egee

Enabling Grids for E-sciencE

Web Services, WSRF and GT4

Mike Mineter

National e-Science Centre, Edinburgh
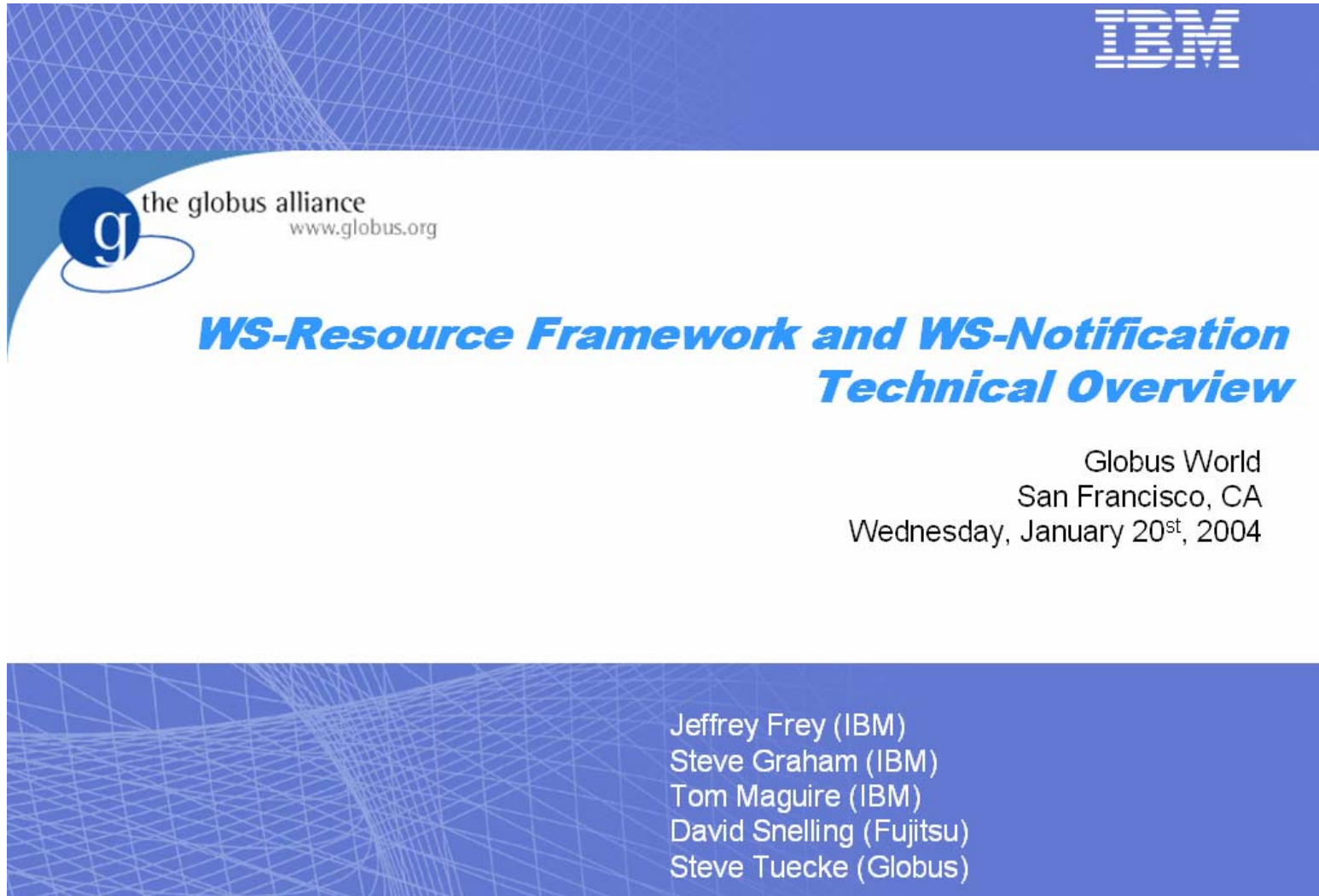
Information Society

http://www.nesc.ac.uk/action/esi/contribution.cfm?Title=385

IBM

the globus alliance
www.globus.org

WS-Resource Framework and WS-Notification Technical Overview

Globus World
San Francisco, CA
Wednesday, January 20th, 2004

Jeffrey Frey (IBM)
Steve Graham (IBM)
Tom Maguire (IBM)
David Snelling (Fujitsu)
Steve Tuecke (Globus)

- **This presentation can be re-used for academic purposes.**

- **However if you do so then please let training-support@nesc.ac.uk know. We need to gather statistics of re-use: no. of events, number of people trained. Thank you!!**

Enabling Grids for E-sciencE

- **An orientation to Web Services and to their role in Grid computing**

- **No prior knowledge assumed**

Enabling Grids for E-sciencE

- " Web Services are _the_ way to build Grids"

- Web Services

- Relevance of Web Services to Grids

- Extending WS for grids

- So where are we now ?

- Where might we be going?!

