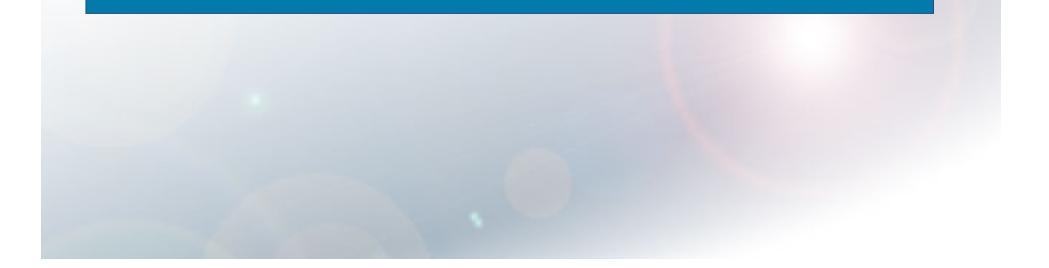
# Introduction to Oracle WebLogic

Presented by: Fatna Belqasmi, PhD, Researcher at Ericsson



# Agenda

- Overview
- Download and installation
- A concrete scenario using the real product
- Hints for the project



# Overview



 Oracle WebLogic Web Server

 Oracle Workshop for WebLogic

# Oracle WebLogic Web Server

- Is a Java Enterprise Edition (Java EE) application server
- Supports the deployment of many types of distributed applications
- Provides a standard set of APIs for creating distributed Java applications that can access a wide variety of services
  - Web Services

....

- Web applications (HTML, Java ServerPages-JSP, ..)
- Remote Method Invocation (RMI)
- Java Database Connectivity (JDBC)

# Oracle WebLogic Web Server

- WebLogic Server 10.3 supports two types of Web Service APIs
  - Java API for XML based Remote Procedure Call (JAX-RPC)
    - Defines the Java APIs for making XML-based remote procedure calls (RPC)
  - Java API for XML based Web Services (JAX-WS)
    - Is a standards-based API for coding, assembling, and deploying Java Web Services
    - Is the successor to the JAX-RPC

# Oracle WebLogic Web Server

- WebLogic Server 10.3 supports the following Web Service standards
  - JAX-WS 2.1
  - JAX-RPC 1.1
  - Web Services for Java EE 1.2
  - Web Services Metadata for the Java Platform 2.0 (JSR-181)
  - Web Services Description Language (WSDL) 1.1
  - Simple Object Access Protocol (SOAP) 1.1 and 1.2
  - Web Services Security (WS-Security) 1.1
  - Universal Description, Discovery, and Integration (UDDI) 2.0

# Oracle Workshop for WebLogic

- Is a set of plug-ins to the Eclipse Integrated Development Environment (IDE) platform
  - Many of the standard features of the workshop are described in the Eclipse documentation, available at <a href="http://eclipse.org">http://eclipse.org</a>
- Allows quick and easy creation, deployment and testing of enterprise applications:
  - Web Services
  - Java
  - Web Applications
  - ...
- The applications can be deployed on a wide variety of servers, including
  - WebLogic Server
  - Tomcat,
  - JBoss, and others

