Griglie e Sistemi di Elaborazione Ubiqui

Corso di Laurea Specialistica
in Ingegneria informatica

Lucidi delle Esercitazioni

Anno Accademico 2005/2006

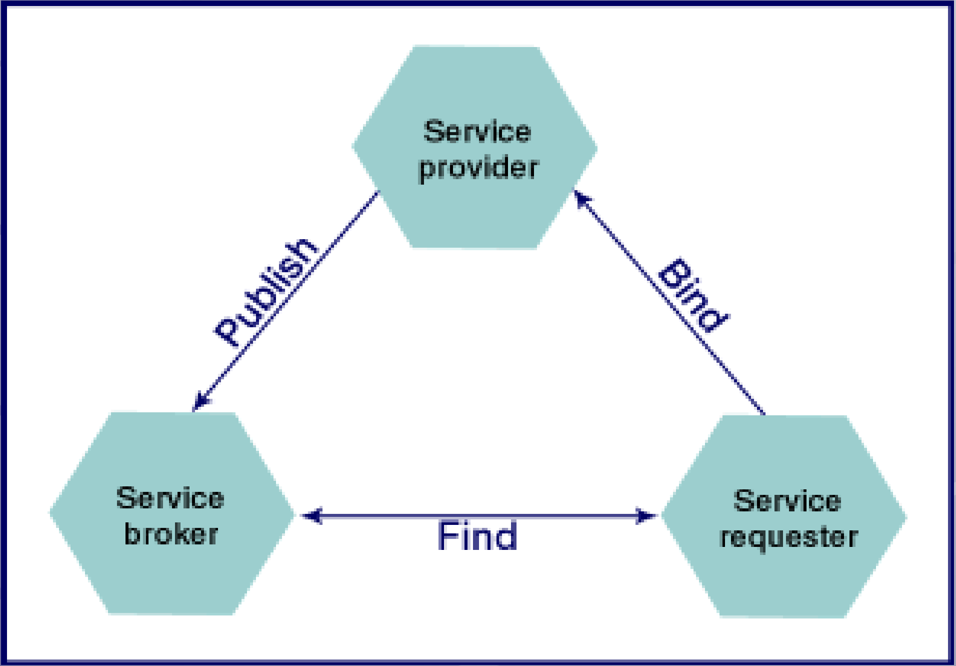
Ing. Antonio Congiusta

Summary

- ✓ Web Services introduction
- ✓ Technical aspects
- ✓ WSRF and GT4
- ✓ The GT4 container
- ✓ GT4 Core installation and testing

Web Services

- ✓ A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL).



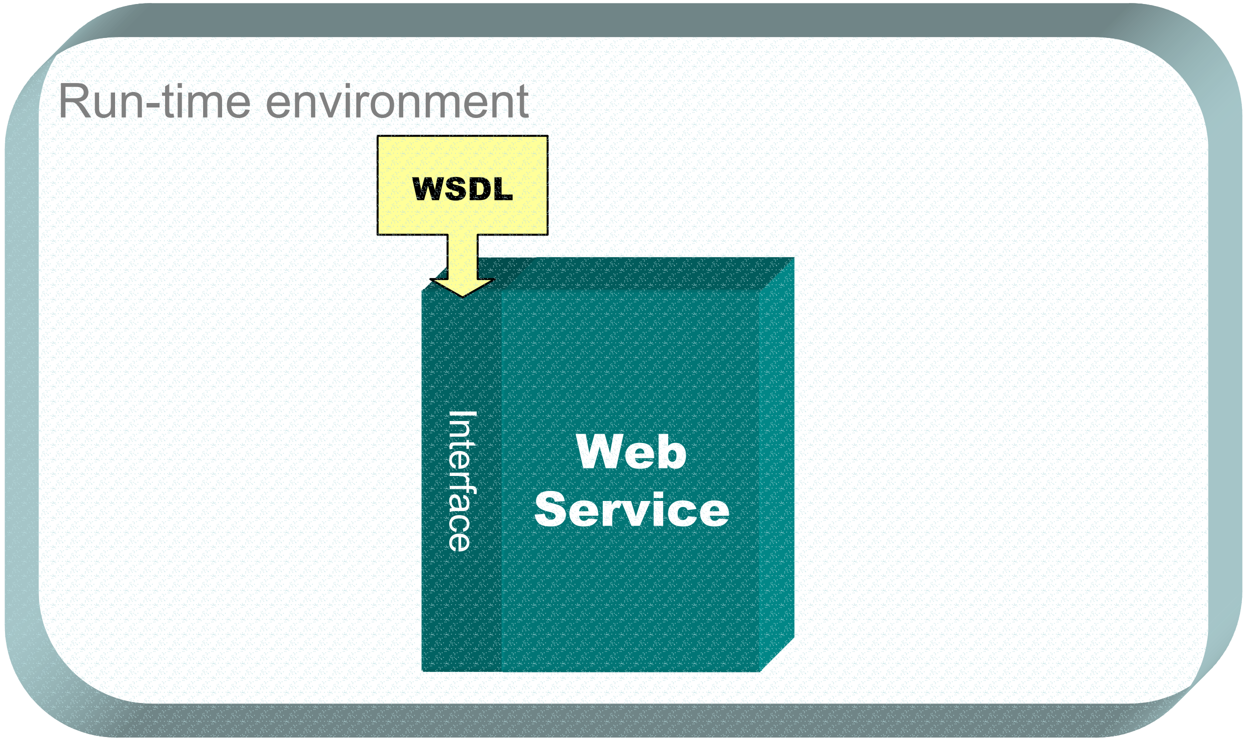
- ✓ Systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with XML messages in conjunction with other Web-related standards.

