

Administrarea Bazelor de Date Managementul în Tehnologia Informației

Sisteme Informatice și Standarde Deschise (SISD)

2009-2010

Curs 11
Servicii web. Introducere J2EE*

*Preluare și adaptare din cursul de LPD 2008-2009



Continut

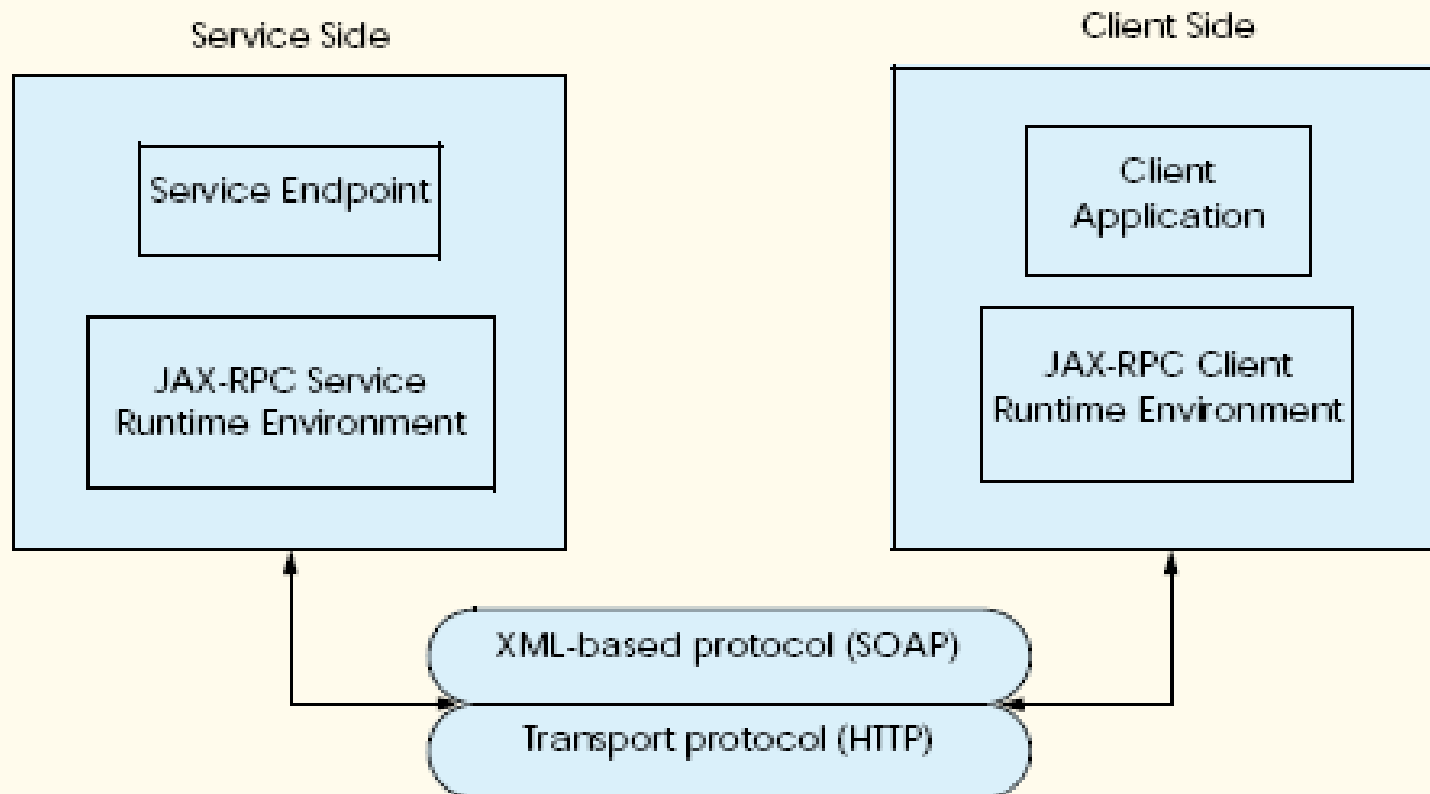
- Programarea serviciilor web – JAX-RPC
- Alte standarde pentru servicii web: UDDI, WSRF, WS-BPEL
- J2EE - Introducere



JAX-RPC - Introducere

- Java API for XML-based RPC
- Oferă mecanisme pentru crearea și accesarea serviciilor web
- Utilizează SOAP și HTTP
- Definieste mapări între tipurile de date Java și tipurile din WSDL
- Diferențe față de Java RMI:
 - Interoperabilitate
 - Limitare a tipurilor de date ce pot fi transmise
- <https://jax-rpc.dev.java.net/>

Arhitectura JAX-RPC [DEI02]





Tipuri de date suportate în JAX-RPC (1)