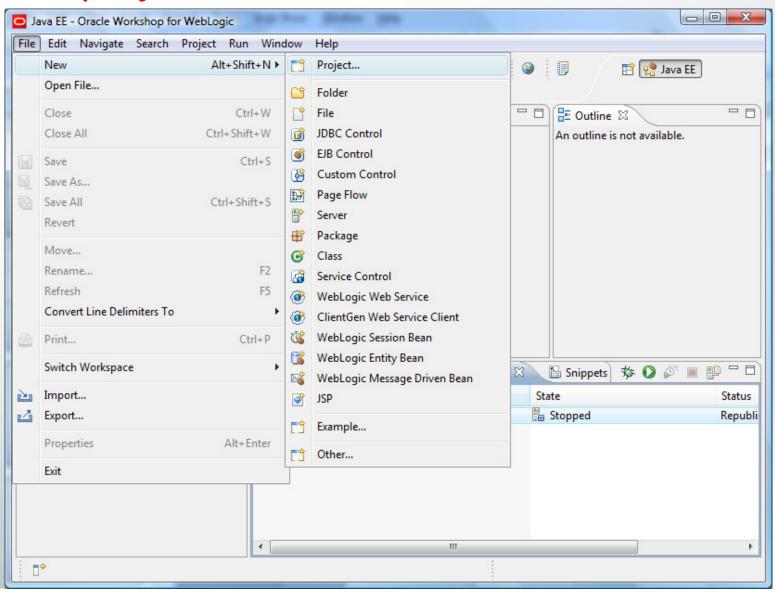
# Download and installation

- Download
  - Free download with registration
  - http://www.oracle.com/technology/software/products/ias/bea\_main.html
  - Accept the license agreement on the top of the page
  - Click the link:
    - "Oracle Workshop for WebLogic 10.3 Package Installer" OR
    - "Oracle Workshop for WebLogic 10.3 Net Installer"
  - Sign in or register if you are new
- Installation
  - Double click the downloaded file and follow the instructions
    - Documentations:
      - <u>http://e-docs.bea.com/wlw/docs103/index.html</u>
      - http://e-docs.bea.com/wls/docs103/webservices.html

# A Concrete Scenario Using the Real Product

- A simple "Hello World" web service
  - Create a Web Service
    - Create project
    - Create package
    - Create a Web Service
    - Add a web service method
  - Add/start weblogic server
    - Create a domain
    - Start server
  - Deploy and test the Web Service
    - Run on server
    - Web logic test client
      - See SOAP, WSDL

#### Create a project



• Create a project

New Project	
Select a wizard Create a Web Service Project.	
<u>N</u> izards:	
type filter text	
설 Java Project 훑 Java Project from Existing Ant Buildfile ﷺ Plug-in Project	
<ul> <li>General</li> <li>EJB</li> <li>Des J2EE</li> </ul>	ш
<ul> <li>&gt; Dava</li> <li>&gt; Devices</li> </ul>	
Web Service Project  Workshop Quick Start Applications  Constant	-
Show All Wizards.	
(?) < <u>Back</u> <u>Next</u> > <u>Finish</u> Cancel	

#### • Create a project

New Web Service Project
Web Service Project
Create a Web Service project in the workspace or at an external location.
Project name: HelloWorldProject
Project contents:
Use default
Directory: C:\bea\user_projects\workspaces\default\HelloWorldProject Browse
Target Runtime
Oracle WebLogic Server v10.3
Configurations
Annotated Web Service Facets JAX-RPC (Recommended) (v10.3)
Support for creating web services with the Web Services Metadata (JSR-181) standard. Additionally supports using WebLogic web service extensions, Apache Beehive Controls, and Apache XMLBeans.
EAR Membership
C Add project to an EAR
EAR Project Name: EAR Vew
(?) < Back Next > Finish Cancel

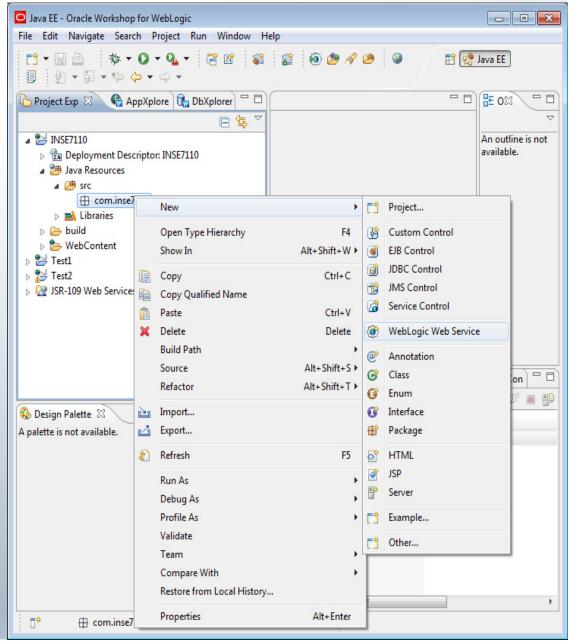
### • Create a new package

Java EE - Oracle Wor	Search Project Run Window H	lelp		
	🎄 🕶 🔕 🕶 📑 🛙		l 🔊 🖉 🖉 🖉 🗐	🖹 😭 Java EE
占 Proje 🛛 🛛 🎧 Ap	opX 🔂 DbXp 🗖 🗖			
A 🛃 HelloWorldPr	New	۰.	Project	outline is not ilable.
HelloworldPr HelloworldPr Deployme	Show In	Alt+Shift+W ▶	📫 File	
<ul> <li>Description</li> <li>Descript</li></ul>	Сору	Ctrl+C	🖆 Folder	
	Copy Qualified Name		🚱 Custom Control	
0	Paste	Ctrl+V	SIB Control	
1	K Delete	Delete	JDBC Control	
	Build Path		JMS Control	
	Refactor	Alt+Shift+T ►	Service Control	
	Import	•	WebLogic Web Service	
	Export	•	@ Annotation	
4	Refresh	F5	Class	
	Close Project		🕼 Enum	
	Close Unrelated Projects		🗊 Interface	
Design Palette 🖾	Rebuild AppXRay Database		🖶 Package	
palette is not availal	Enable AppXRay		Source Folder	
	Validate		🚳 Dynamic Web Project	
	Run As		Dynamic Web Project from Existing	Source Status
	Debug As	•	Enterprise Application Project	Repub
	Profile As	•	MTML	
	Team	•	JSP 🧐	
	Compare With	•	🕆 Server	
	Restore from Local History		😚 Servlet	
	Source	•	😚 Example	
<u></u> ]∲ 🛃 He	Java EE		Cther	
	Properties	Alt+Enter		