eGee

Enabling Grids for E-sciencE

Web Services

Grid Technology

Grid Services

- **Commerce**
- **Standards**
- **Tools**

- **Research driven**
- **Data-intensive**
- **Compute intensive**
- **Collaboration – sharing of resources**
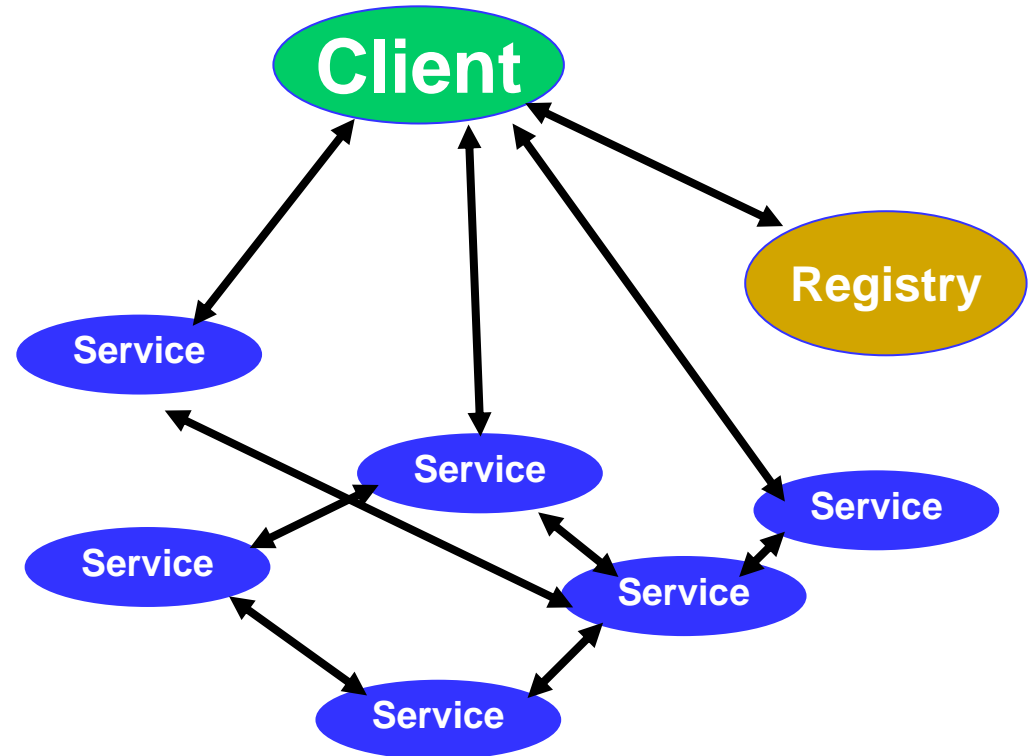   - **- Trust: opening resources**

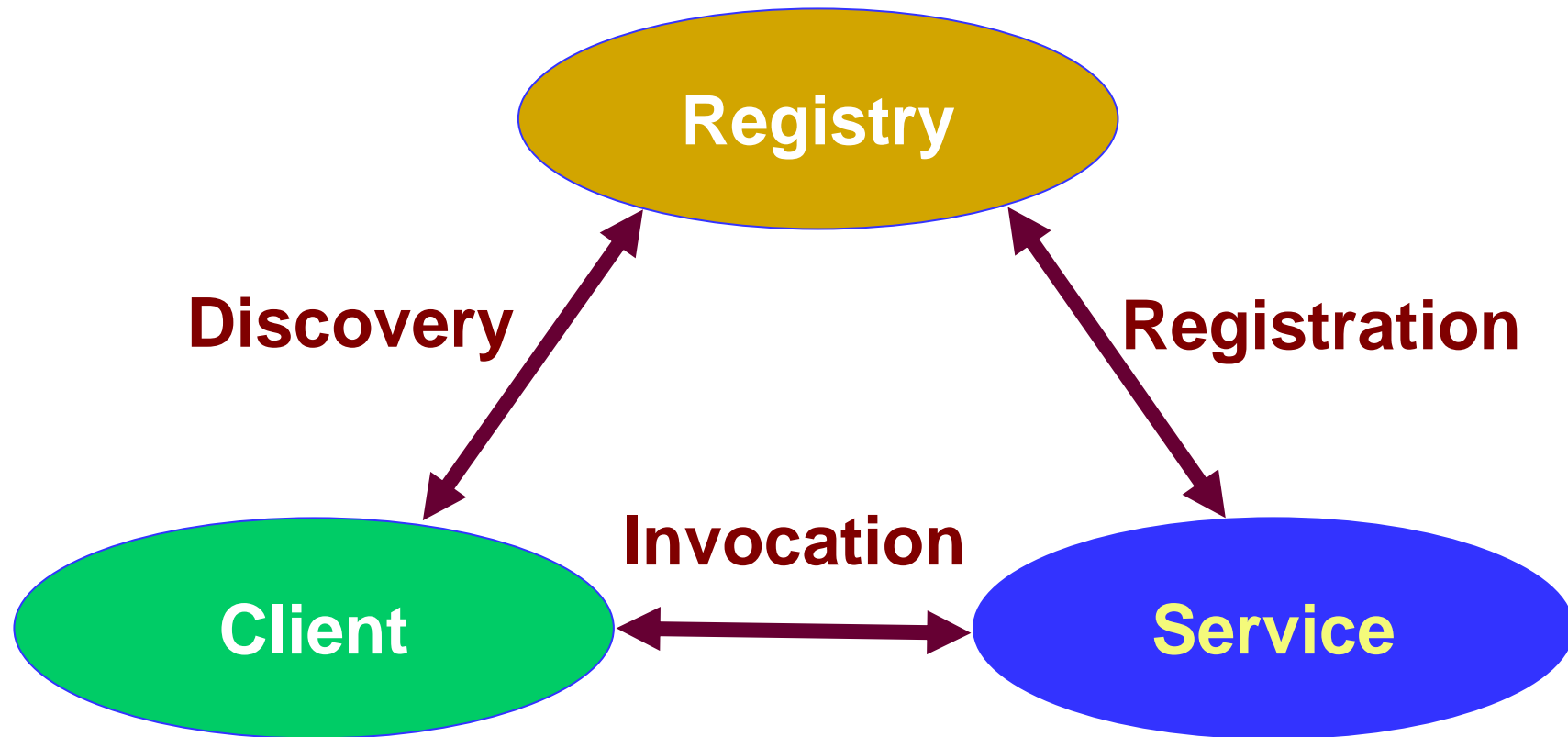**infrastructure for the information society**

eGee

- **History**
  1. Web browsing
  2. Web pages with content from applications
  3. Applications that are useable by software clients

- **Web Services are software components that are..**
  - Accessible across a network
  - Loosely coupled
    - Defined by the messages they receive / send
    - Modular and self-contained
    - So can change service implementation without changing interfaces
  - Interoperable: each service has a description that is accessible and can be used to create software to invoke that service
- **… and based on standards**
  - Usually built on (extensions of) standards made ubiquitous by the Web: http(s), XML, … and for which tools are already built.
  - Developed in anticipation of new uses – e.g. can compose workflow
  - Encouraging adoption

- **Accessible across a network**

- **Loosely coupled, defined by the messages they receive / send**

- **Interoperable: each service has a description that is accessible and can be used to create software to invoke that service**