Tip Java	Tip XML / SOAP
boolean (Boolean)	xsd:boolean (soapenc:boolean)
byte (Byte)	xsd:byte (soapenc:byte)
double (Double)	xsd:double (soapenc:double)
float (Float)	xsd:float (soapenc:float)
int (Integer)	xsd:int (soapenc:int)
long (Long)	xsd:long (soapenc:long)
short (Short)	xsd:short (soapenc:short)

Tipuri de date suportate in JAX-RPC (2)

Tip Java	Tip XML / SOAP
BigDecimal	xsd:decimal
BigInteger	xsd:integer
Calendar	xsd:dateTime
Date	xsd:dateTime
String	xsd:string



Tipuri de date suportate in JAX-RPC (3)

- Prin JAX-RPC se pot transfera și obiecte din clase care îndeplinesc condițiile:
 - Nu implementează `java.rmi.Remote`
 - Au un constructor public fără argumente
 - Campurile publice sunt tipuri suportate de JAX-RPC
 - Sunt conforme cu modelul JavaBeans (au metode set/get)
 - Proprietățile bean-ului sunt tipuri suportate de JAX-RPC
- Tablouri uni- sau multi-dimensionale ale căror elemente sunt tipuri suportate de JAX-RPC



API-uri JAX-RPC

- Pentru server: nu exista
- Pentru client – in `javax.xml.rpc`:
 - **Stub** – trebuie sa fie implementata de stub-urile generate
 - **Call** – pentru invocare dinamica
 - **Service**
 - **ServiceFactory**
 - **JAXRPCException**

Stiluri pentru programarea clientilor

- **Static:** proxy generat pe baza descrierii WSDL a serviciului
- **Dynamic proxy:** prin interfata Stub
- **Dynamic Invocation Interface (DII)**
 - <http://users.skynet.be/pascalbotte/rcx-ws-doc/dii.htm>

Interfata Stub:

```
package javax.xml.rpc;
public interface Stub {
    // ...
    void _setProperty(String name, Object
value);
    Object _getProperty(String name);
    java.util.Iterator _getPropertyNames();
}
```

Exemplu Stub:

javax.xml.rpc.Stub

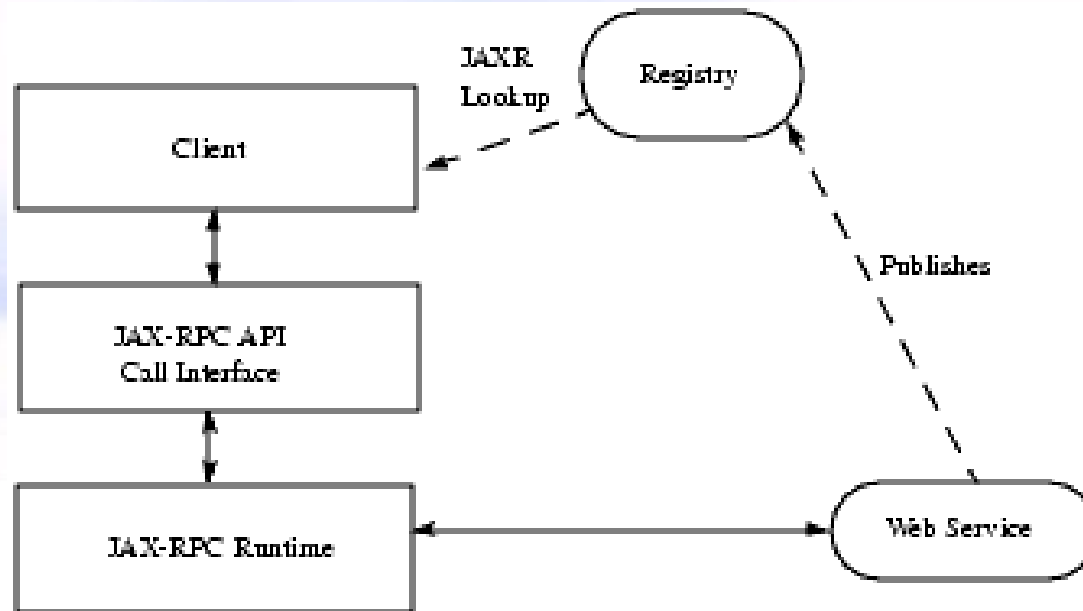
MyService

MyServiceSoapBinding_Stub



Moduri de apelare a metodelor

- *Synchronous request-response*: clientul invoca metoda și este blocat până primește răspunsul
- *One-way RPC*: metoda nu întoarce un rezultat clientului
- *Non-blocking RPC*: clientul invoca metoda și continuă procesarea; ulterior poate începe așteptarea blocantă sau poate interoga despre existența rezultatului





Exemplu – serviciu web simplu [Han02]

```
import java.util.*;

public class NHLService {
    HashMap standings = new HashMap();

    public NHLService() {
        // NHL - part of the standings as per 04/07/2002
        standings.put("atlantic/philadelphia", "1");
        standings.put("atlantic/ny islanders", "2");
        standings.put("atlantic/new jersey", "3");
        standings.put("central/detroit", "1");
        standings.put("central/chicago", "2");
        standings.put("central/st.louis", "3");
    }

    public String getCurrentPosition(String division, String team) {
        String p = (String)standings.get(division + '/' + team);
        return (p == null) ? "Team not found" : p;
    }
}
```