• Create a new package

🖸 New Java Pa	ckage	
J <b>ava Packa</b> Create a Java p		
Creates folders	corresponding to packages.	N2
Source folder:	HelloWorldProject/src	Browse
Name:	hello.com	
?	Finish	Cancel

- Create a Web Service
  - Specify the names and parameters of all of the service's exposed operations
- Steps
  - Create a WebLogic
     WS
  - Add the methods
  - Configure the methods' parameters



• Create a Web Service

New Web Service	_ • • <b>×</b>
Web Service Create a new web service.	
Enter or select the parent folder:	
HelloWorldProject/src/hello/com	
<ul> <li>Image: Image: HelloWorldProject</li> <li>Image: Src</li> <li>Image: Hello</li> <li>Image: HelloWorld</li> </ul>	
0	Finish Cancel

#### • Create a Web Service

🖸 Java EE - INSE7110/src/com/inse7110/HelloWorld.java - Oracle Workshop for WebLogic						
File Edit Navigate Search Project Run Window Help						
□ • □     □ <td< td=""><td>😭 🐵 🤌 🞱 🔋 📑 😭 Java EE</td></td<>	😭 🐵 🤌 🞱 🔋 📑 😭 Java EE					
🚹 Projec 🛛 🕼 AppX 🔂 DbXpl 🖓 🗖 🐼 HelloWorld.java [Designer]						
<ul> <li>INSE7110</li> <li>Deployment Descriptor: INSE7110</li> <li>Java Resources</li> <li>Java Resources</li> <li>Com.inse7110</li> <li>Libraries</li> <li>Libraries</li> <li>WebContent</li> <li>Test1</li> <li>Test2</li> <li>SR-109 Web Services</li> </ul>	e ↓ 2 × × • ↓ com.inse7110 ↓ import declar ⊖ telloWorld • hello()					
Design View Source View						
🚯 Design Palette 🛛 🦳 🗖 🗖 🔝 Problems 🖉 Tasks 🔲 P	roperties 👫 Servers 🖾 🔚 Snippets 🚍 Console 📃 🗖					
□◆						