Primary Web Services Technologies

- ✓ Simple Object Access Protocol (SOAP)
  - ▪ Structure for transporting XML documents
  - ▪ Over SMTP, HTTP, FTP, RPC
- ✓ Web Service Description Language (WSDL)
  - ▪ XML technology -- describes interface of a WS
  - ▪ Standardizes input/output representation
- ✓ Universal Description, Discovery, and Integration Language (UDDI)
  - ▪ Registry for web services

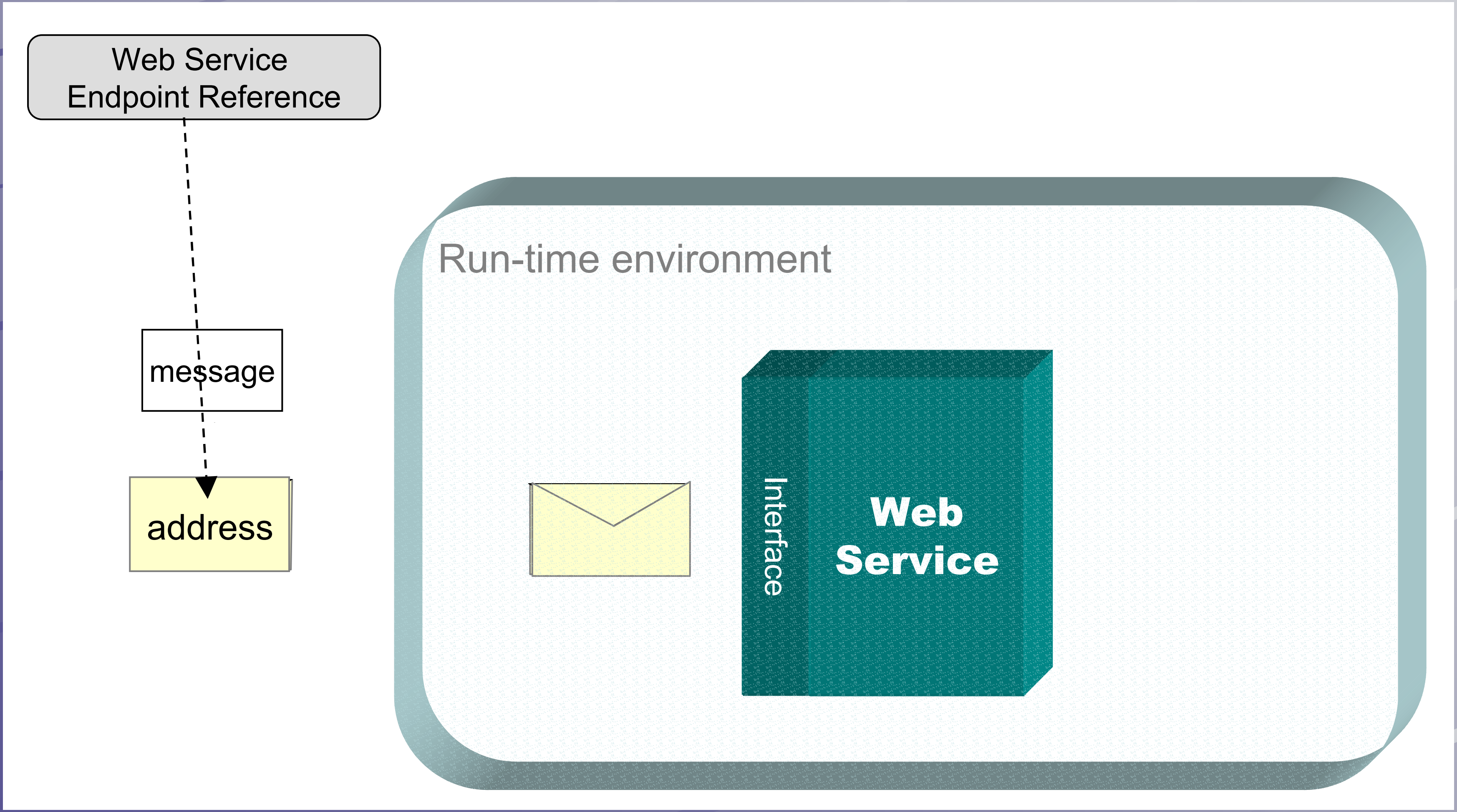
General Web Service invocation model

Web Service



General Web Service invocation model

Web Service



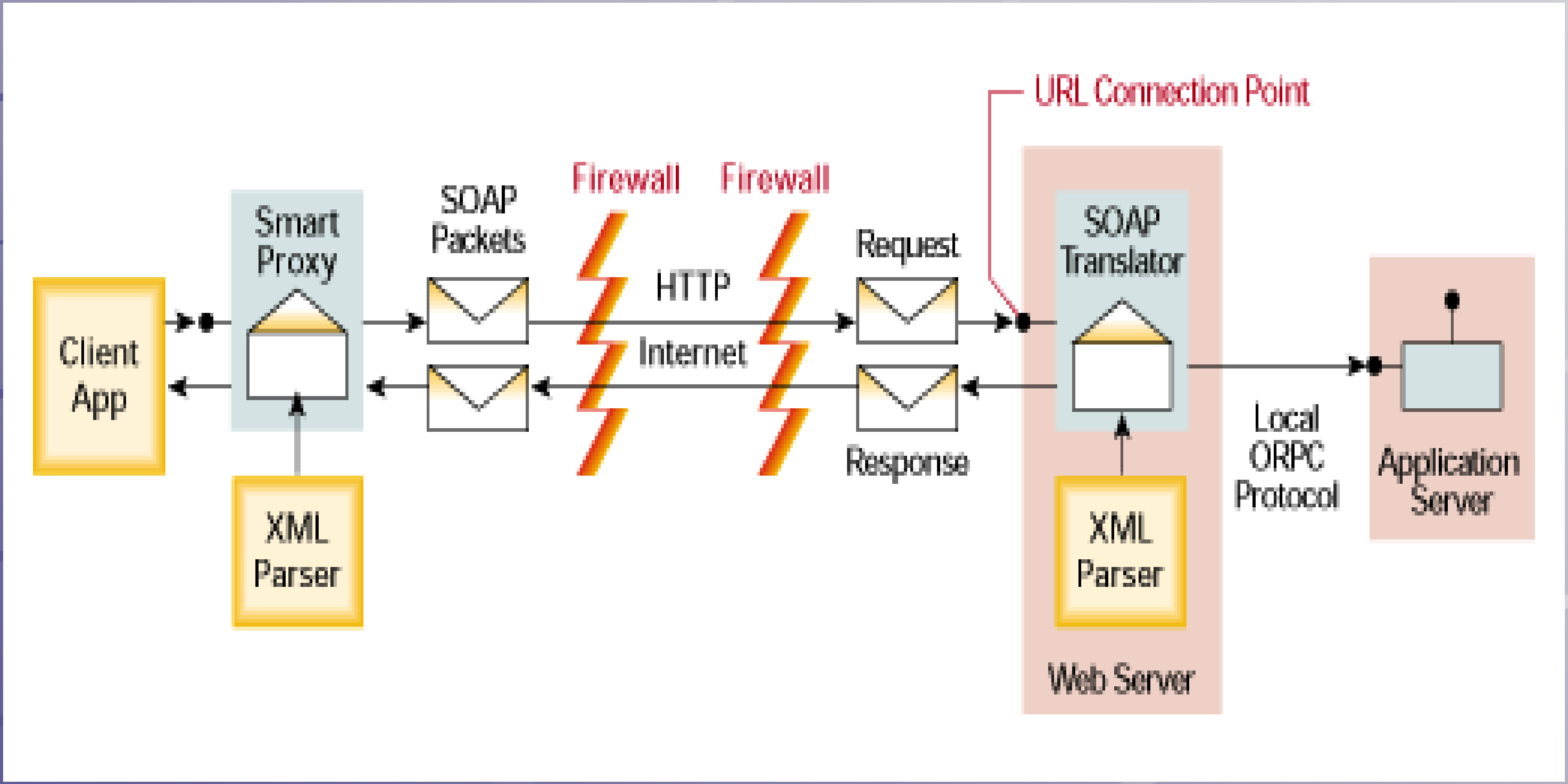
Simple Object Access Protocol (SOAP)

