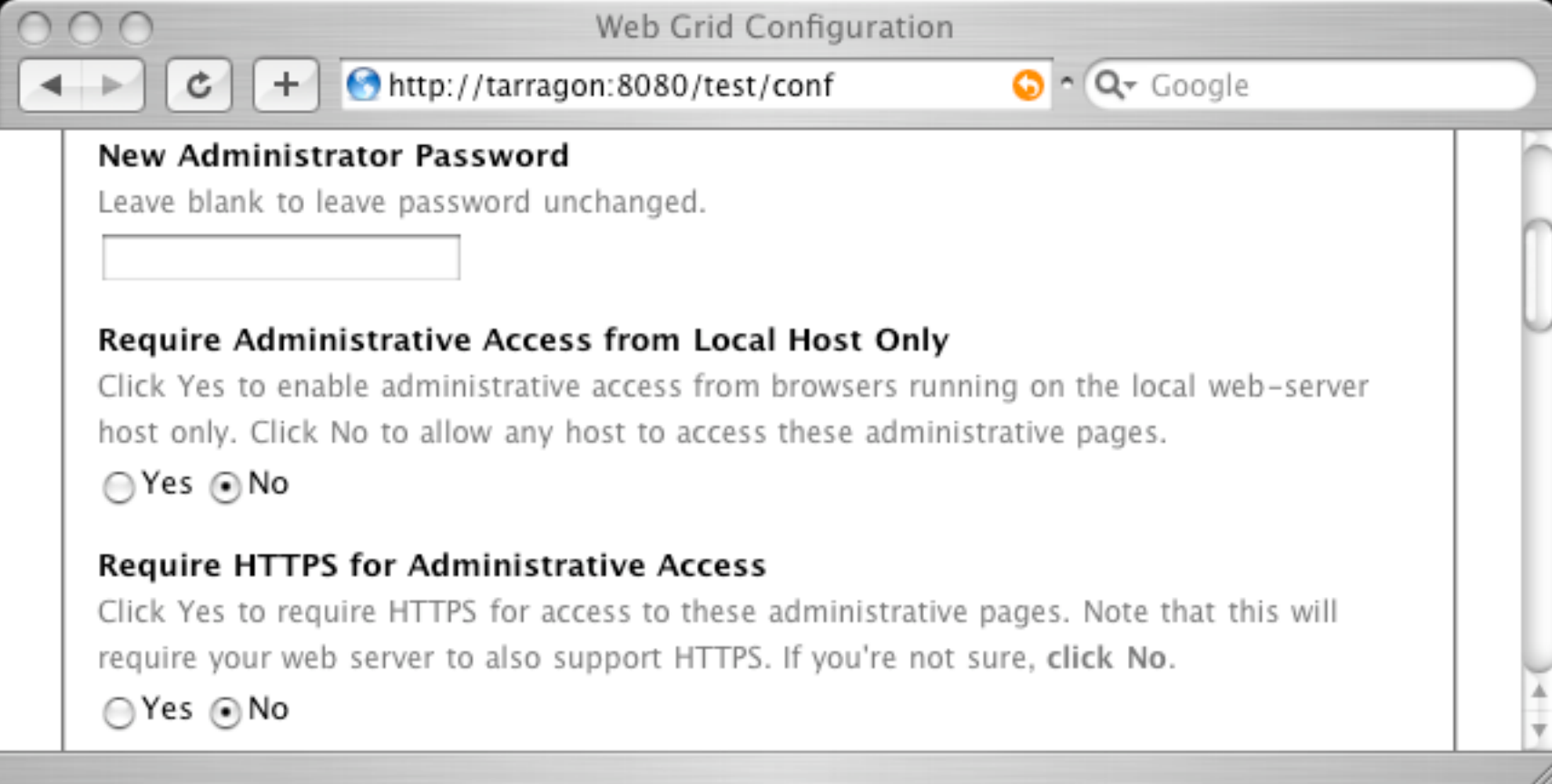
Require Administrative Access from Local Host Only
Click Yes to enable administrative access from browsers running on the local web-server host only. Click No to allow any host to access these administrative pages.