Exemplu – client pentru serviciul web [Han02]

```
package hansen.playground;
import org.apache.axis.client.Call;
import org.apache.axis.client.Service;
import javax.xml.rpc.namespace.QName;
import java.net.*;
public class NHLServiceClient {
    public static void main(String [] args) throws Exception {

        Service service = new Service();
        Call call = (Call)service.createCall();
        String endpoint =
"http://localhost:8080/axis/NHLService.jws";
        call.setTargetEndpointAddress(new URL(endpoint));
        call.setOperationName(new QName("getCurrentPosition"));
        String division = args[0];
        String team = args[1];
        String position =
            (String)call.invoke(new Object [] {new String(division),
new String(team)});

        System.out.println("Got result : " + position);
    }
}
```



Continut

- Programarea serviciilor web – JAX-RPC
- Alte standarde pentru servicii web: UDDI, WSRF
- J2EE - Introducere



UDDI

- Universal Description, Discovery and Integration
- Grup de specificații prin care:
 - furnizorii de servicii pot publica informații despre acestea
 - clienții pot căuta serviciul dorit
- Registru (registry) UDDI: locul unde sunt publicate informațiile despre serviciile mai multor furnizori
- Listing: o intrare dintr-un registru (conține informații despre un serviciu)



Listing-uri UDDI

- Specificate prin WSDL
- Elemente:
 - White Pages – informații de baza (modul de contactare a companiei care furnizează serviciul)
 - Yellow Pages – informații detaliate despre companie (coduri de identificare, capacități)
 - Green Pages – informațiile necesare pentru a utiliza serviciul (descrierea interfețelor, URL-uri etc.)
- Nu este obligatoriu ca un listing să conțină toate cele 3 elemente

Exemplu – listing UDDI [UDDI06] (1)

```
<?xml version="1.0" encoding="utf-8" ?>
<businessEntity xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
businessKey="5774466b-089d-4fa8-b8cb-fa2ead5329c5"
operator="Microsoft Corporation"
authorizedName="RealNames UDDI Publisher"
xmlns="urn:uddi-org:api">
<discoveryURLs>
  <discoveryURL
useType="businessEntity">http://uddi.microsoft.com/discovery?
businessKey=5774466b-089d-4fa8-b8cb-fa2ead5329c5</discoveryURL>
</discoveryURLs>
  <name>FreePsychicNetwork.com</name>
  <description xml:lang="en">
This is a UDDI Business Registry entry for
"FreePsychicNetwork.com".</description>
<contacts>
<contact useType="Main">
  <description xml:lang="en">Main contact information</description>
  <personName>Alma E Mounger</personName>
  <phone useType="Main">4079441169</phone>
  <email useType="Main">jimmyrayswindle@hotmail.com</email>
```




Exemplu – listing UDDI [UDDI06] (2)

```
<address sortCode="" useType="Main">
  <addressLine>2775 Wells Ct</addressLine>
  <addressLine>Kissimmee, FL 34744</addressLine>
  <addressLine>FL</addressLine>
</address>
</contact>
<contact useType="Technical">
  <description xml:lang="en">Technical contact
information</description>
  <personName>Alma E Mounger</personName>
  <phone useType="Technical">4079441169</phone>
  <email useType="Technical">jimmyrayswindle@hotmail.com</email>
<address sortCode="" useType="Technical">
  <addressLine>2775 Wells Ct</addressLine>
  <addressLine>Kissimmee, FL 34744</addressLine>
  <addressLine>FL</addressLine>
</address>
</contact>
...
</contacts>
</businessEntity>
```



WSRF - Introducere

- Problema: multe servicii necesita retinerea unor informatii de stare (stateful) – numarul de produse din stoc, evidenta rezervarilor pentru clienti etc.
- Specificatiile initiale pentru serviciile web nu includ informatii de stare
- Solutie initiala: stocarea informatiilor in fisiere / baze de date
- Solutie noua (2004): specificatia **WSRF (Web Services Resource Framework)** – implementata in Globus Toolkit 4



WS-Resource

- Asocierea dintre un serviciu web și o resursă
- Definită prin asocierea unui document XML și a unui portType pentru un serviciu web
- *Implied Resource Pattern*: set de convenții pentru realizarea asocierii
- Un serviciu poate utiliza mai multe resurse
- Identificarea unei resurse în cadrul serviciului: prin intermediul *WS-Addressing*



Modelul de adresare WS-Addressing

EndpointReference:EndpointReferenceType

Address:AttributedURI

ReferenceProperties:ReferencePropertiesType

ReferenceParameters:ReferenceParametersType

PortType:AttributedQName

ServiceName:ServiceNameType

- mod de a adresa serviciile web independent de protocolul de transport
- implementat cu ajutorul header-elor SOAP
- definit prin fișiere .xsd

