

Understanding Web Services based on

<https://jax-rpc.dev.java.net/whitepaper/1.1/index-part1.html>
<http://java.sun.com/webservices/docs/1.3/tutorial/doc/JAXRPC5.html>

B. Ramamurthy

1

9/24/2004

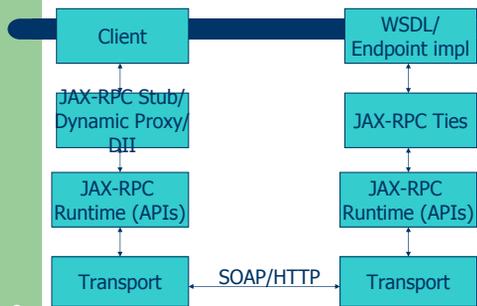
Web Services Support Tools

- Java WS Developer Package: (JWS DP) integrated developer package to develop, test, and deploy XML applications, WS, Web applications.
- For enterprise class products of WS you may want to use Sun One Studio, or any J2EE server products.

2

9/24/2004

Application Architecture



3

9/24/2004

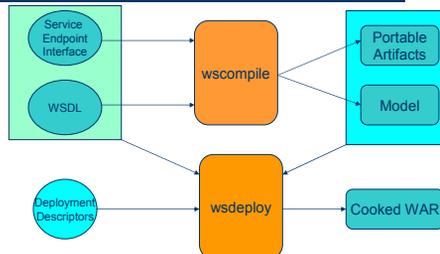
Develop/Deploy (DD) Cycle (server side)

- Develop/deploy cycle uses wscompile and wsdeploy tools to create "artifacts"
- Wscompile is invoked on a service endpoint interface or WSDL to generate artifacts specified by JAX-RPC specs and a model file.
- A model is a JAX-RPC representation of a web service generated by wscompile, usually in .xml.gz file.
- Artifacts generated by wscompile are bundled together to form a war file along with a deployment descriptor (raw WAR file). Raw WAR file contains portable artifacts.
- wsdeploy tool takes this raw WAR file as an input, processes it and generates a WAR file (cooked WAR file) which can be deployed in a servlet container, exposing web service and associated WS to a client. Cooked WAR file contains implementation-specific artifacts.

4

9/24/2004

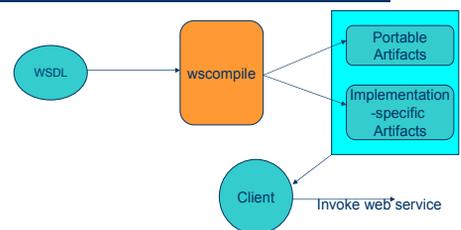
DD Cycle



5

9/24/2004

Invocation Model (client side) or Invoke Cycle



6

9/24/2004

wscompile

- Tool to generate client-side and server-side artifacts (support files) required by JAX-RPC.

- It can run from the shell using

```
wscompile [options] config_file
```

Ex:

```
wscompile -gen -classpath lib/foo.jar;/bar.jar -generated config.xml
```

Ant task for wscompile:

```
<wscompile
  gen = true
  base = "${build}"
  classpath = "xyz.jar"
  config = "config.xml">
```

7

9/24/2004

config.xml

- Configuration file is the primary source of information for wscompile tool.
- The xml file has elements for service, wsdl, modelfile etc.
- End point definitions for server and port number are also specified.

8

9/24/2004

wsdeploy tool

- wsdeploy tool is used to generate a WAR file that can be deployed in a container such as Tomcat, J2EE, JRun4, JBoss.
- It takes a raw WAR file that contains a service endpoint interface, a service endpoint implementation, any value types, any service-specific exceptions, a model file, a deployment descriptor and generates an implementation-specific cooked WAR file.
- This generates serializers, ties, runtime descriptors, other files needed for runtime deployment.

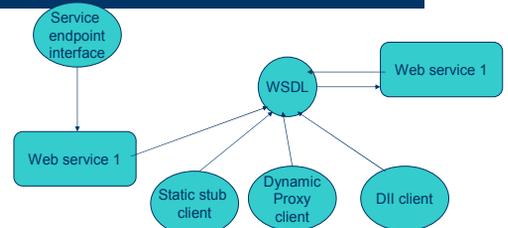
- Usage:

```
wsdeploy -o target.war myapp.war
```

You will use an ant task equivalent.

9

Client Scenarios



10

9/24/2004

Static Stub Client code

- Create a stub object

```
stub = (Stub) (new
  MyHelloService_Impl().getHelloIFPort())
```

Set endpoint address: stub._setProperty ...

Cast Stub to endpoint interface address

```
HelloIF hello = (HelloIF) stub
```

Invoke the method on hello object.

11

9/24/2004

Building Static Stub Client

- Ant tasks:
generate-stubs
compile-client
package-client

- Generate-stubs uses wscompile command given below:

```
wscompile -gen:client -d build -classpath config-wsdl.xml
```

- You may also use "ant run" for an java application client.

12

9/24/2004