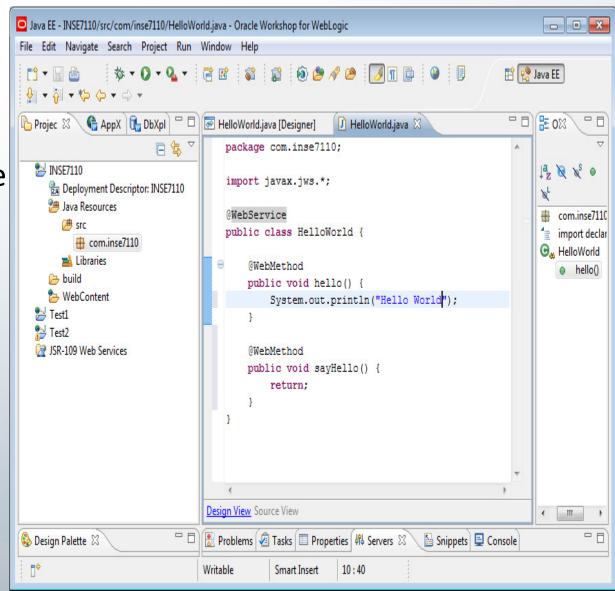
#### Create a Web Service

🖸 Java EE - INSE7110/src/com/inse7110/HelloWorld.java - Oracle Workshop for WebLogic					
File Edit Navigate Search Project Run	Window Help				
★     ↓       ★     ↓       ↓ </td <td>D - Q -   ≧ ⊠   S   S   S   S   S   S   S   S   S  </td> <td>🤌 Java EE</td>	D - Q -   ≧ ⊠   S   S   S   S   S   S   S   S   S	🤌 Java EE			
Projec 🛛 🏾 🏠 AppX 🔂 DbXpl 🖵 🗖	🐼 HelloWorld.java [Designer] 🖾 👘 🗖				
<ul> <li>INSE7110</li> <li>Deployment Descriptor: INSE7110</li> <li>Java Resources</li> <li>src</li> <li>com.inse7110</li> <li>Libraries</li> <li>build</li> <li>WebContent</li> <li>Test1</li> <li>Test2</li> <li>JSR-109 Web Services</li> </ul>	O HelloWorld Web Service         hello         Edit Source         New Web Method         New Callback         New Control Reference         Print         Preferences         Run As         Debug As         Profile As         Validate         Upgrade Source File(s)         Team         Compare With         Replace With	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓			
🚯 Design Palette 🛛 🗖 🗖	Problems 🕢 Tasks 🔲 F Web Services				
	Insert				

#### • Create a Web Service

🖸 Java EE - INSE7110/src/com/inse7110/HelloWorld.java - Oracle Workshop for WebLogic					
File Edit Navigate Search Project Run	Window Help				
1     - </td <td>) - 9 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)</td> <td>Java EE</td>	) - 9 (2) (2) (2) (2) (2) (2) (2) (2) (2) (2)	Java EE			
Projec 🛛 🌘 AppX 🔂 DbXpl 🗖 🗖	🐨 *HelloWorld.java [Designer] 🛛 🚺 *HelloWorld.java 🦳 🗖	10 × 0			
<ul> <li>INSE7110</li> <li>Deployment Descriptor: INSE7110</li> <li>Java Resources</li> <li>src</li> <li>com.inse7110</li> <li>Libraries</li> <li>build</li> <li>WebContent</li> <li>Test1</li> <li>Test2</li> <li>JSR-109 Web Services</li> </ul>	HelloWorld Web Service     hello     sayHello	↓ a b com.inse7110 a import declar G HelloWorld e hello()			
	Design View <u>Source View</u>	۰ III +			
🗞 Design Palette 🛛 🗖 🗖	🖹 Problems 🖉 Tasks 🔲 Properties 👭 Servers 🛛 🕞 Snippets 📮 Console	- 0			
□◆	Writable				

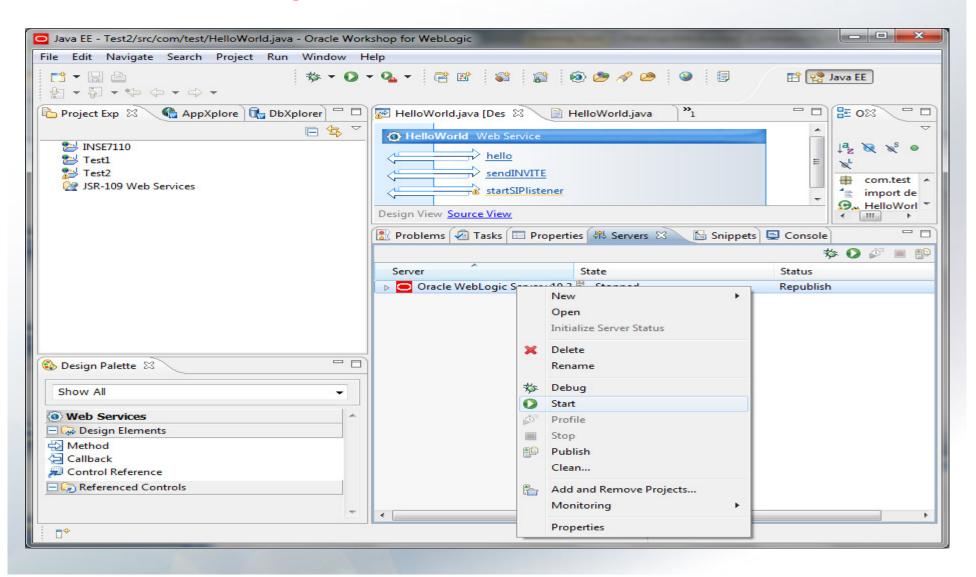
- Add the Web Service code
  - Implement the business logic of your Web Service
- Using both the Design and Source Views
- Programming language
   – Java