Yes No

Administrative Access Control

- Protected by password
- Can be allowed from local host only
 - Browser must run on same host as servlet container
- Can be allowed via HTTPS only
 - Servlet container must support HTTPS

Administrative Access Control



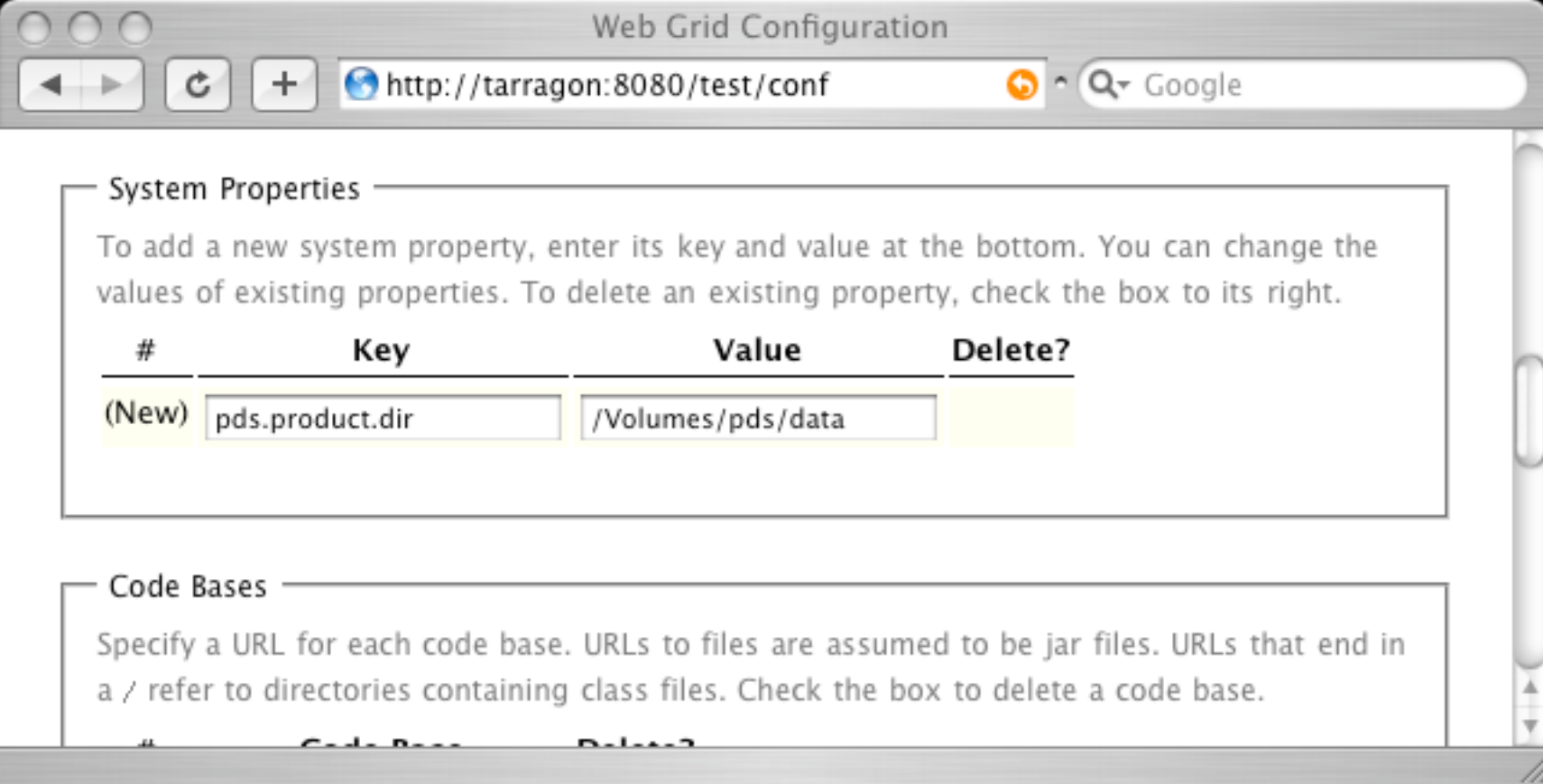
The screenshot shows a web browser window with the title "Web Grid Configuration". The address bar contains the URL "http://tarragon:8080/test/conf". The page content includes three sections:

- New Administrator Password**
Leave blank to leave password unchanged.
- Require Administrative Access from Local Host Only**
Click Yes to enable administrative access from browsers running on the local web-server host only. Click No to allow any host to access these administrative pages.
 Yes No
- Require HTTPS for Administrative Access**
Click Yes to require HTTPS for access to these administrative pages. Note that this will require your web server to also support HTTPS. If you're not sure, click No.
 Yes No