- ✓ A SOAP message is fundamentally a one-way transmission between SOAP nodes, from a SOAP sender to a SOAP receiver
- ✓ SOAP messages are expected to be combined by applications to implement more complex interaction patterns:
  - request/response
  - multiple, back-and-forth "conversational" exchanges
  - etc.

SOAP - Acronym for

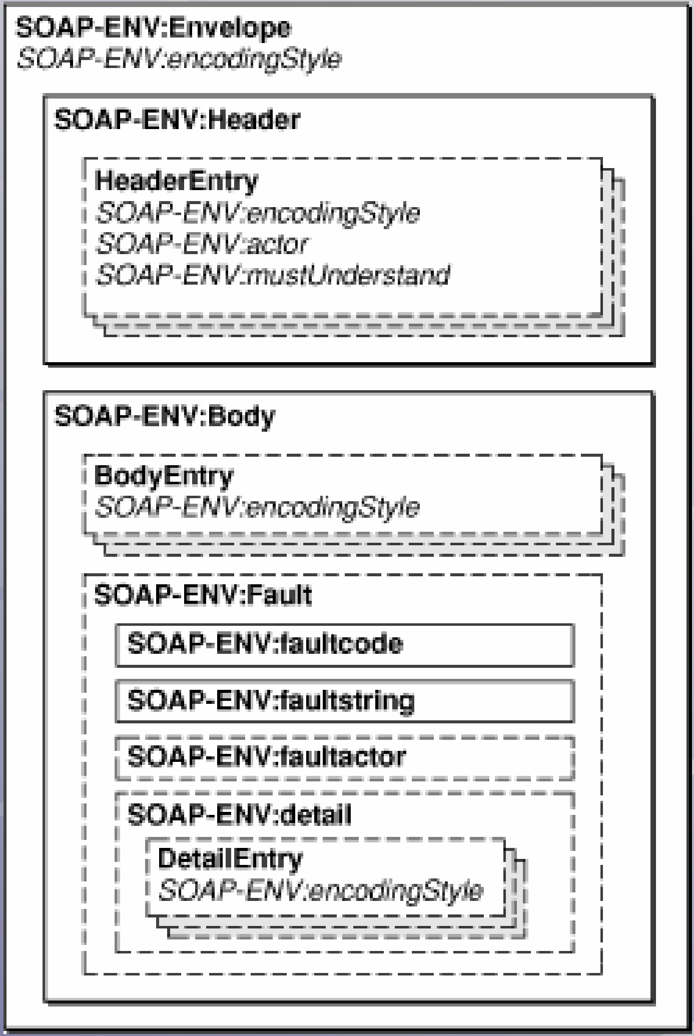
- ✓ Simple: Transporting XML structured messages across internet using HTTP
- ✓ Object: transportation of COM objects
  - Common Object Model (COM): open software architecture from DEC, Microsoft, allowing interoperation between ObjectBroker and OLE
  - Microsoft evolved COM into DCOM.
- ✓ Access - a philosophy: services will be easier to deploy when binding them to common protocols (HTTP)
  - Most firewalls already pass through web page data
- ✓ Protocol: SOAP is an XML based protocol used to exchange distributed data over HTTP
  - Origins in RPC

SOAP Architecture



SOAP Components and Elements

- ✓ Components:
  - Formatting conventions
  - Transport/protocol binding
  - Encoding rules
  - RPC mechanism
- ✓ Elements:
  - **Envelope**: Required
  - **Header** [Optional] - use:
    - ✓ Transaction data
  - **Body**: Required, use:
    - ✓ Method call and its parameters



SOAP Syntax Rules

- ✓ A SOAP message MUST be encoded using XML
- ✓ A SOAP message MUST use the SOAP Envelope namespace
- ✓ A SOAP message MUST use the SOAP Encoding namespace
- ✓ A SOAP message must NOT contain a DTD reference
- ✓ A SOAP message must NOT contain XML Processing Instruction

SOAP Request/Response Example

```
POST /InStock HTTP/1.1
Host: www.stock.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-
    envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soa
    p-encoding">

  <soap:Body
    xmlns:m="http://www.stock.org/stock">
   <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
   </m:GetStockPrice>
  </soap:Body>

</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: application/soap; charset=utf-8
Content-Length: nnn

<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-
    envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soa
    p-encoding">

  <soap:Body
    xmlns:m="http://www.stock.org/stock">
   <m:GetStockPriceResponse>
    <m:Price>34.5</m:Price>
   </m:GetStockPriceResponse>
  </soap:Body>

</soap:Envelope>
```

Web Services Description Language (WSDL)

- ✓ You now know how to make and XML doc
- ✓ You now know how messages are exchanged with XML between a client and a server.
  - Client Request: select a method, submit data
  - Server Response: return XML data
- ✓ WSDL is an XML document that describes a Web service.
  - It specifies the location of the service and the operations (or methods) the service exposes.