• Add a weblogic server

🖸 Java EE - Test2/src/com/test/HelloWorld.java - Oracle Workshop for WebLogic					
File Edit Navigate Search Project Run Window Help					
	<b>Q</b> -	🖹 😰 🗳 😭 🔗 😕	🔏 🙆 🔕 📘 🖉 🖻 😰	Java EE	
		New	🕆 Server		
Project Exp 🛛 🌑 AppXplore 🔂 DbXplorer 🗖 🗖	6	Open	d.java 💦 🗖 🗖	E 02 - D	
		Initialize Server Status			
🛃 INSE7110	×	Delete		Jªz 😿 🗙 ●	
Test1		Rename		×L	
₽d Test2 22 JSR-109 Web Services	蓉	Debug		🖶 com.test 🔺	
	0	Start		1 import de	
	j.	Profile		P <sub>≪</sub> HelloWorl ≡ o helloi	
		Stop		o sendI	
		Publish		· · · · ·	
🚯 Design Palette 🕱 📃 🗖	<b>E</b>	Clean	ervers 🛛 📔 Snippets 📃 Console		
	R-	Add and Remove Projects	3	🌣 🜔 🖉 🔳 😰	
Show All		Monitoring •	Status		
Web Services		Properties	ped Republish	h	
🖃 🥋 Design Elements	_		1		
Method -	•		III	•	