- **Based on standards (for which tools do / could exist)**

- **Developed in anticipation of new uses**

eGee

Enabling Grids for E-sciencE

eGee

service A:
consumes B

service B

Messages

Web services
container

Web services
container

O/S

TCP/IP

O/S

A

messages

B
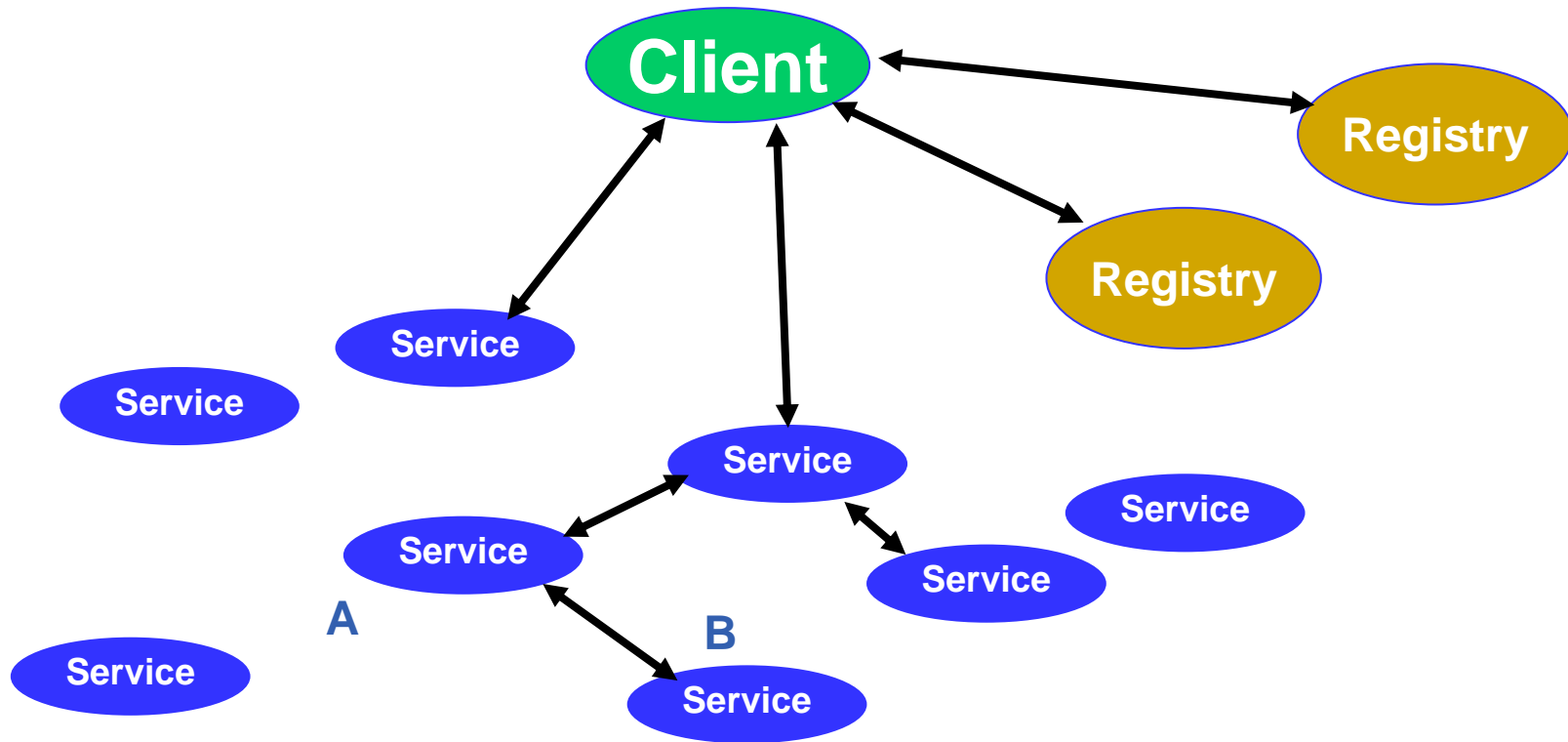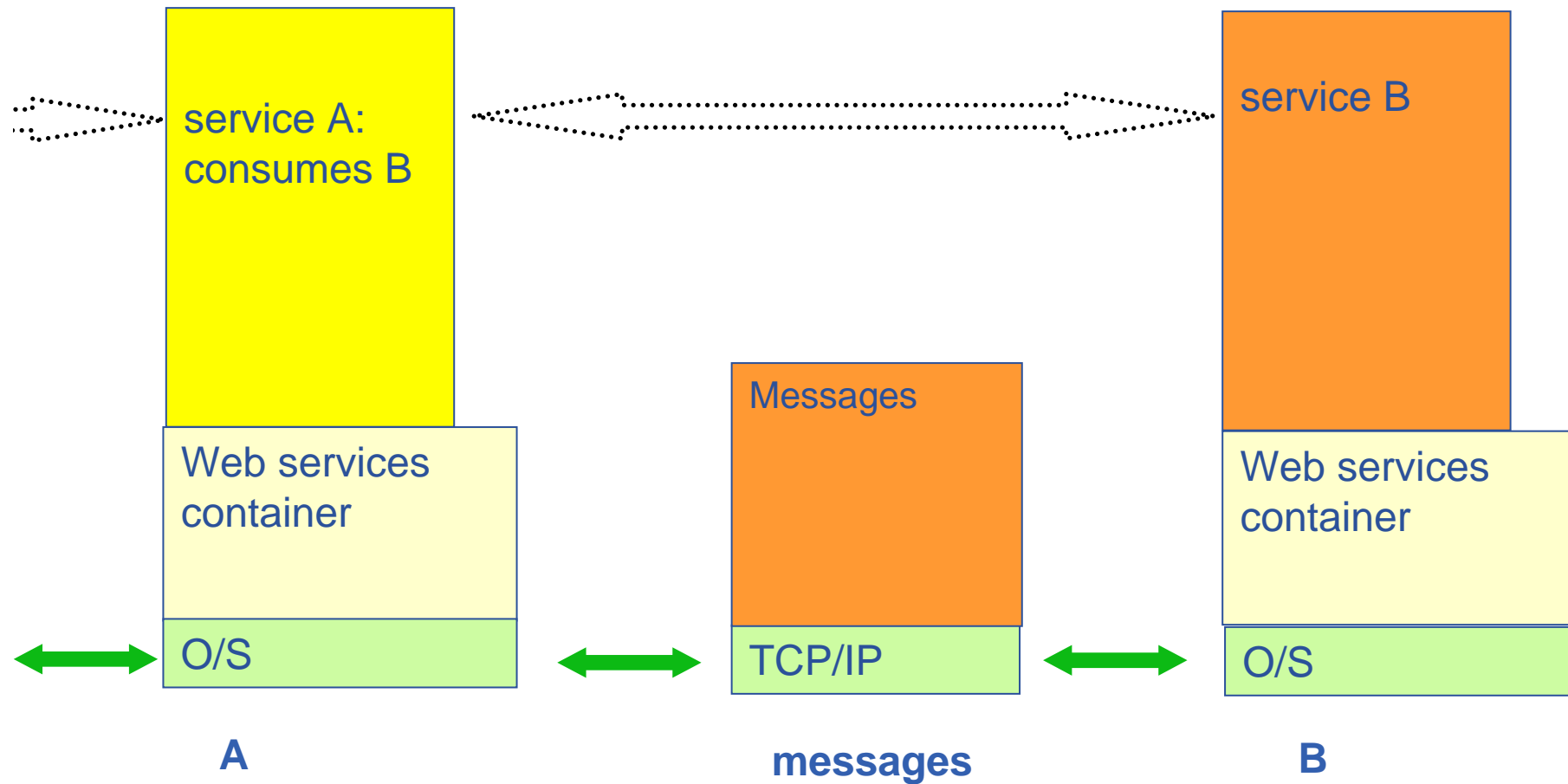
These messages define service B