Web Services Description Language (WSDL)

- ✓ WSDL is written in XML
  - WSDL is an XML document
- ✓ WSDL is used to :
  - locate Web services
  - describe Web services
- ✓ WSDL is used by WSRF
  - not yet a W3C standard: "a *suggestion* for describing services for the W3C XML Activity on XML Protocols"
    - ✓ http://www.w3.org/TR/wsdl

WSDL Document Structure

<definitions>: Root WSDL Element

<types>: data types to be transmitted

<messages>: message to be transmitted

<portType>: operations (functions) supported

<bynding>: message transmission protocol

<service>: service location

WSDL example

```
<?xml version="1.0" encoding="UTF-8"?>
<definitions name="HelloService"
   targetNamespace="http://www.ecerami.com/wsdl/HelloService.wsdl"
   xmlns="http://schemas.xmlsoap.org/wsdl/"
   …
   <message name="SayHelloRequest">
      <part name="firstName" type="xsd:string"/>
   </message>
   …
   <portType name="Hello_PortType">
      <operation name="sayHello">
         <input message="tns:SayHelloRequest"/>
         <output message="tns:SayHelloResponse"/>
      </operation>
   </portType>
   <binding name="Hello_Binding" type="tns:Hello_PortType">
      <soap:binding style="rpc" transport="http://schemas.xmlsoap.org/soap/http"/>
      <operation name="sayHello">
         <soap:operation soapAction="sayHello"/>
         <input>
            <soap:body
               encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
               namespace="urn:examples:helloservice"
               use="encoded"/>
         </input>
         <output>
            …
         </output>
      </operation>
   </binding>
   …
</definitions>
```

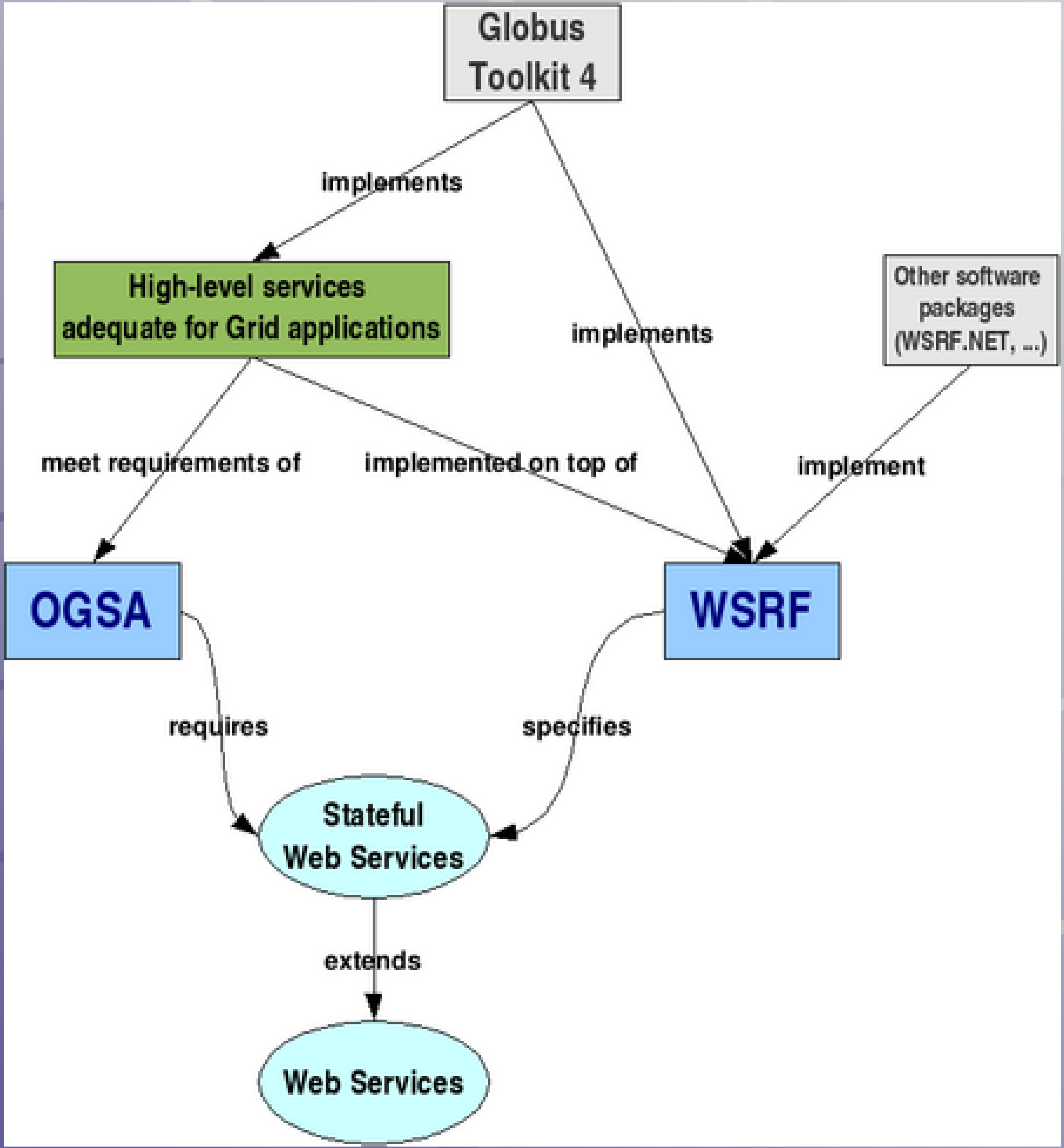
Useful Links

- ✓ Good Books:
  - D. Chappel, T. Jewell, “Java Web Services,” Orielly, 2002
  - E. Cerami, “Web Services Essentials,” Orielly, 2002
  - Oellermann, “Architecting Web Services,” AI Press, 2001
- ✓ ANT: Manual
  - http://ant.apache.org/manual/index.html
- ✓ Web Services Arch:
  - http://www.w3.org/TR/2004/NOTE-ws-arch-20040211/
- ✓ XML 1.0 Standard:
  - http://www.w3.org/TR/2000/REC-xml-20001006
- ✓ XML Schema:
  - http://www.w3.org/TR/xmlschema-0/

OGSA Defines Basic Capabilities

- ✓ **Infrastructure Services**
- ✓ **Execution Management Services**
- ✓ **Data Services**
- ✓ **Resource Management Services**
- ✓ **Security Services**
- ✓ **Self-Management Services**
- ✓ **Information Services**
- ✓ **Security Considerations**