Setting System Properties

- Many query handlers consult system properties
 - For database URLs, product directories, etc.
- Set System Properties via web interface
 - Or pre-configure the WEB-INF/config.xml file

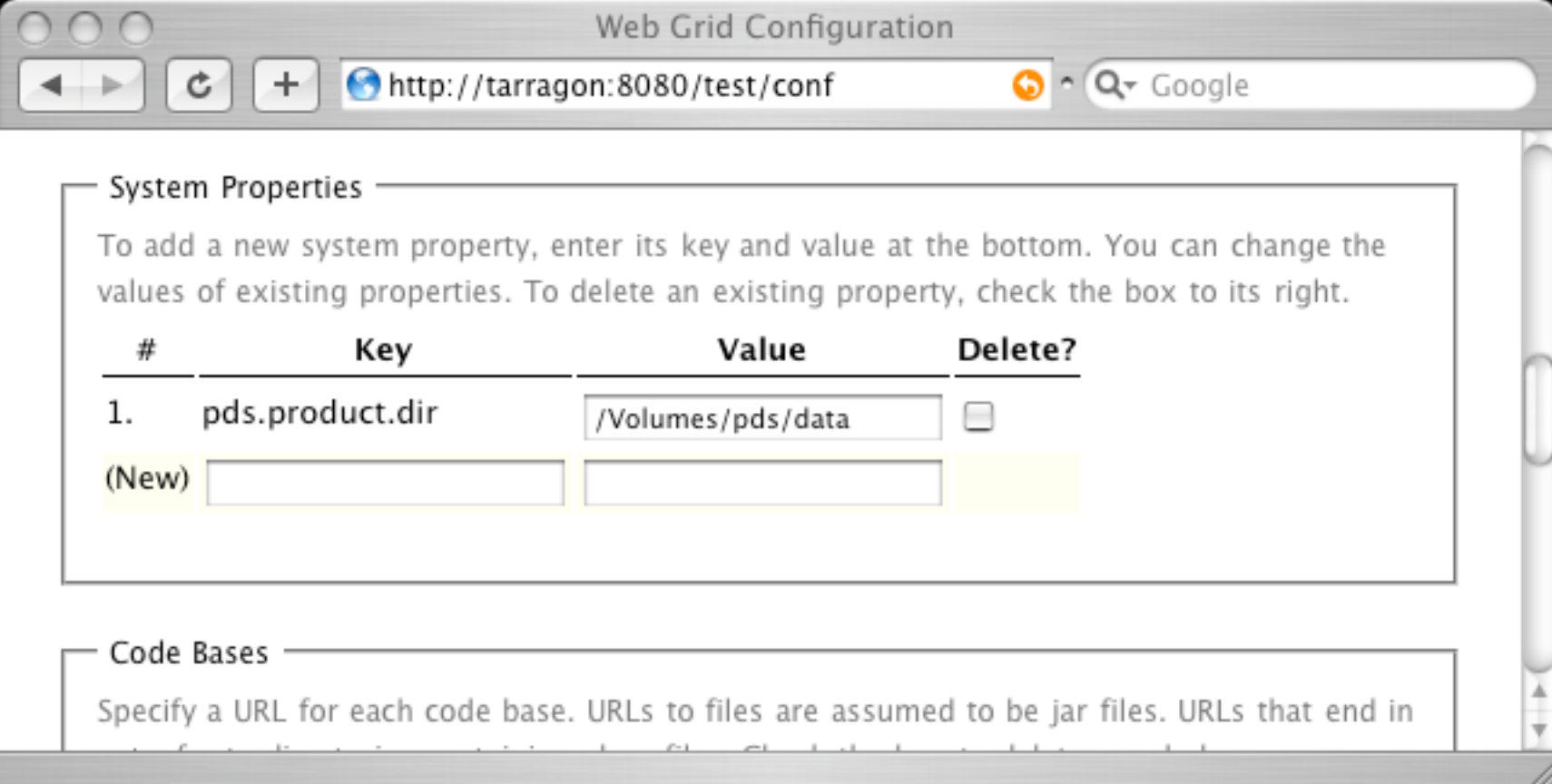
Adding a Property



The screenshot shows a web browser window with the title "Web Grid Configuration". The address bar contains the URL "http://tarragon:8080/test/conf". Below the browser window, there is a section titled "System Properties" with a sub-header "System Properties". The text below the sub-header reads: "To add a new system property, enter its key and value at the bottom. You can change the values of existing properties. To delete an existing property, check the box to its right." Below this text is a table with four columns: "#", "Key", "Value", and "Delete?". The first row of the table is highlighted in yellow and contains the following data: "# (New)", "Key pds.product.dir", "Value /Volumes/pds/data", and "Delete?" with an unchecked checkbox.