service A:
consumes B

service B

Messages

Web services
container

Web services
container

O/S

TCP/IP

O/S

A

messages

B

eGee
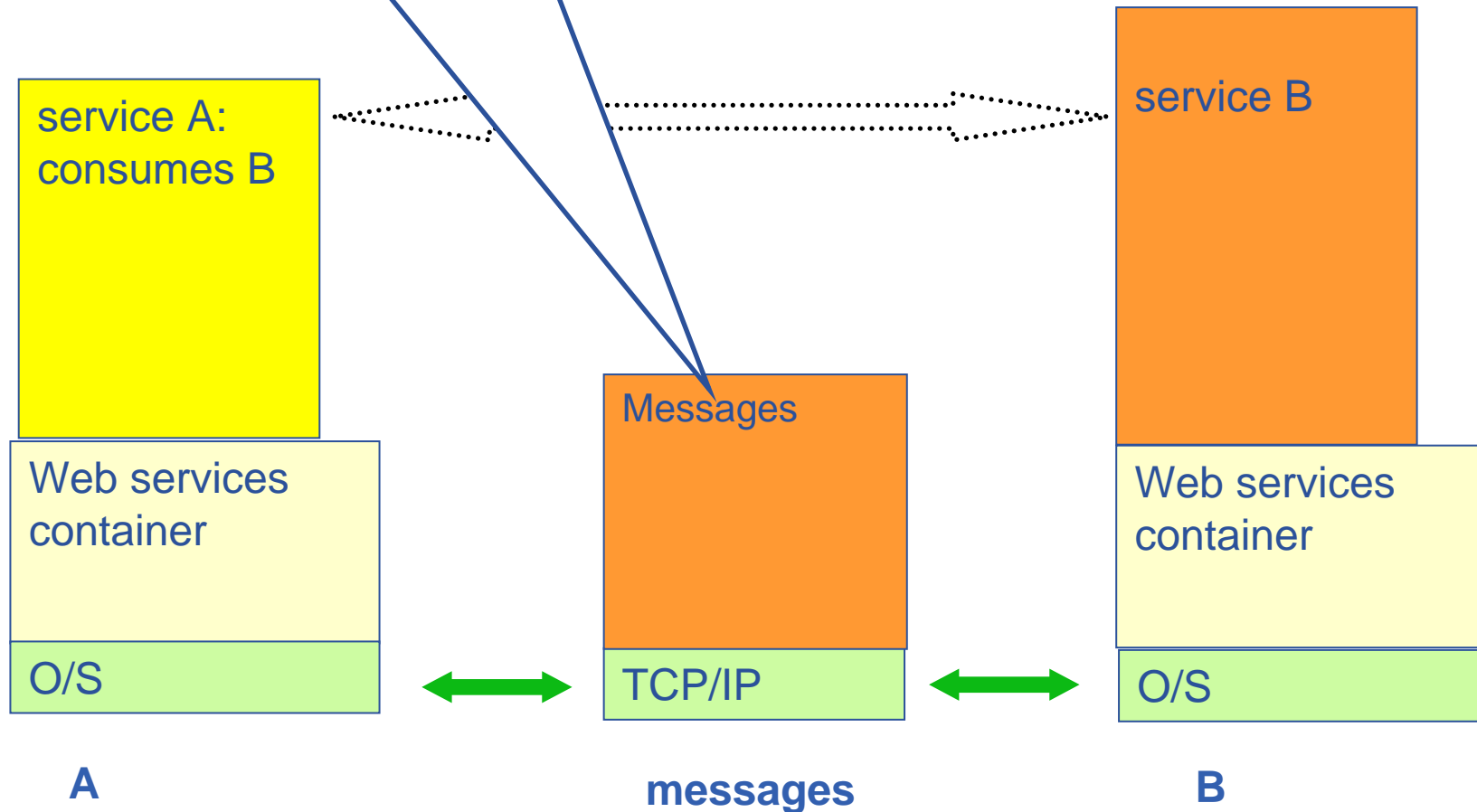
- **Commonly used for WS - original purpose: carry HTML**
- **HTTP request methods**
  - **GET : specify a URL**
  - **POST: key-value pairs**
- **Defaults to port 80**
- **HTTPS: encrypts / decrypts**

service B

Messages

Web services container

http, https

O/S

TCP/IP

Web services container

O/S

A

messages

B

eGee

XML

eXtendable Markup Language

- **Human readable (sort of) – so not tied to one architecture**
- **Extendable: can define new data types, belong to "namespaces", expressed by schemas**

service B

Web services container

O/S

messages

XML

http, https

TCP/IP

Web services container

O/S

A

messages

B

eGee

Enabling

Service B is described by WSDL, "Web Service Description Language". Includes:

- *Where*: " end point" URL where B receives messages
- *What*: Message definitions
- *How*: "Bindings" to use service

service A: consumes B

service B

Messages

Web services container

XML

http, https

O/S

TCP/IP

Web services container

O/S

A

messages

B

Enabling Grids for E-sciencE
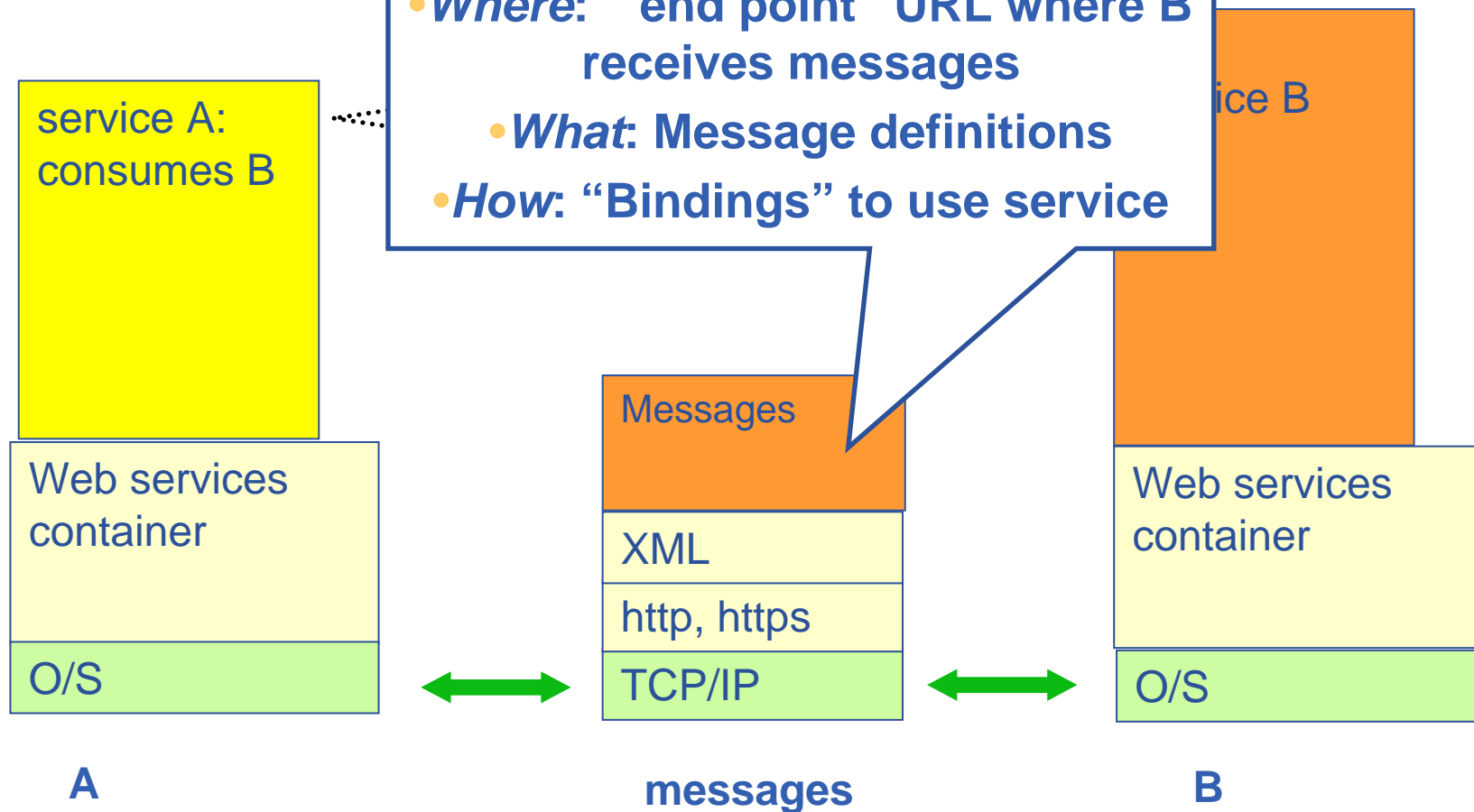
- **API's are derived from the WSDL by tools**
- **Developer adds code for specific client**
- **NO API LIBRARY SUPPLIED**

service A: consumes

API's for B

service B

Web services container

O/S

Messages

XML

http, https

TCP/IP

Web services container

O/S

A

messages

B

Enabling Grids for E-sciencE

> **XML-based messaging protocol**
> **Header: extendable for security, accounting,**
> **Body: message content**

service A:
consumes B

API's for
B

Web services
container

O/S

Messages

SOAP

http, https

TCP/IP

service B

Web services
container

O/S

A

messages

B

Enabling Grids for E-sciencE

This determines whether code is in Java, C, Perl, Python…

service A consume

API's for B

Web services container

O/S

A

Messages

SOAP

http, https

TCP/IP

messages

service B

Web services container

O/S

B

egee

WSDL for B → JAXRPC **deploy**

compile

Insert the web into the invoke and into the return

A   B

invoke

B.Op (a,b)

return

c

Java program with 2 classes

A   B-stub   Stack

invoke

Bs.Op (a,b)

...

return

c

...

Consumes B (client)

Op(a,b)

Soap envelope

HTTP message

Web

Ret(c)

Soap envelope

HTTP message

Sk Stack eleton   B

...

invoke

B.Op (a,b)

return

c

...