Exemplu -Fragment din definiția XSD:

```
<xs:complexType name="ReferencePropertiesType">
  <xs:sequence>
    <xs:any processContents="lax" minOccurs="0"
      maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

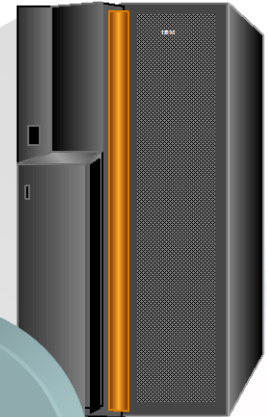
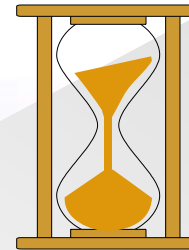


Specificatii WSRF

- WS-ResourceLifetime – mecanisme pentru distrugerea resurselor
- WS-ResourceProperties – mecanisme pentru obtinerea, modificarea, stergerea proprietatilor resurselor
- WS-RenewableReferences – obtinerea de update-uri pentru referinte
- WS-ServiceGroups
- WS-BaseFaults – tip XML pentru specificarea erorilor

Modelul WSRF (Grid Initiative 2006)

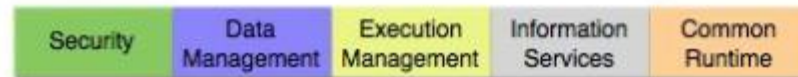
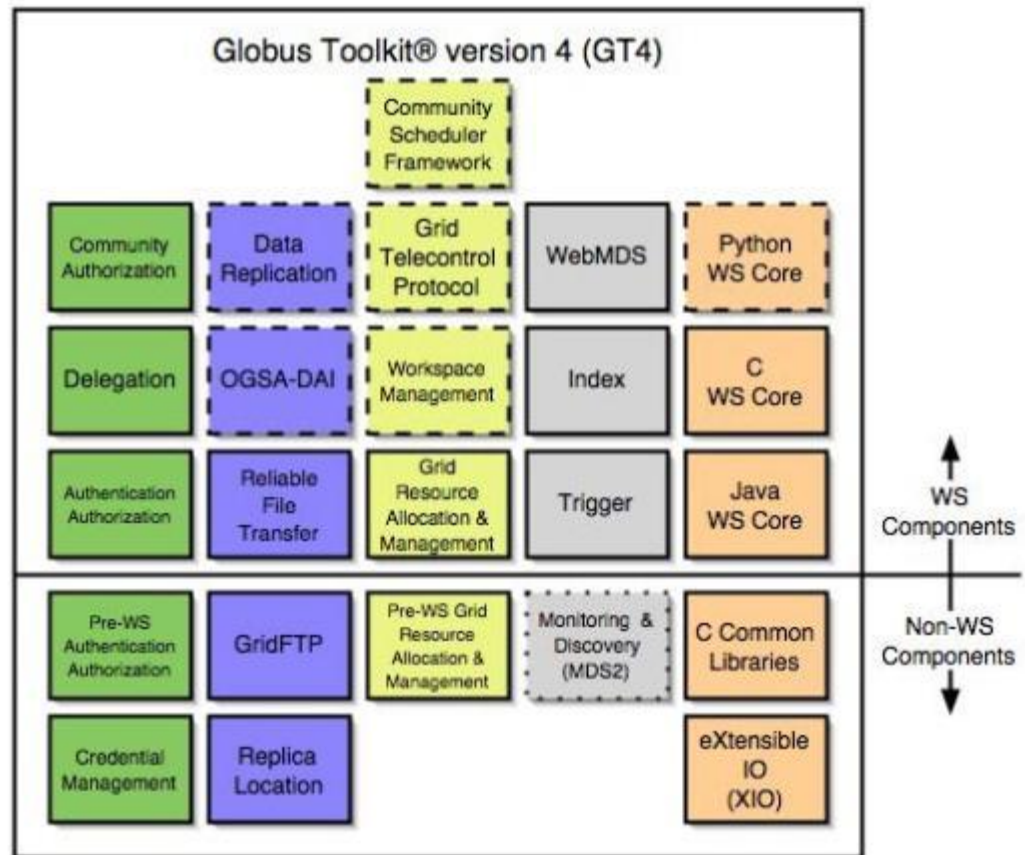
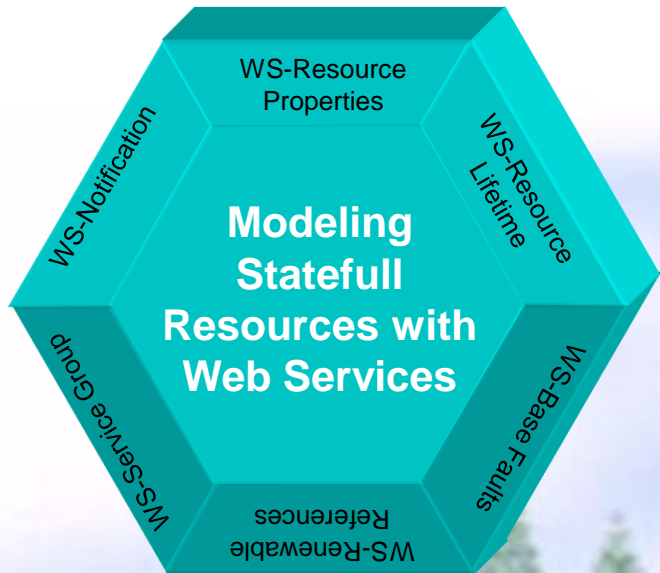
- **WS-Resource Properties**
 - Resource state and metadata “Projected” as an XML document