#	Key	Value	Delete?
(New)	<input type="text" value="pds.product.dir"/>	<input type="text" value="/Volumes/pds/data"/>	<input type="checkbox"/>

Updating Properties



The screenshot shows a web browser window with the title "Web Grid Configuration". The address bar contains the URL "http://tarragon:8080/test/conf". The page content is divided into two sections: "System Properties" and "Code Bases".

System Properties

To add a new system property, enter its key and value at the bottom. You can change the values of existing properties. To delete an existing property, check the box to its right.

#	Key	Value	Delete?
1.	pds.product.dir	<input type="text" value="/Volumes/pds/data"/>	<input type="checkbox"/>
(New)	<input type="text"/>	<input type="text"/>	<input type="checkbox"/>

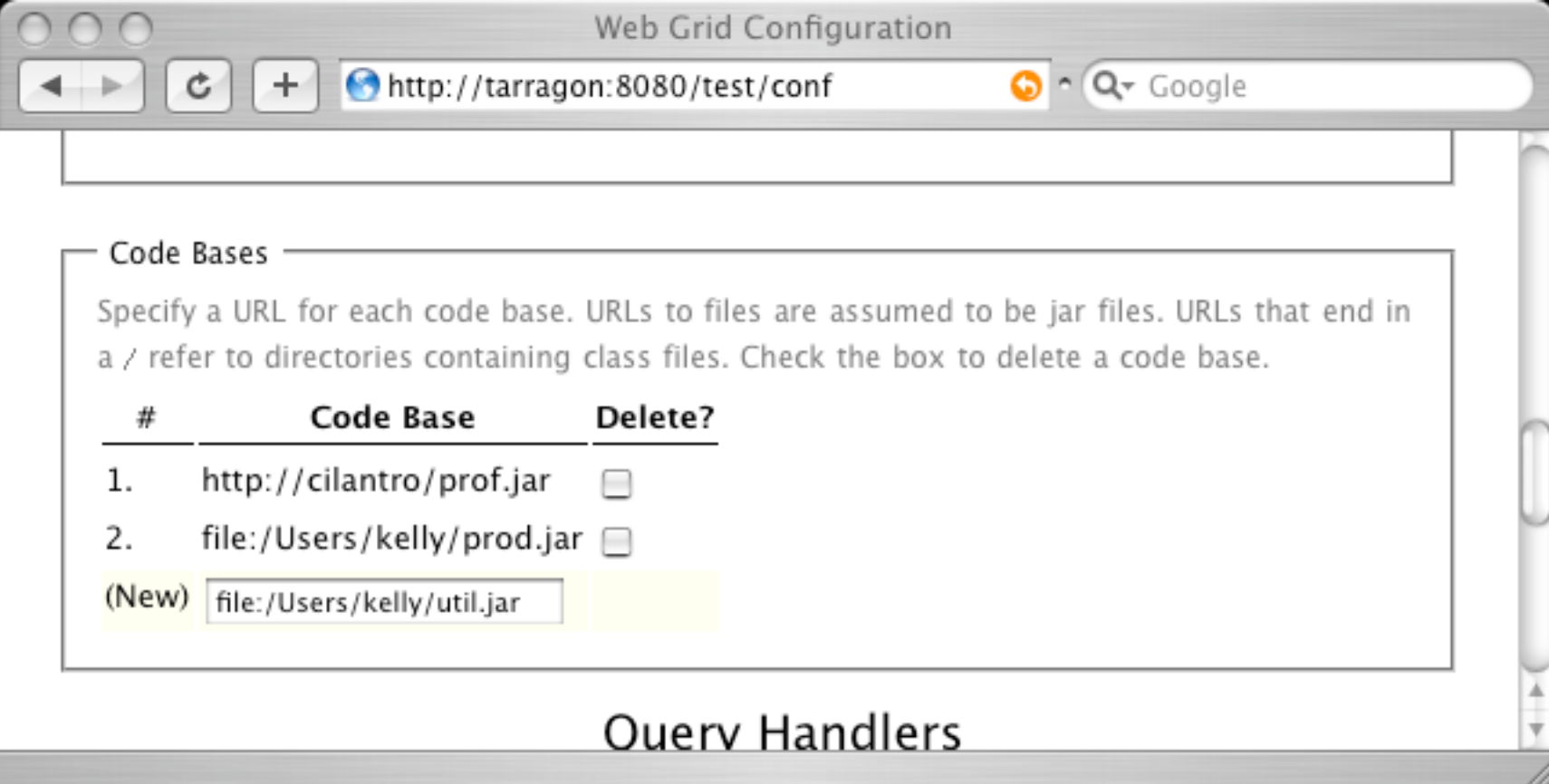
Code Bases

Specify a URL for each code base. URLs to files are assumed to be jar files. URLs that end in

Setting Class Path

- Web grid includes basic OODT framework
 - But no profile and product handlers
- You specify class path
 - By listing “code bases”
 - Code bases are URLs to JAR files
 - Or to directories containing class files
- URLs
 - Can be remote, not residing on server system

Specifying Code Bases



Web Grid Configuration

http://tarragon:8080/test/conf

Code Bases

Specify a URL for each code base. URLs to files are assumed to be jar files. URLs that end in a / refer to directories containing class files. Check the box to delete a code base.

#	Code Base	Delete?
1.	http://cilantro/prof.jar	<input type="checkbox"/>
2.	file:/Users/kelly/prod.jar	<input type="checkbox"/>
(New)	<input type="text" value="file:/Users/kelly/util.jar"/>	

Query Handlers

Pre-Packaged Code Bases

- Don't specify them via the web
 - Or in `WEB-INF/config.xml`
- Include them in `WEB-INF/lib`

Specifying Query Handlers

- Give class names to product and profile handlers
 - Or for pre-packaged releases, list them in WEB-INF/config.xml

Specifying Handlers