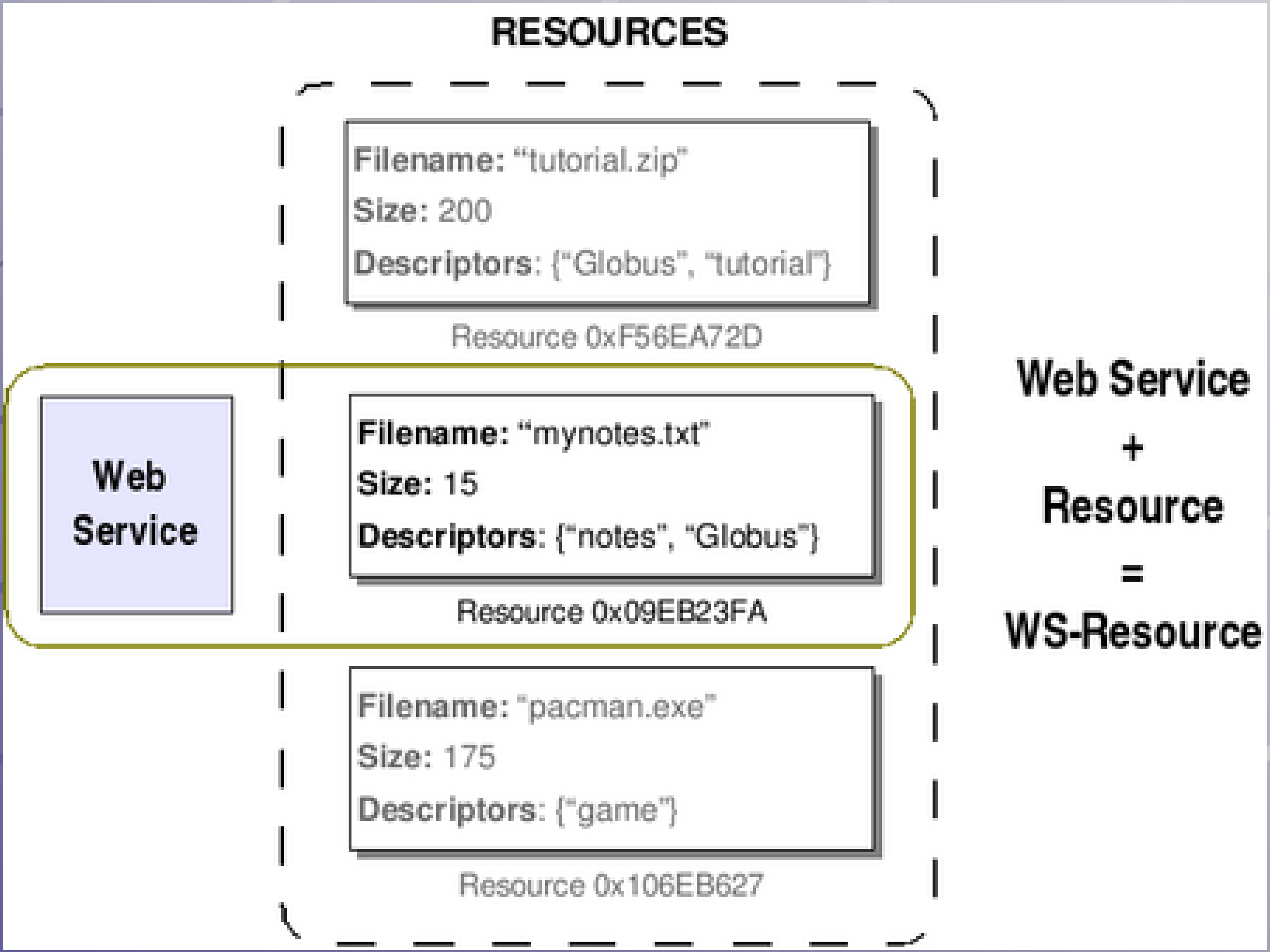
OGSA, WSRF, and GT4



WS-Resource: Stateful Resources

- ✓ Ws-Resource = Web Service + WSRF
- ✓ A *stateful* resource is something that exists even when you're not interacting with it.
  - E.g. database backend service
- ✓ Stateful resources have *properties* that define state
  - these properties are how you interact with them
  - Properties have values
  - Add/remove/change properties and values dynamically
- ✓ WSRF Specification:
  - a WS-Resource is the combination of a Web service and a stateful resource on which it acts.

WS-Resource: Stateful Resources



WSRF Specifications

- ✓ List is still changing, but basically includes..
- ✓ Core:
  - WS-Resource Framework (WSRF)
  - WS-ResourceProperties (WSRF-RP)
  - WS-ResourceLifetime (WSRF-RL)
  - WS-ServiceGroup (WSRF-SG)
  - WS-Base Faults(WSRF-BF)
- ✓ Related:
  - WS-Notifications
  - WS-Addressing

WS-Addressing

- ✓ Web Services have always had addressing:
  - URIs (Uniform Resource Identifiers)
  - Looks like URLs:
    - ✓ http://webservices.mysite.com/weather/us/WeatherService
- ✓ For a Web Service URI:
  - Typically pass URI to a program
  - If you typed a Web Service URI into your web browser, you would probably get an error message or some unintelligible code
    - ✓ Some services include a polite response page

WS-Resource Factory

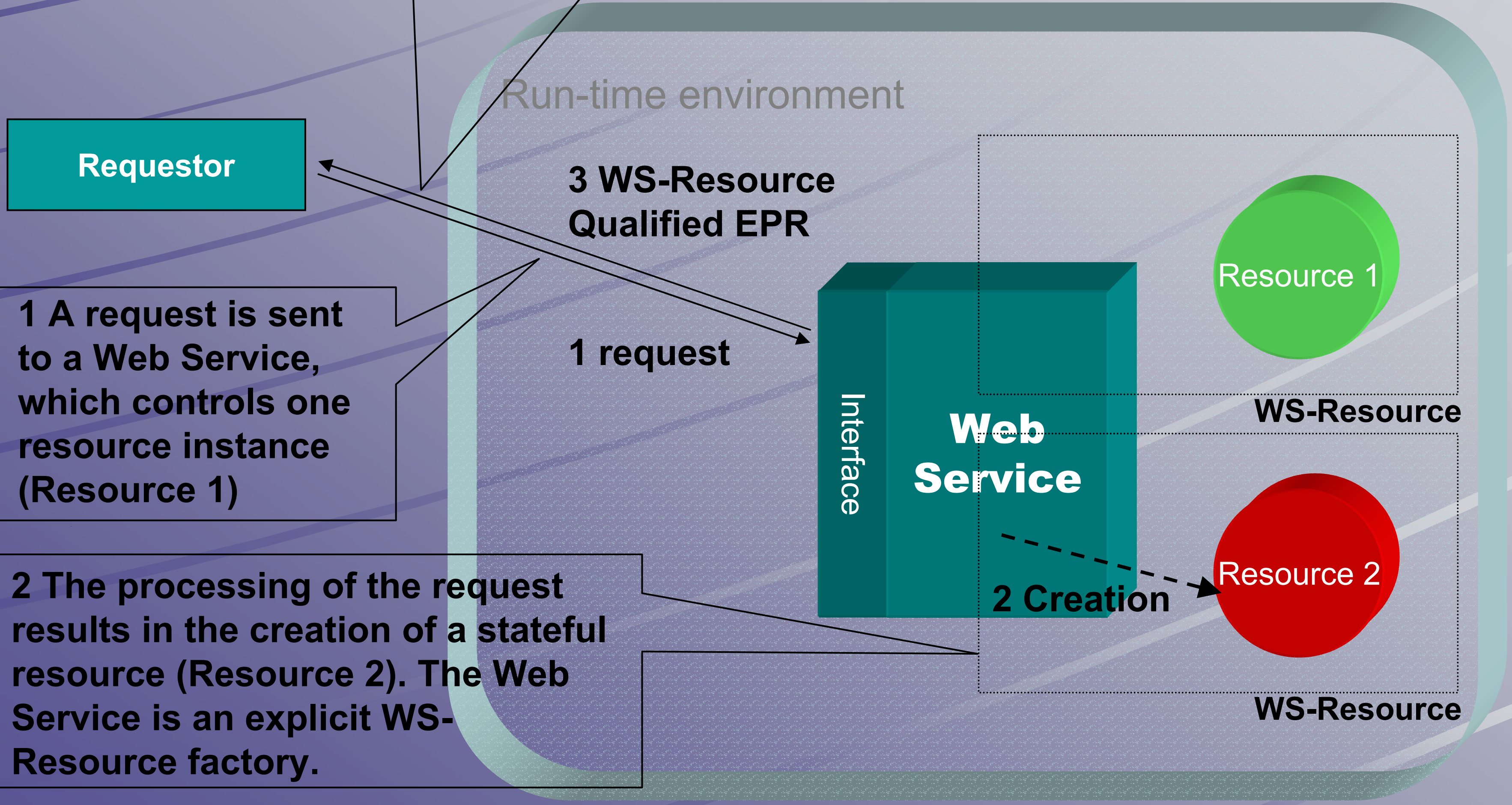
- ✓ *Definition*: any Web service capable of bringing a WS-Resource into existence and assigning the new WS-Resource an identity.
- ✓ *Phases* of the creation process:
  1. a new stateful resource instance is created;
  2. the created instance is assigned an identity;
  3. the new stateful resource is assigned to a Web service.
- ✓ The response message of a WS-Resource factory operation must include a WS-Resource-qualified endpoint reference containing a WS-Resource context that refers to the new WS-Resource
- ✓ The WS-Resource-qualified endpoint reference can be implicitly returned by placing it into a registry for later retrieval.