 - Query and Set operations
- **WS-Resource LifeTime**
 - Explicit destruction or “Soft state” time-to-live
 - Provides for cleanup of resource instances



resource

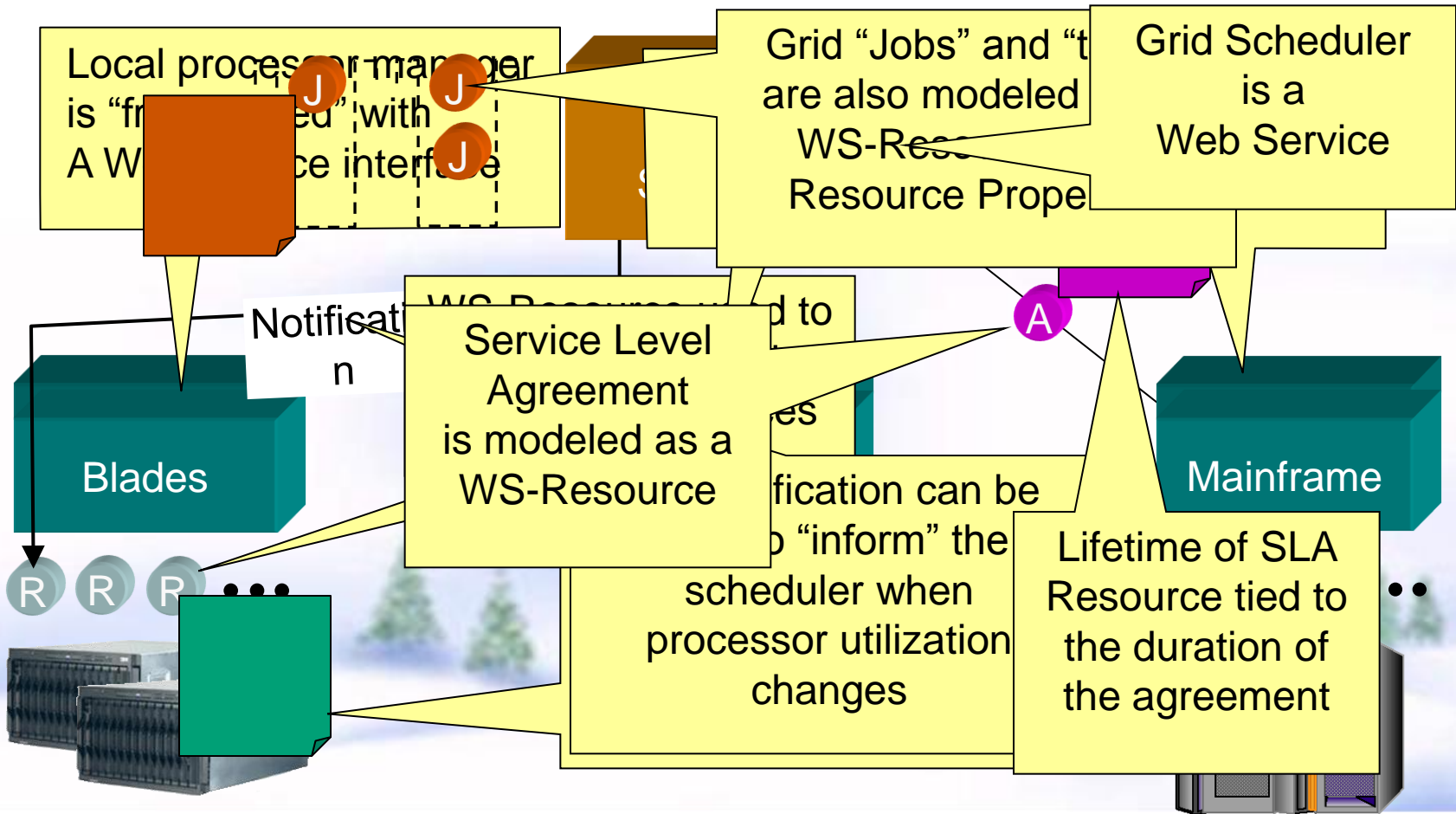
```
<ProcessorProperties>
  <ProcID>5A34C1DE03</ProcID>
  <ProcArchitecture>Power6.2</ProcArchitecture>
  <ProcSpeedMIPS>400</ProcSpeed>
  <ProcCacheMB>256<ProcCache>
  <ProcRunning>1</ProcRunning>
</ProcessorProperties>
```


OGSA – Globus

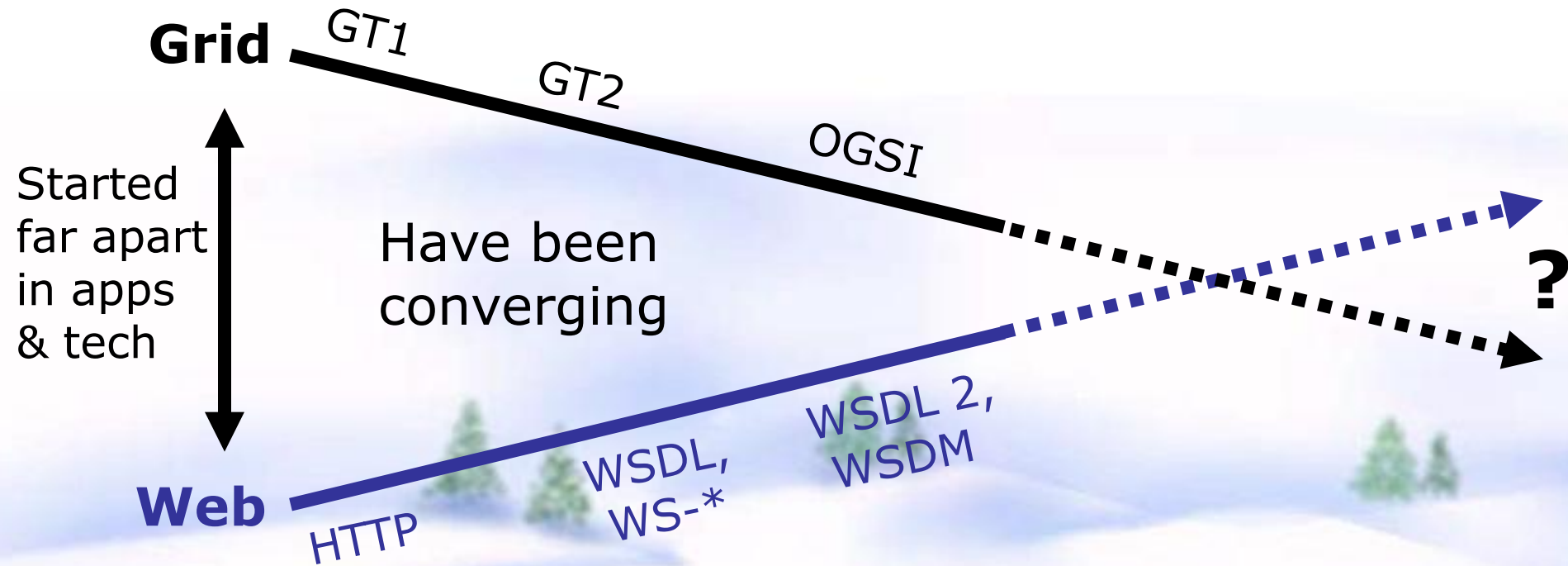


- Core GT Component: public interfaces frozen between incremental releases; best effort support
- Contribution/Tech Preview: public interfaces may change between incremental releases
- Deprecated Component: not supported; will be dropped in a future release

Exemple de WS pentru Grid



Grid si Web



Cloud computing



Amazon EC2

<http://aws.amazon.com/ec2/>

Google & IBM

AZURE

<http://www.azure.com/>





Business processes (workflows)

- Aplicații complexe: este necesară executarea mai multor programe / invocarea mai multor servicii web
- Există dependențe între aplicațiile / serviciile executate
- Workflow: succesiunea de pași executați într-o aplicație complexă
- Limbaje pentru definirea de workflow-uri: Condor, DAGMan, Karajan (Java CoG Kit), WS-BPEL



WS-BPEL

- Standard pentru specificarea de workflow-uri pentru servicii web
- Propus de IBM, Microsoft, Siebel, BEA și SAP
- Limbaj bazat pe XML, având la baza WSFL și XLANG
- Invocare de servicii: sincron / asincron
- Permite execuție secvențială și / sau paralelă a serviciilor
- Tratarea erorilor: mecanism de excepții
- Identitatea serviciilor (“partenerii”) poate fi stabilită dinamic
- Execuție ierarhică

Exemplu – WS-BPEL [MIG05]