The screenshot shows a web browser window titled "Web Grid Configuration" with the address bar containing "http://tarragon:8080/test/conf". The page is divided into two main sections:

- Product Query Handlers:** A panel with a title bar and a text instruction: "Specify the class name for each product query handler; the class must implement the `jpl.eda.product.QueryHandler` interface." Below this is a table with three columns: "#", "Class Name", and "Delete?". The first row is highlighted in yellow and contains "(New)", "jpl.pds.FileQueryHandler", and a yellow square.
- Profile Query Handlers:** A panel with a title bar and a text instruction: "Specify the class name for each profile query handler; the class must implement the `jpl.eda.profile.handlers.ProfileHandler` interface." Below this is a table with three columns: "#", "Class Name", and "Delete?". It contains three rows: 1. "1.", "jpl.pds.MarsProfileHandler", and a checkbox; 2. "2.", "jpl.pds.MoonProfileHandler", and a checkbox; 3. "(New)", an empty text input field, and a yellow square.

At the bottom left of the browser window, there is a "Save Changes" button.

Make Queries

- Profile Queries
 - Use `jpl.eda.profile.ProfileClient`
 - Object name = `http://host/appname/prof`
- Product Queries
 - Use `jpl.eda.product.ProductClient`
 - Object name = `http://host/appname/prod`

Or Use a Browser!

- Profile queries
 - `http://host/appname/prof?q=expr`
 - Response is always text/xml list of matching profiles
 - Or empty `<profiles/>`
- Product queries
 - `http://host/appname/prod?q=expr`
 - Response is matching product
 - Or 404 Not Found
 - MIME type depends on product

Request Parameters

- The “q”
 - DIS-style keyword query
 - Web grid makes an XMLQuery out of it
- The “xmlq”
 - A complete XMLQuery, in XML format
- If both are specified, xmlq is preferred

Pre-Packaged WARs

- To make a pre-packaged, pre-configured WAR for a specific application:
 - Extract web-grid-1.1.0.war
 - Customize WEB-INF/config.xml
 - Add JARs to WEB-INF/lib
 - Customize welcome page index.html
 - Package it up into a new WAR
- Example: see pds-web-ps in CVS