WS-Resource: explicit WS-Resource factory

3 The endpoint reference of WS-Resource instance n.2 is returned. Endpoint reference = wsa:Address + wsa:ReferenceProperties

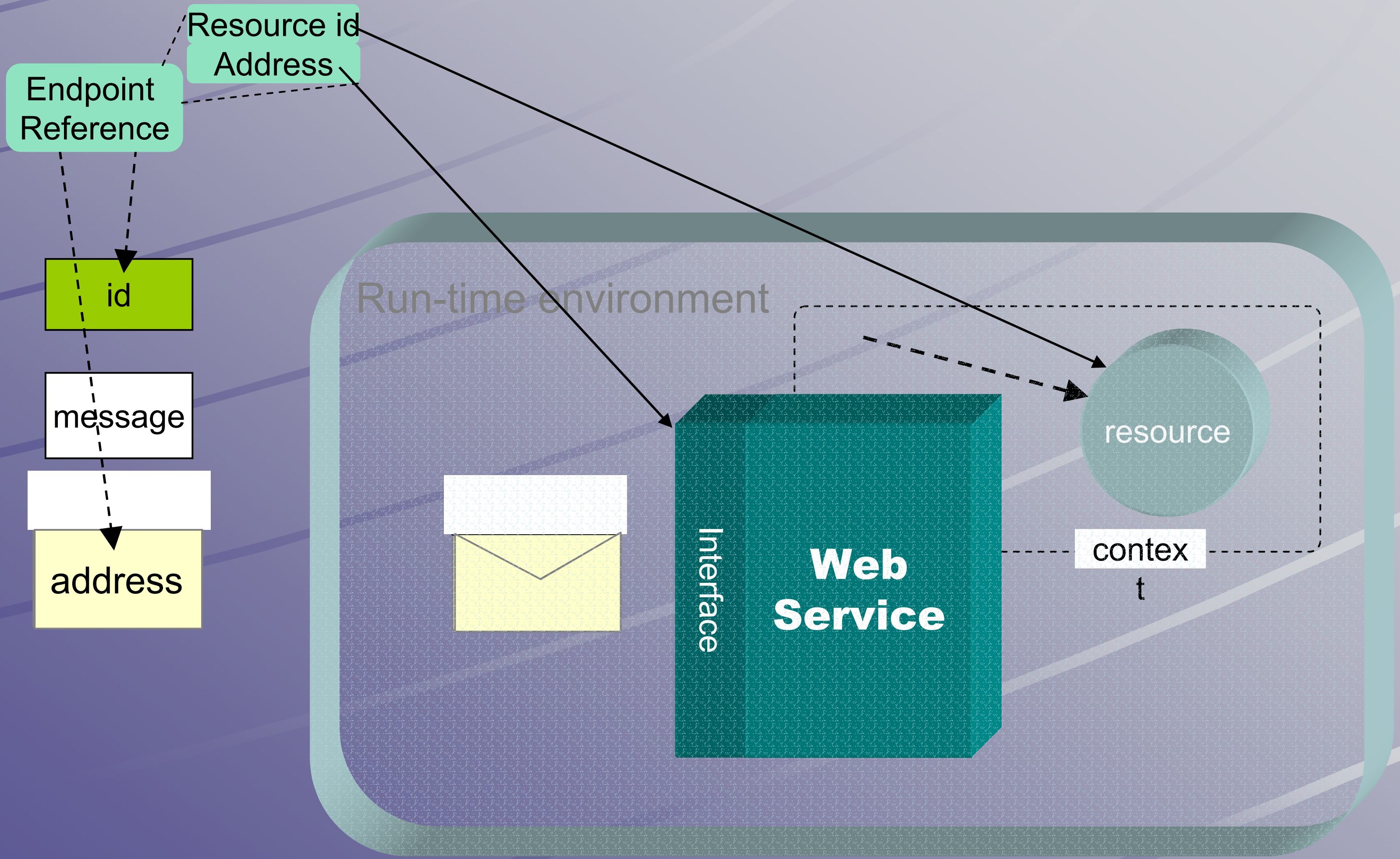
1 A request is sent to a Web Service, which controls one resource instance (Resource 1)

2 The processing of the request results in the creation of a stateful resource (Resource 2). The Web Service is an explicit WS-Resource factory.