```
<?xml version="1.0" encoding="UTF-8"?>
<process ... > <!-- import-uri de spatii de nume -->
<partnerLinks>
  <partnerLink name="printService"
partnerLinkType="print:printLink"
  partnerRole="printService"/>
</partnerLinks>
<variables>
  <variable name="hello_world"
messageType="print:PrintMessage" />
</variables>
<assign>
  <copy>
    <from><literal>Hello World</literal></from>
    <to>${hello_world.value}</to>
  </copy>
</assign>
  <invoke partnerLink="printService" operation="print"
inputVariable="hello_world" />
</process>
```



Continut

- Programarea serviciilor web – JAX-RPC
- Alte standarde pentru servicii web: UDDI, WSRF
- **J2EE - Introducere**



Enterprise = “-ilities”

- Availability
- Dependability
- Distributability
- Maintainability
- Reusability
- Reliability
- Scalability
- Recoverability
- Etc... -ilities

Availability

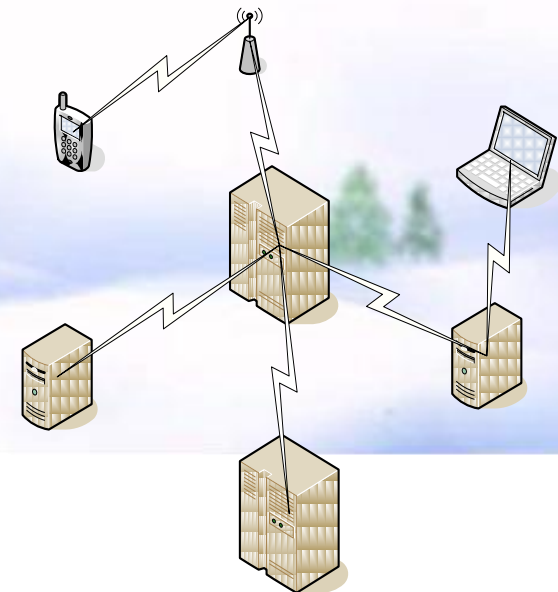
- Disponibilitatea resurselor unui sistem pentru o perioadă de timp

Dependability

- *"[...] the trustworthiness of a computing system which allows reliance to be justifiably placed on the service it delivers [...]"*
 - [IFIP WG10.4 on DEPENDABLE COMPUTING AND FAULT TOLERANCE](#)

Distributability

- Implicarea multiplelor resurse / medii
- Resure distribuite geografic
- Independenta si interoperabilitate





Maintainability

- “You are developing tomorrow’s legacy code”
- Menținerea și păstrarea stării sistemelor
- Posibilitatea de recuperare din eroare.
 - *What makes a system maintainable?*

Reusability

- Pentru a utiliza din nou, după salvare sau după prelucrare.
- Reducerea duplicării codului / “Copy & Paste”
- Ușor de menținut
- Rapid de dezvoltat

Reliability

- Încrederea în sistem că va oferi răspunsurile așteptate
- Disponibilitate 24-7-365
- Oferă suport pentru erorile frecvente



Scalability

- “Refers to how much a system can be expanded. The term by itself implies a positive capability. For example, "the device is known for its scalability" means that it can be made to serve a larger number of users without breaking down or requiring major changes in procedure.”
 - Computer Desktop Encyclopedia

Recoverability

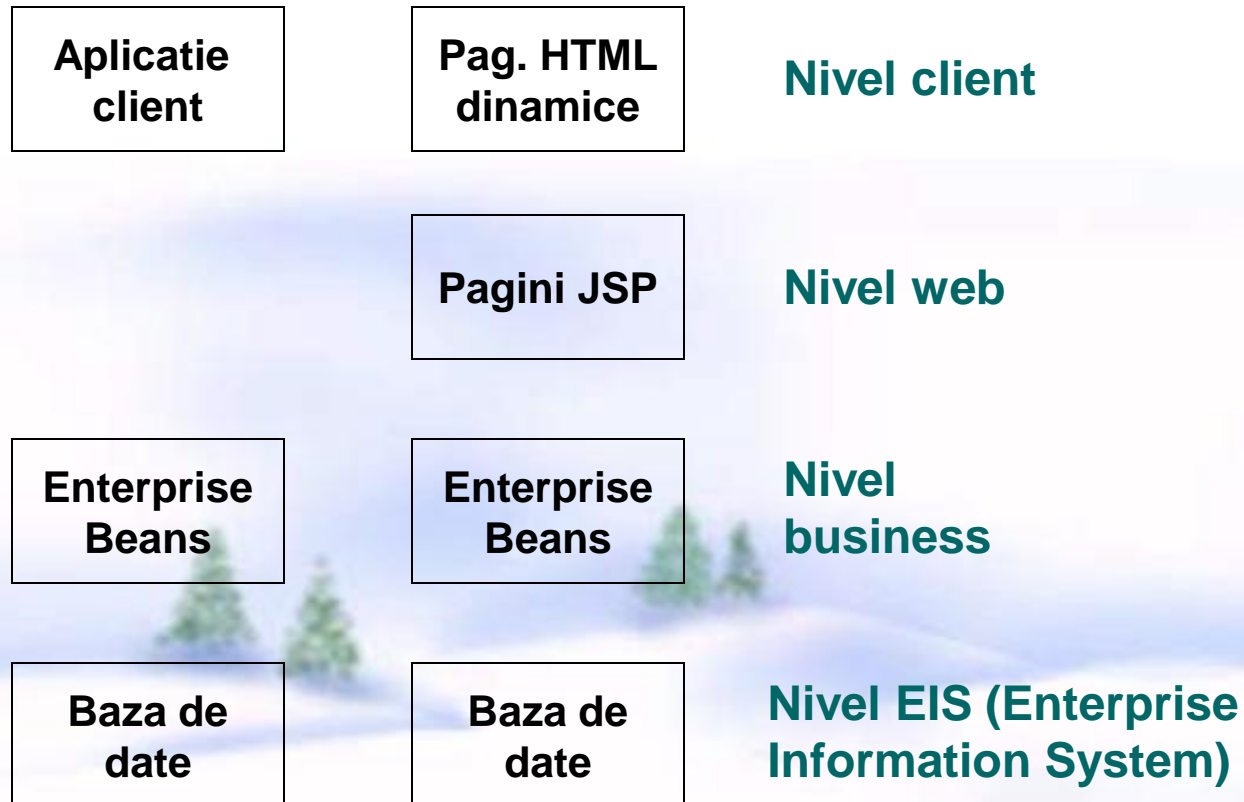
- Punct unic de eșec
- Clustere de servere
- Planuri urgente de recuperare din eroare
- Management al dezastrelor



J2EE (Java 2 Enterprise Edition)

- Specificatii pentru dezvoltarea aplicatiilor de tip *enterprise*, aparute in 1999
- Scop: construirea unui standard pentru dezvoltarea de servicii middleware
- Bazata pe componente
- Arhitectura pe mai multe nivele