Performance

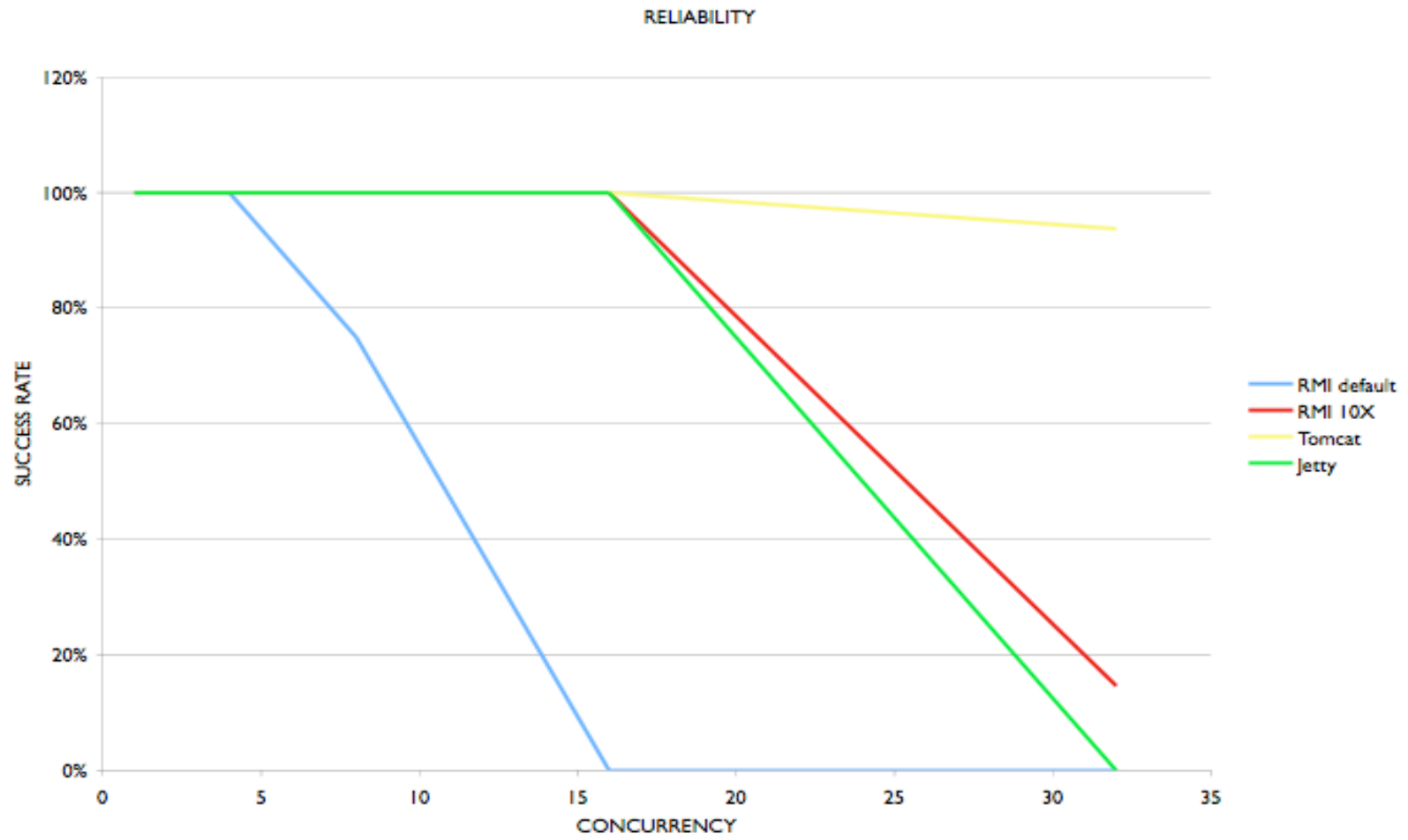
- System configuration
 - Server: Power Mac 2.0GHz dual G5, Mac OS X
 - Client: Mac Mini 1.42GHz G4, Mac OS X
 - 100baseTX fast Ethernet
- Tested configurations
 - RMI (default block size, 4096 bytes)
 - RMI (10X block size, 40960 bytes)
 - Web Grid under Tomcat 5.5.9
 - Web Grid under Jetty 5.1.4