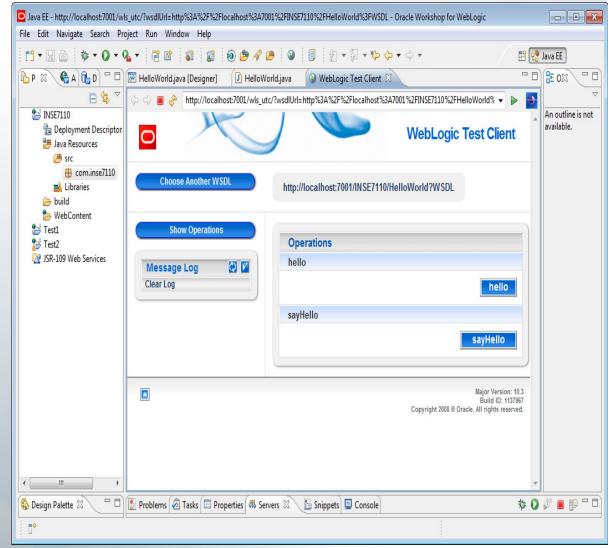
Start the weblogic server

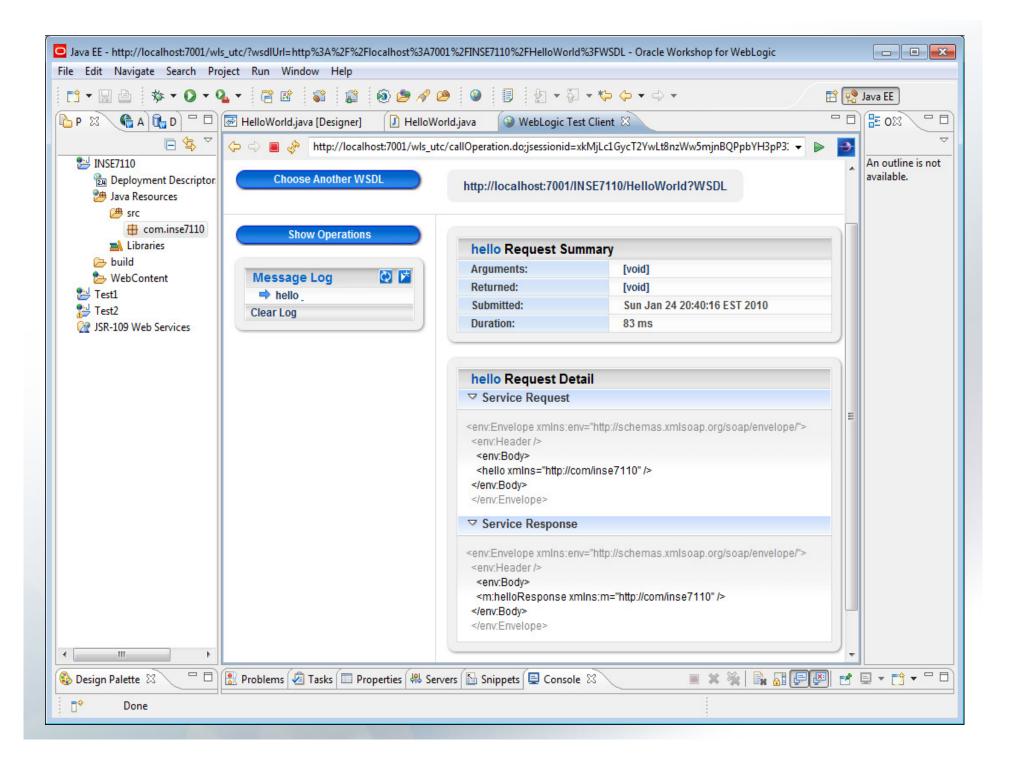


### • Deploy a Web Service

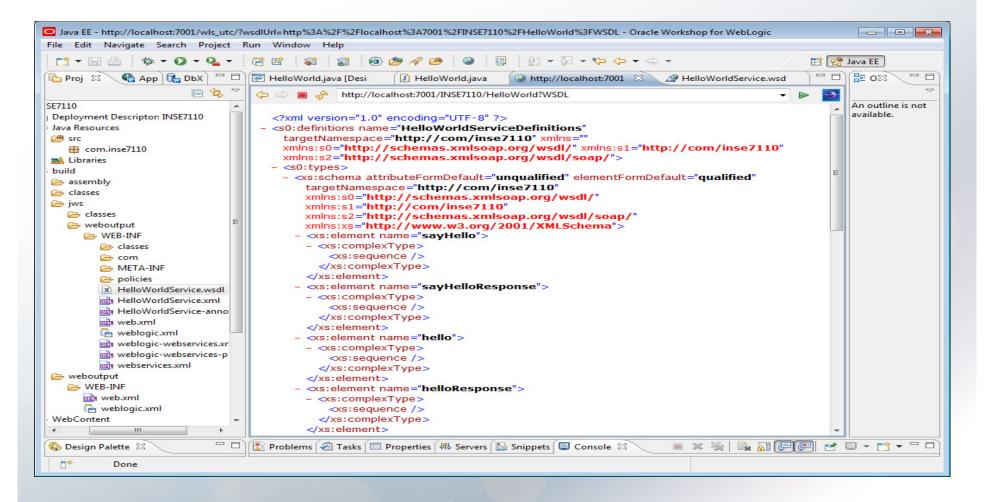
🖸 Java EE - INSE7110/src/com/inse7110/HelloWorld.java - Oracle Workshop for WebLogic								
File Edit Navigate Search Project Run Window Help								
□1 • □     □ • • • • • • • • • •     □2     0								
P X	🕒 P 🙁 🍓 A 🔂 D 🖓 🗗 🚺 HelloWorld.java 🖄 🎯 WebLogic Test Client 🎽 👘 🖓 🖓 🖓							
	package com.inse7110;							
⊿ ≥ INSE7		New	+					
⊳ 📴 D ⊿ 😕 Ja		Show In	Alt+Shift+W ►		tom.inse7110			
∠		Сору	Ctrl+C	orld {	🚖 import declar			
	e	Copy Qualified Name			⊖ <sub>≪</sub> HelloWorld			
Þ 🗁 bi	Ē	Paste	Ctrl+V	lo() {	hello()			
> 🍃 W	×	Delete	Delete	println("Hello World");				
D 22 Test1 D 22 Test2		Build Path	+					
⊳ 🐼 JSR-1		Refactor	Alt+Shift+T ►					
		Import	+	Hello() {				
		Export	+					
	8	Refresh	F5					
	-	Close Project						
		Close Unrelated Projects						
		Rebuild AppXRay Databa	re					
		Enable AppXRay	30					
		Validate	. [					
		Run As	•	1 Run on Server Alt+Shift+X, R				
		Debug As Profile As	•	Image: System 2     Java Applet     Alt+Shift+X, A       Image: Java Application     Alt+Shift+X, J				
		Team		3 Java Application Alt+Shift+X, J				
		Compare With		Open Run Dialog				
		Restore from Local Histor	D/	- F				
•		Source			< <u>ااا</u> ۲			
🚯 Design Pa		Java EE	*	erties 🙌 Servers 📔 Snippets 📮 Console 🖾				
₽		Properties	Alt+Enter					

- Test a Web Service
  - Using Test View
- Test View
  - Invoke a web service method from a browser
  - View the XML messages that are exchanged