- Flexibilitate, posibilitatea de utilizare a componentelor

Aplicatii pe mai multe nivele [PAW01]





Componente J2EE

- Componente client: aplicații (de obicei cu interfața grafică), applet-uri
 - “thin clients”
- Componente web:
 - **Servleti** – clase Java care procesează cereri în mod dinamic
 - **JSP** (Java Server Pages) – pagini HTML cu porțiuni de cod Java
- Componente business: **EJB (Enterprise Java Beans)** – procesări specifice domeniului
 - Session beans
 - Entity beans
 - Message-driven beans



Servicii J2EE

- Model pentru securitate
- Model pentru tranzactii
- Servicii de nume (JNDI)
- Model de conectivitate la distanta



Tipuri de containere J2EE

- Container EJB – gestionează componentele EJB ale unei aplicații; rulează pe un server J2EE
- Container web – gestionează componentele web (servlet, JSP); rulează pe un server J2EE
- Container de aplicații client – rulează pe mașina clientului
- Container de applet-uri – browser web

Impachetarea componentelor

- Fișiere .ear – *Enterprise ARchive*
- *Deployment descriptor*
 - Fișier XML ce specifică setările pentru aplicație
 - Poate fi modificat independent de codul sursă
- Componentele JAR / WAR (Web Archive) se adaugă la EAR din linia de comandă sau cu ajutorul unei interfețe grafice



Referinte

- Specificatiile JAX-RPC, disponibile la <http://java.sun.com/xml/downloads/jaxrpc.html>
- [UDDI02] http://searchwebservices.techtarget.com/tip/1,289483,sid26_gci809279_00.html – An inside look at UDDI
- www.globus.org/wsrf/specs/ws-wsrf.pdf - The WS-Resource Framework
- [MIG05] http://www.eclipse.org/tptp/platform/documents/design/choreography_html/tutorials/wsbpel_tut.html - WS-BPEL tutorial, de A. Miguel
- [PAW01] <http://java.sun.com/developer/technicalArticles/J2EE/Intro/> - J2EE Technology Center, de M. Pawlan



Referinte suplimentare

- [DEI02]
www.deitel.com/newsletter/20020523/articles/JavaWebServices_ **JAX-RPC**.pdf - JAX-RPC Overview, fragment din cartea “Java Web Services for Experienced Programmers”, Deitel Developer Series, 2002
- [Han02] <http://javaboutique.internet.com/tutorials/Axis/> - Web Services with Axis, de K. Hansen
- <http://www-128.ibm.com/developerworks/webservices/library/ws-uddiv3/> - Rocket Ahead with UDDI v3
- <http://www.informit.com/articles/article.asp?p=31833&rl=1> – An introduction to J2EE
- <http://www.ibm.com/developerworks/java/library/ws-java1.html>



Sumar

- Programarea serviciilor web
 - JAX-RPC
 - Programarea clientilor
- UDDI
- WSRF
- WS-BPEL
- J2EE
 - Introducere
 - Componente J2EE
 - Containere J2EE