Provides B (server)
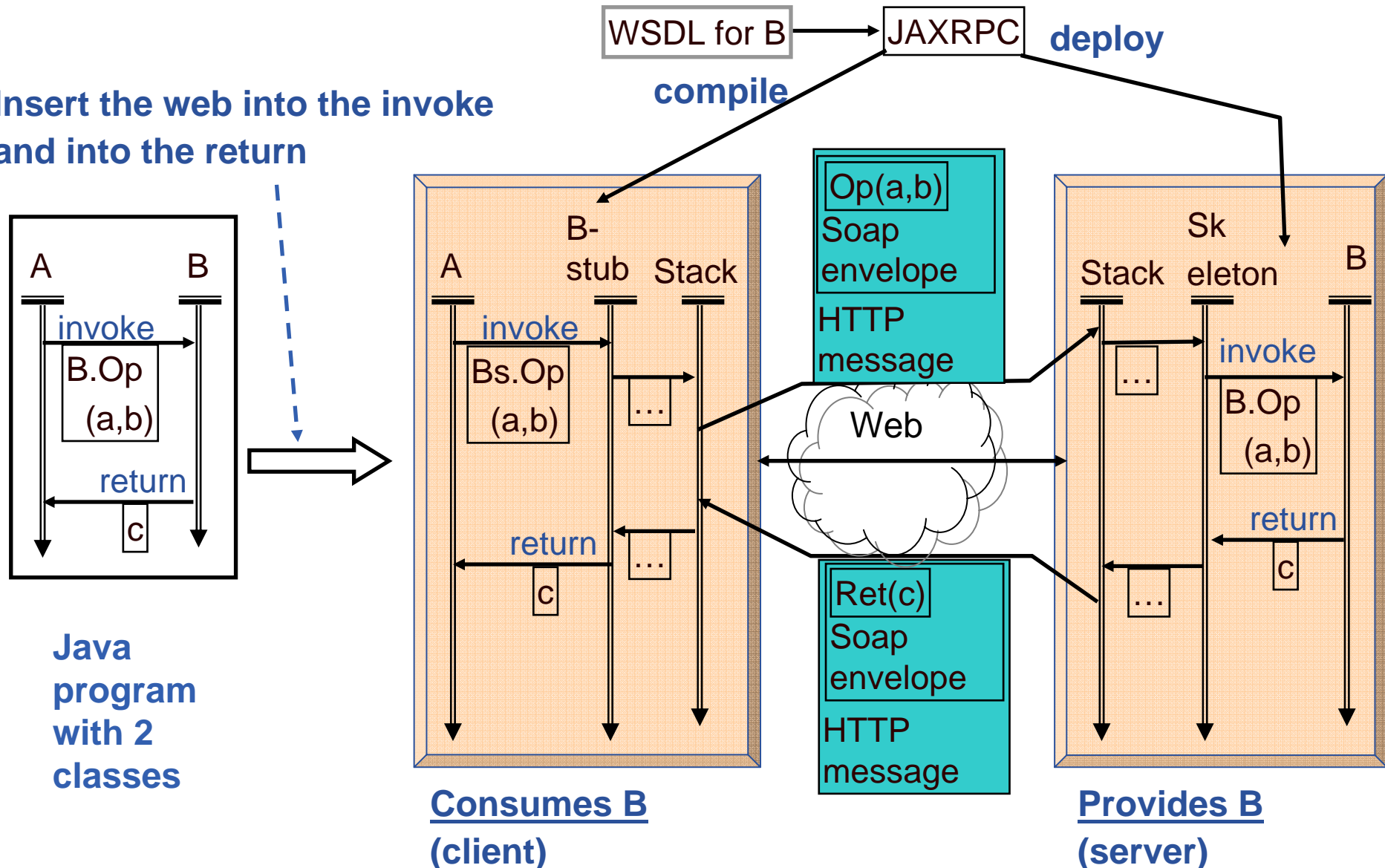
- **WS-I (Interoperability) delivers practical guidance, best practices and resources for developing interoperable Web services solutions.**

- **http://www.ws-i.org/**

Open standards:

- **SOAP: protocol for message passing**

- **Web Service Description Language: to describe services**

- **UDDI: Universal Description, Discovery and Integration**

- **WS-Security: incorporates security**
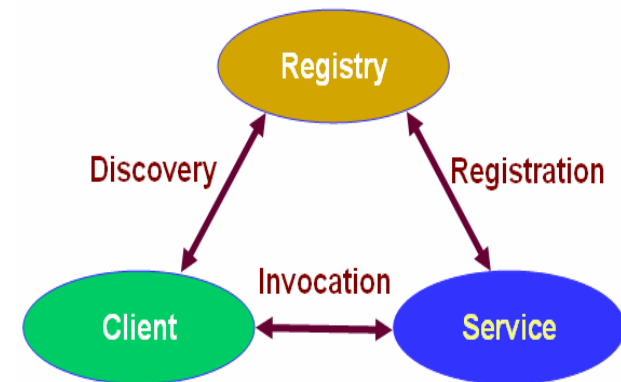
Enabling Grids for E-sciencE

- " Web Services are *the* way to build Grids"

- Web Services

- Relevance of Web Services to Grids ⟸

- Extending WS for grids

- So where are we now ?

Enabling Grids for E-sciencE

- **Software components that are..**
  - Accessible across a network
  - Loosely coupled
    - Defined by the messages they receive / send
    - Modular and self-contained
    - So can change service implementation without changing interfaces
  - Interoperable: each service has a description that is accessible and can be used to create software to invoke that service

- **… and based on standards**
  - Tools, interoperability, …
  - Developed in anticipation of new uses – e.g. can compose workflow
- **i.e. what web services exist for!**
- **So now building grid architecture based on WS**
- **But there are additional challenges!!!!!**

eGee
Enabling Grids for E-sciencE

Web Services

- **Goals**
  - Computational presentation & access of Enterprise services
  - Marketing integrated large scale software and systems
  - Model for independent development
  - Model for independent operation

Grids

- **Goals**
  - Inter-organisational collaboration
  - Sharing information and resources
  - Framework for collaborative development
  - Framework for collaborative operation