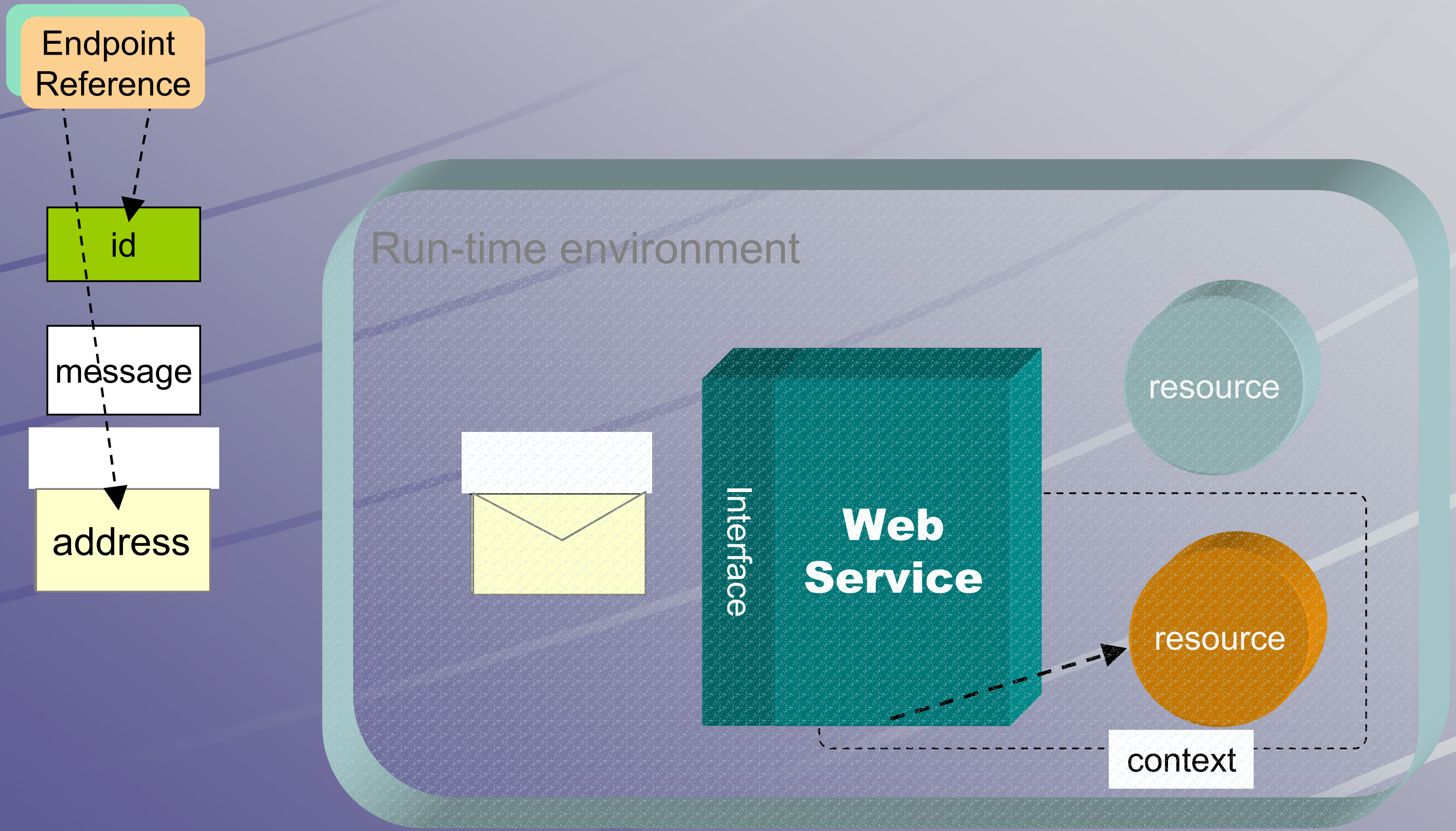
The WS-Resource framework model

Using a Web service to access a WS-Resource



The WS-Resource framework model

Using a Web service to access another WS-Resource



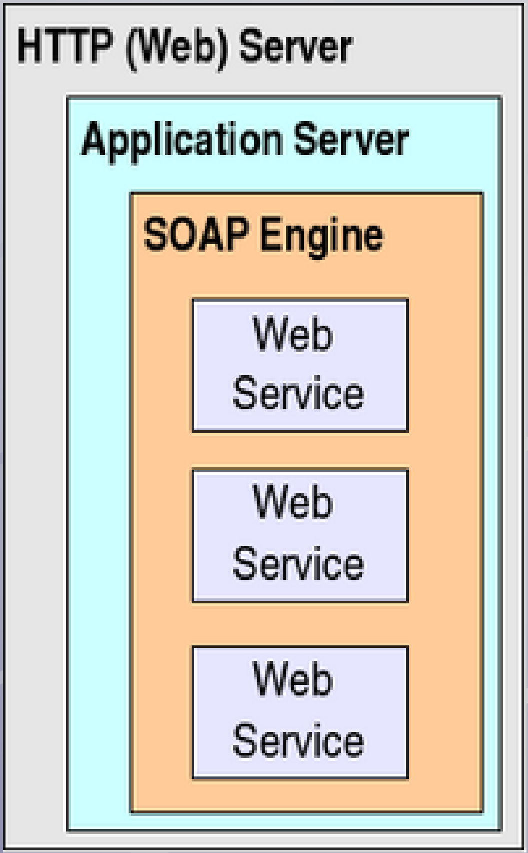
WSRF and Globus-specific features of WSDL

- ✓ **Resource properties**:
  - *wsrp:ResourceProperties* attribute of the *portType* element -- specify service resource properties are. Remember that the resource properties are were we'll keep all our state information.
- ✓ **WSDL Preprocessor**
  - *wsdlpp:extends* attribute of the *portType* element allows existing WSRF portTypes to be included in our portType without having to copy-and-paste from official WSRF WSDL files.
  - use the value of that attribute to generate correct WSDL which includes our own portType definitions plus any WSRF portType we might need in our service.
  - Globus-specific feature that is included to make life easier for programmers.
- ✓ **createResource operation**:
  - analogous to object creation returning "instance name" or an "instance reference".
  - extra operation besides service specific ones.

createResource operation returns an endpoint reference (EPR)

WS Software stack used by GT4 WSRF

- ✓ HTTP Server
  - Apache HTTP Server
- ✓ Application Server
  - Apache Tomcat
- ✓ SOAP Engine
  - Apache AXIS
  - Supports *wsdl2java* tool - build Java proxies and skeletons from WSDL docs.
- ✓ Web Service
  - User App