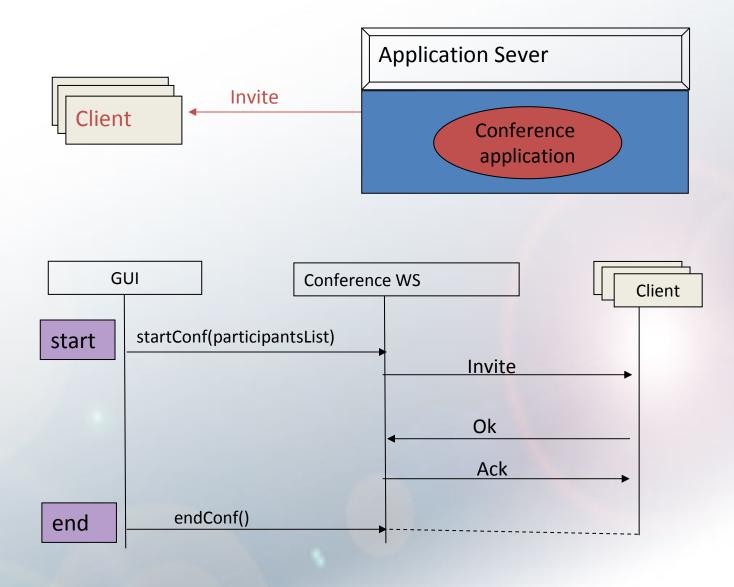


#### - View the WSDL file



# Hints for the project

# What to implement



# Hints for the project

- Server side:
  - a conferencing web service that implements a simplified conference focus
    - uses SIP stack to listen to, receive, process and send SIP messages
      - Use JAIN SIP
- Client side: SIP client
  - Download or
  - Code your client using SIP stack
- Use of JMF for media

# How to download JAIN SIP

- JSR 32 Jain SIP
- API doc and .jar can be downloaded at:
  - <u>http://jcp.org/aboutJava/communityprocess/mrel/jsr032/index.html</u>
  - Unzip
- Reference implementation .jar can be downloaded at:
  - <u>http://download.java.net/communications/jain-sip/nightly/</u>
  - Click: "jain-sip-ri/", you can find all nightly build implementations .jar
  - You can also find sdp implementations there...
- When implementing a web service using Jain SIP, put the two jar files (one api, one impl) into your project: \WEB-INF\lib
  - jsip\_api\_v1.2.jar
  - jain-sip-ri-1.2.xx.jar

- Tip: you may need log4j-xxx.jar if you always have running errors when using Jain SIP (this depends on which sip impl that you use)
  - <u>http://www.apache.org/dyn/closer.cgi/logging/log4j/1.2.15/apache-log4j-1.2.15.zip</u>

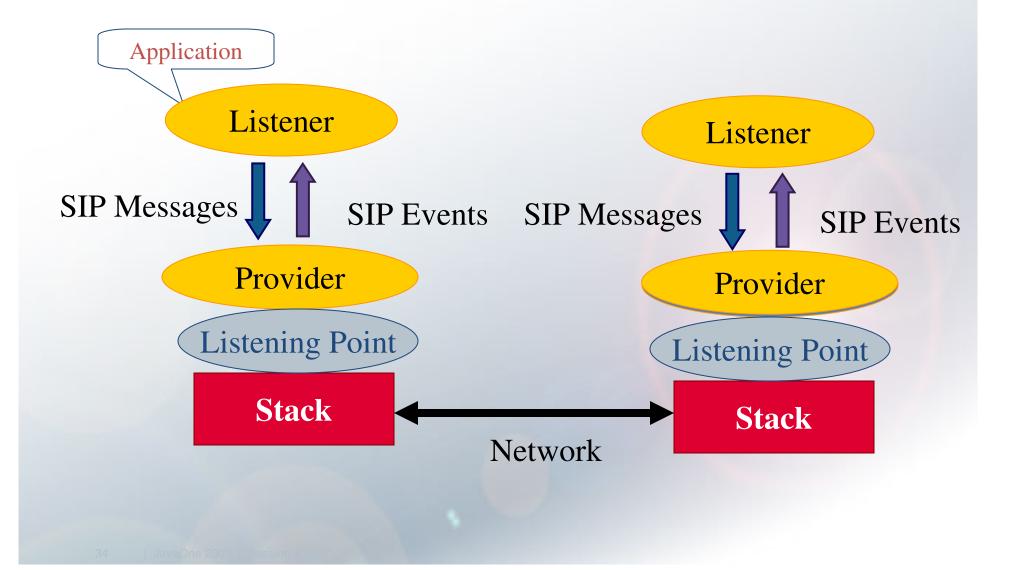
# A brief introduction to JAIN SIP

- JAIN-SIP is a java-standard interface to a SIP signaling stack
- It provides a low level Java API specification for SIP Signaling
- It was designed for the developers who require fine grained access to the SIP protocol
- It can be used in:
  - Clients
  - Servers
  - Proxies
- JAIN SIP reference implementation is open source, very stable, and very widely used
  - Introduction to JAIN SIP
    - <u>http://www.oracle.com/technology/pub/articles/dev2arch/2007/10/introduction-jain-sip.html</u>