eGee
Enabling Grids for E-sciencE

Web Services

- **Complex services created & delivered persistently by owner organisation**
- **Client interactions short-lived**
- **Multi-organisation integration responsibility of client**
  - Workflow enactment
  - Transaction coordination
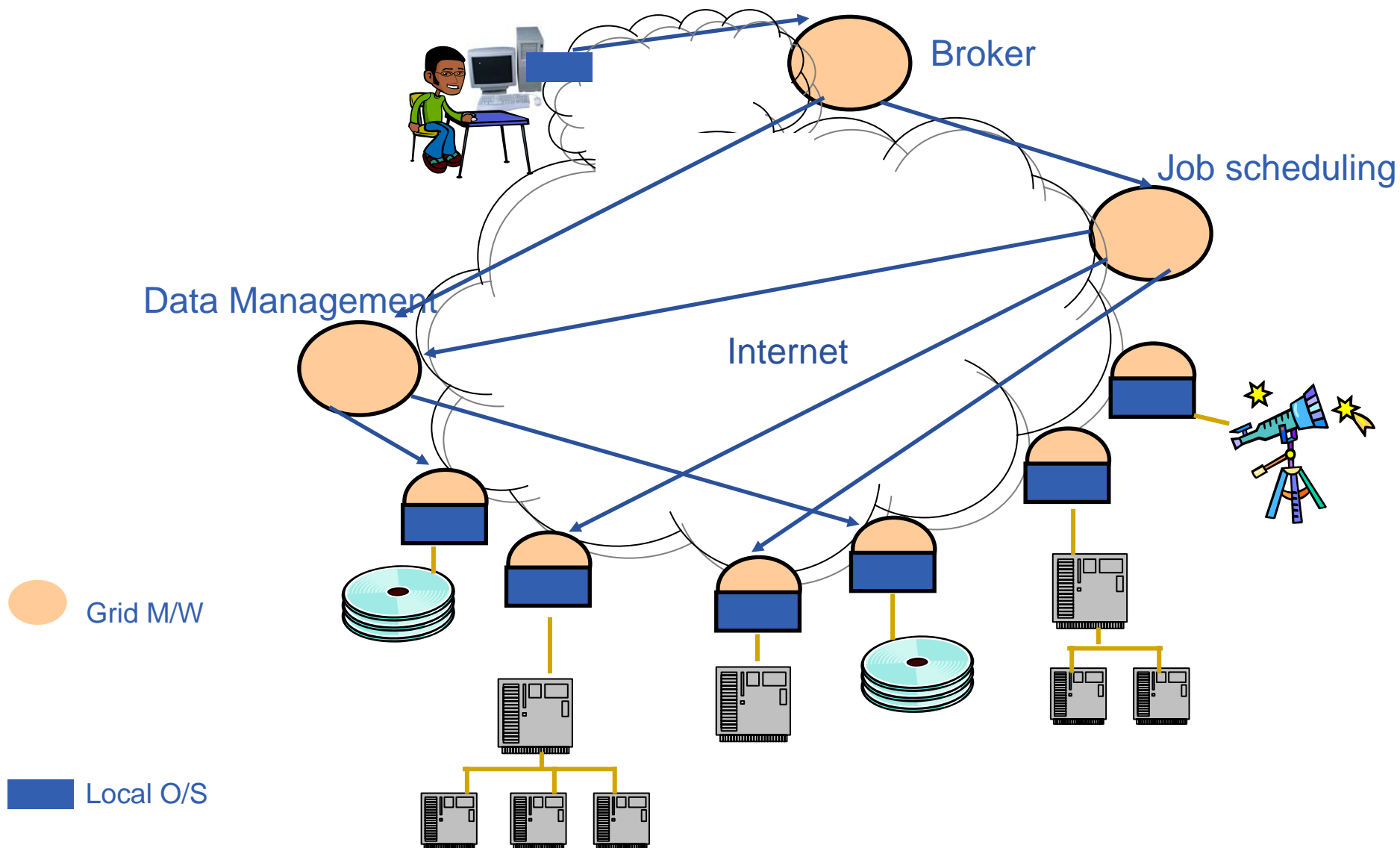 - May be by an intermediate service

Grid Services

- **All of WS patterns +**
- **Dynamic services / resources**
- **Long-lived interactions**
- **Persistent computational integration**
  - Data management
  - Computation management
- **Persistent operational infrastructures**
  - EGEE managing European-scale grid
- **System organised optimisation**
- **End-to-end security (and non-repudiation)**
- **Virtual Organisations**
  - Establish multi-organisation security policies