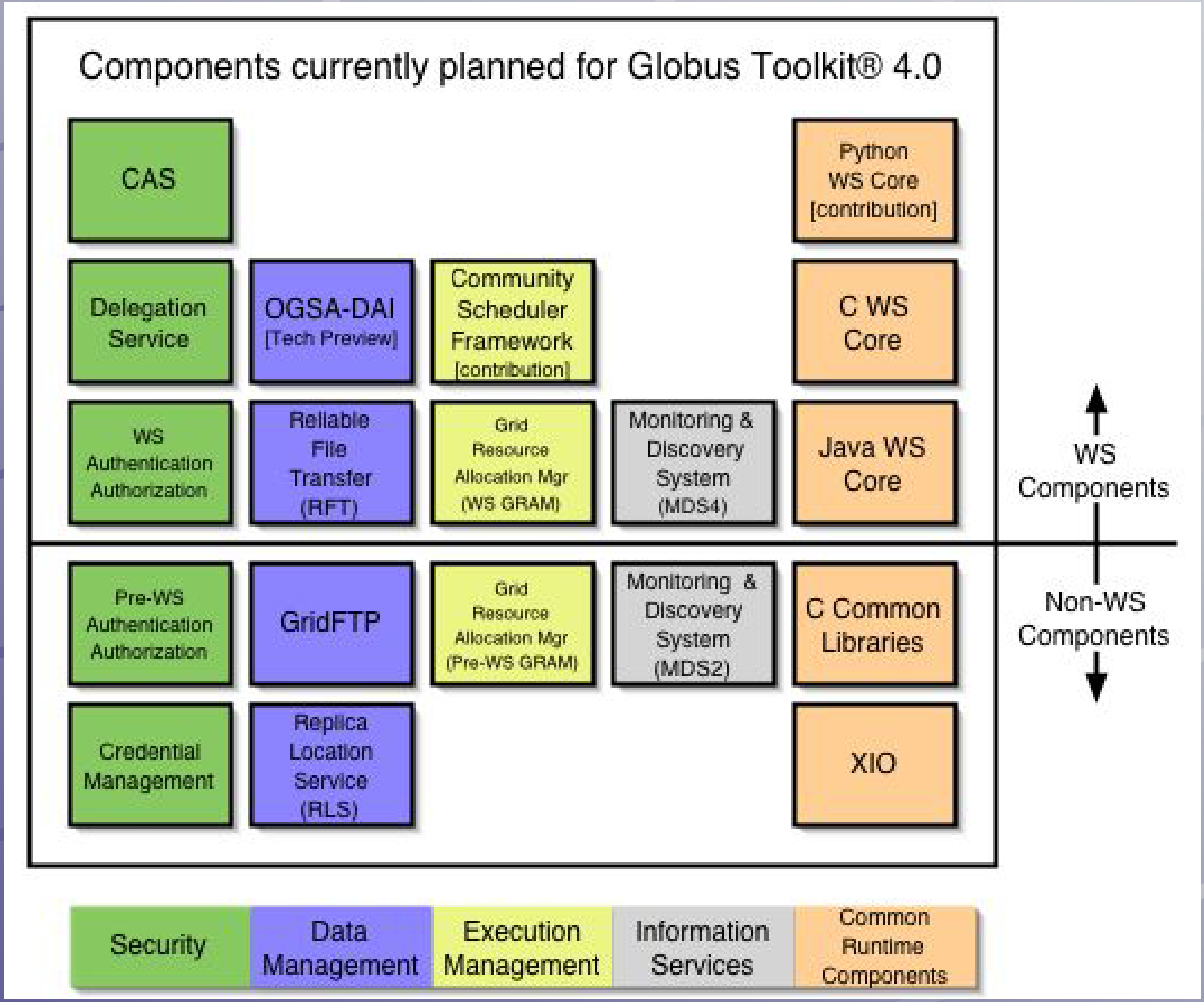
Definition of a GT4 Container

- ✓ **GT4 containers:** term that denotes Web service containers with a set of common features:
  - implements SOAP over HTTP as a message transport protocol and transport-level and WS-Security message-level security for all communications;
  - implements WS-Addressing, WSRF, and WS-Notification functionality
  - supports logging via Log4j, which implements the Jakarta Commons Logging API
  - defines WSRF WS-Resources with properties providing access to information about services deployed in the container and container properties such as version and start time.

GT4 Java Containers

- ✓ GT4 Java WS Core code
  - implements WSRF and WS-Notification as well as supporting code for security and management.
  - code designed to be used with Apache Axis as a SOAP engine plus other relevant Apache components such as the WS-Addressing and WS-Security
- ✓ To produce a complete GT4 Java container, you can host GT4 Java WS Core + Axis combination either as a:
  - "simple Java container" (easier installation and administration -- recommended unless already running Tomcat)
  - Tomcat: more featureful but complex servlet container
- ✓ This containter can also host other GT4 services:
  - GRAM, RFT, MDS-Index, MDS-Trigger, and MDS-Archive.

GT4 Roadmap


Components currently planned for Globus Toolkit® 4.0
CAS
Python WS Core [contribution]
Delegation Service
OGSA-DAI [Tech Preview]
Community Scheduler Framework [contribution]
C WS Core
WS Authentication Authorization
Reliable File Transfer (RFT)
Grid Resource Allocation Mgr (WS GRAM)
Monitoring & Discovery System (MDS4)
Java WS Core
WS Components
Non-WS Components
Pre-WS Authentication Authorization
GridFTP
Grid Resource Allocation Mgr (Pre-WS GRAM)
Monitoring & Discovery System (MDS2)
C Common Libraries
Credential Management
Replica Location Service (RLS)
XIO
Security
Data Management
Execution Management
Information Services
Common Runtime Components

GT WSRF core

- ✓ Container
  - Hosts services
  - Built on top of Apache Axis
- ✓ Clients
  - Interact with services
- ✓ Build tools
  - For writing new services
  - Based around Apache Ant

GT4 WSRF Core Istallation

- J2SE 1.4.2+ SDK from Sun, IBM, HP, or BEA.
- Ant 1.5.1+ (1.6.1+ if using Java 1.5). (Apache web site)
- ws-core-4.0.1-bin (from Globus site)
- globus build tool (optional) → requires Python under Windows

Make sure you have JAVA_HOME set to the directory where JAVA is installed on your machine.

Set ANT_HOME to the directory where Ant is installed

Set GLOBUS_LOCATION environment variable to point to the GT4 home directory (i.e. c:\ws-core-4.0.0).

Add ANT_HOME\bin to the PATH environment variable.

Question Time

? ?

?

?

?

?

The more you ask…

…the less I question you!