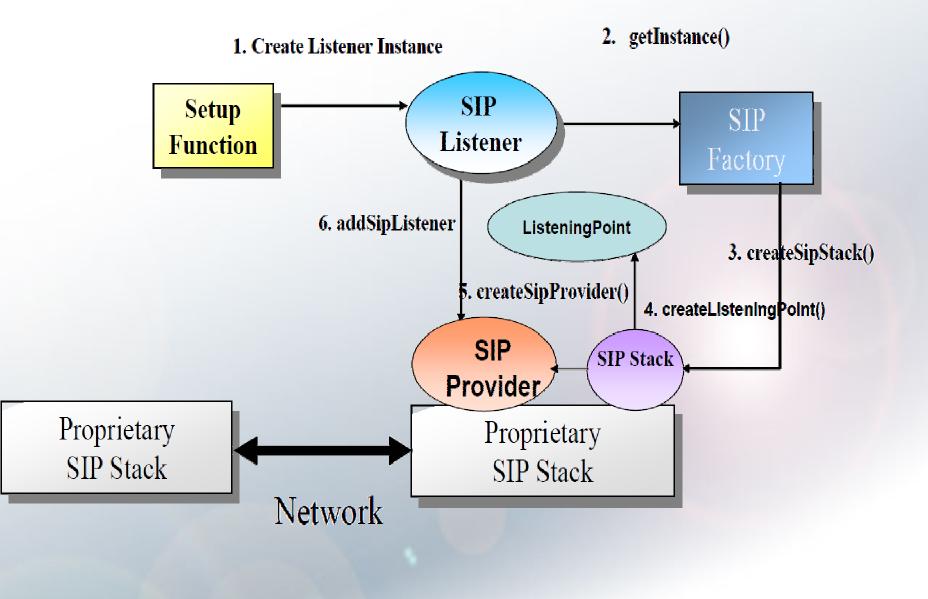
### A brief introduction to JAIN SIP

- JAIN SIP services
  - Provides methods to format and send SIP messages
  - Parses incoming messages and enable the application to access the messages' fields via a standardized JAVA interface
  - Invokes appropriate application handlers when appropriate (message arrivals, transaction time-outs)
  - Provide Transaction support and manage Transaction state and lifetime on behalf of a user application.
  - Provide Dialog support and manage Dialog state and lifetime on behalf on a user application

### **JAIN SIP Architecture**



# A brief introduction to JAIN SIP



# A brief introduction to JAIN SIP

#### **Basic steps:**

- **1. public class** MySipListener **implements** javax.sip.SipListener {...}
- 2. SipFactory sipFactory = SipFactory.getInstance();
- 3. SipStack sipStack = sipFactory.createSipStack(properties);
- ListeningPoint listeningPoint = sipStack.createListeningPoint(address, port, transport);
- 5. SipProvider sipProvider = sipStack.createSipProvider(listeningPoint);
- 6. sipProvider.addSipListener(mySipListener);

#### • Introduction to JAIN SIP

<u>http://www.oracle.com/technology/pub/articles/dev2arch/2007/10/introduction-jain-sip.html</u>

# SDS or Web Logic ?

- Ericsson SDS:
  - Need to configure IMS first...
  - Can use SIP Servlet
  - Trigger: from SIP client or HTTP Servlet
    - Both need programming
  - Conference client registration: register to CSCF

- Oracle Web Logic
  - No configuration is needed
  - Can use SIP Stack (lower level of abstraction than SIP Servlet)
  - Trigger: from web logic test client or any web service that talks SOAP
    - The first option is provided by the tool, no need to code
  - Conference client registration: register to your SIP Listener (in your WS logic)
- What's in common:
  - Media handling: RTP/RTCP (JMF)
  - Conference client: SIP client that talks RTP

# References

- http://download.oracle.com/docs/cd/E12840\_01/wls/docs103/intro/chap1.html
- http://download.oracle.com/docs/cd/E12840\_01/wls/docs103/webserv\_intro/overview.htm l#choose
- http://download.oracle.com/docs/cd/E12840\_01/wls/docs103/webserv\_intro/standards.ht ml#wp1078494
- http://www.oracle.com/technology/products/workshop/index.html
- http://e-docs.bea.com/wlw/docs103/index.html
- http://e-docs.bea.com/wls/docs103/webservices.html
- http://www.oracle.com/technology/pub/articles/dev2arch/2007/10/introduction-jainsip.html