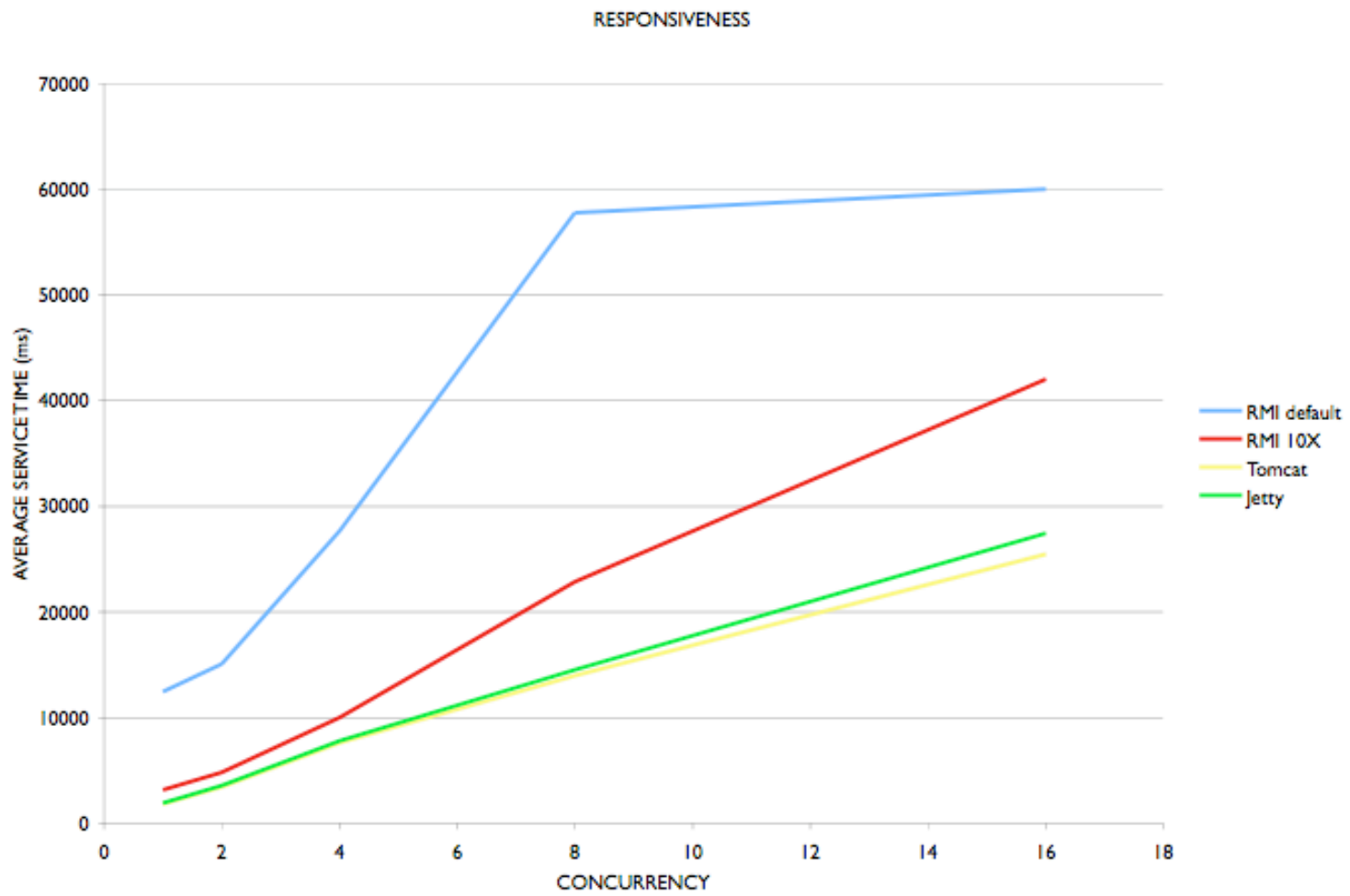
Performance

- Tested handlers: PDS File Query Handler
 - Retrieve a DIRLIST I
 - Retrieve a PDS_ZIPD of a small directory
 - Retrieve a PDS_ZIPD_SIZE
 - Extract a label
 - Retrieve a large raw file
- Tested concurrency: 1, 2, 4, 8, 16, 32, ... queries at a time
- Time limit: 60 seconds for any one query to finish

Test Results

- Executive summary:
 - Web Grid under Tomcat kicks butt
 - Followed by Web Grid under Jetty
 - Followed by RMI with 10X block size
 - Followed by RMI with default block size
 - (CORBA will always suck)





Recomendation

- Web Grid provides
 - Higher reliability
 - Better performance
 - Less administration
- Deploy it
 - For ERNE
 - For PDS
 - Everywhere

Summary

- Old RPC-based approach
- Web grid
 - Deploying
 - Configuring
 - Using
- Performance