eGee

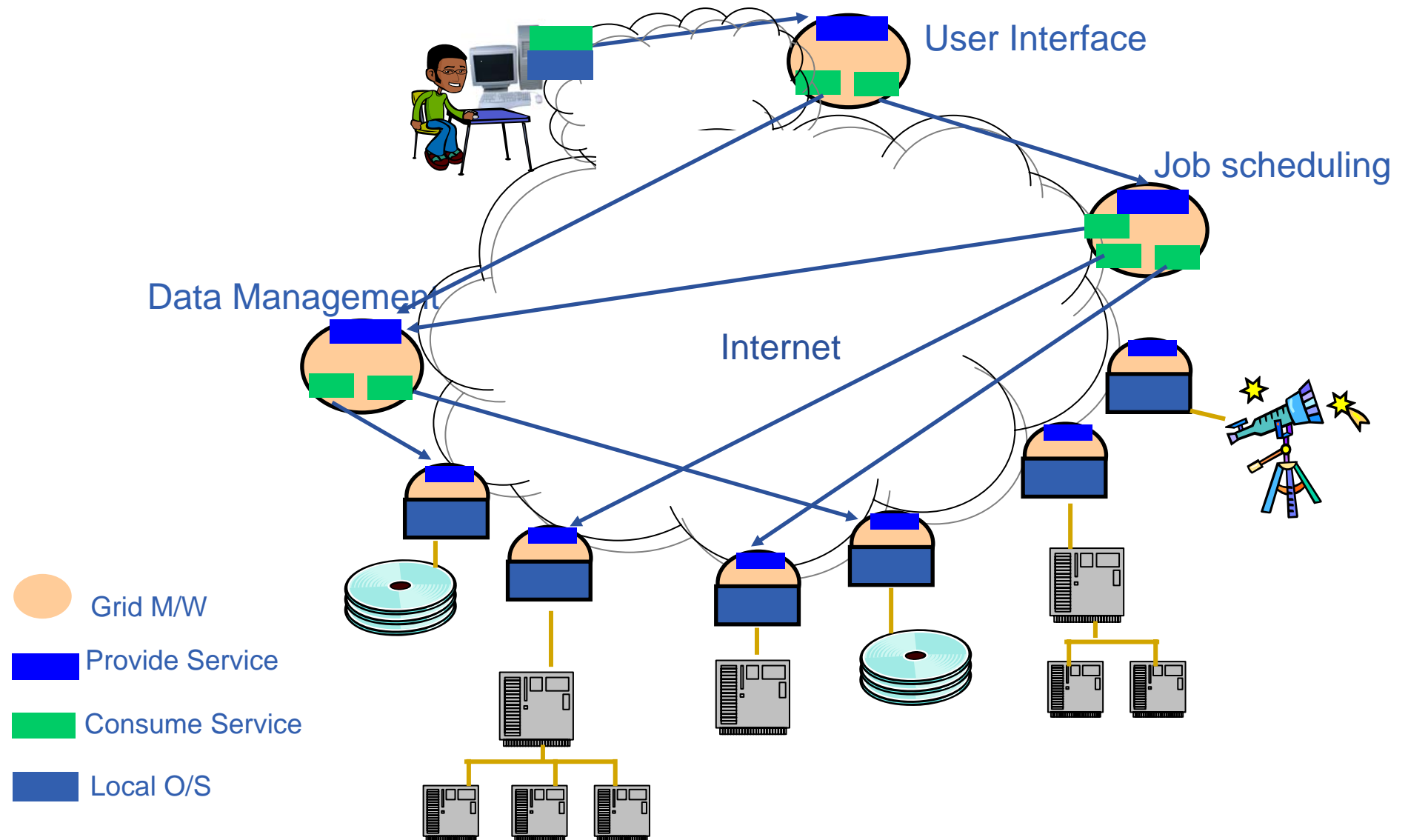Enabling Grids for E-sciencE

Web Services

- **Commercially successful operational applications**
- **Several good toolsets available**
  - Mostly costly to use outside academia
- **Workflow enactment**
  - BPEL4WS
- **Scale, usability & reliability problems in free-ware**
  - Many fixes were needed to Apache Tomcat
- **Much momentum**
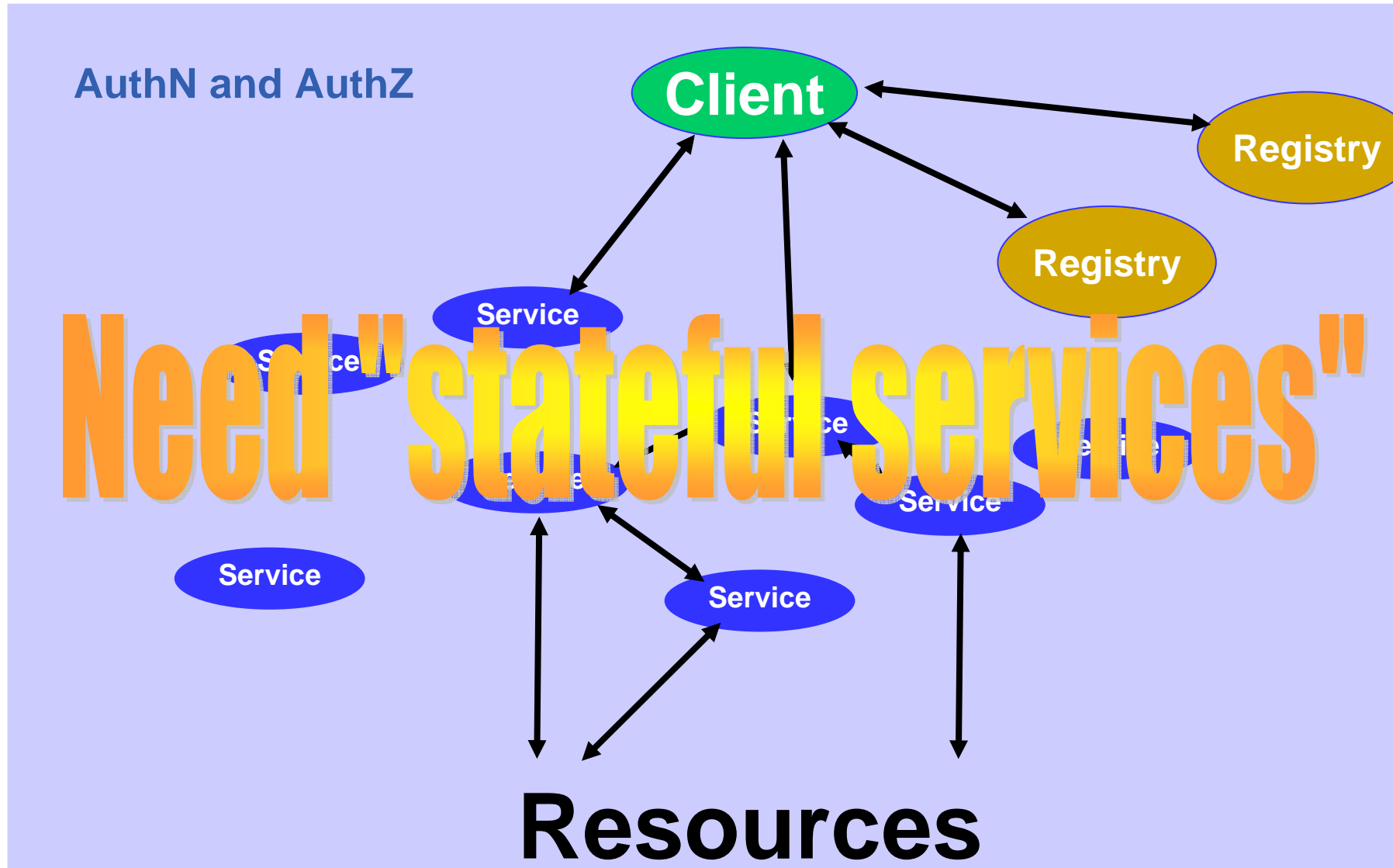 - Very high levels of investment

Grids

- **Operational research projects and grids**
  - >100 projects use GT2 or GT3
- **No toolsets**
- **Scientific workflow**
  - High-level work-load generators
  - Chimera, Pegasus, Taverna, …
- **Some very robust and well tested technologies**
  - Condor, GT2, VDT, GT3.2, LCG2, EGEE1
- **All free-ware**
- **Performance, usability and reliability problems**
- **Much momentum**
- **High levels of investment**

Broker

Job scheduling

Data Management

Internet

Grid M/W

Local O/S

eGee

User Interface

Job scheduling

Data Management

Internet

Grid M/W

Provide Service

Consume Service

Local O/S

eGee

AuthN and AuthZ

Client

Registry

Registry

Service

Service

Service

Service

Service

Service

Need "stateful services"

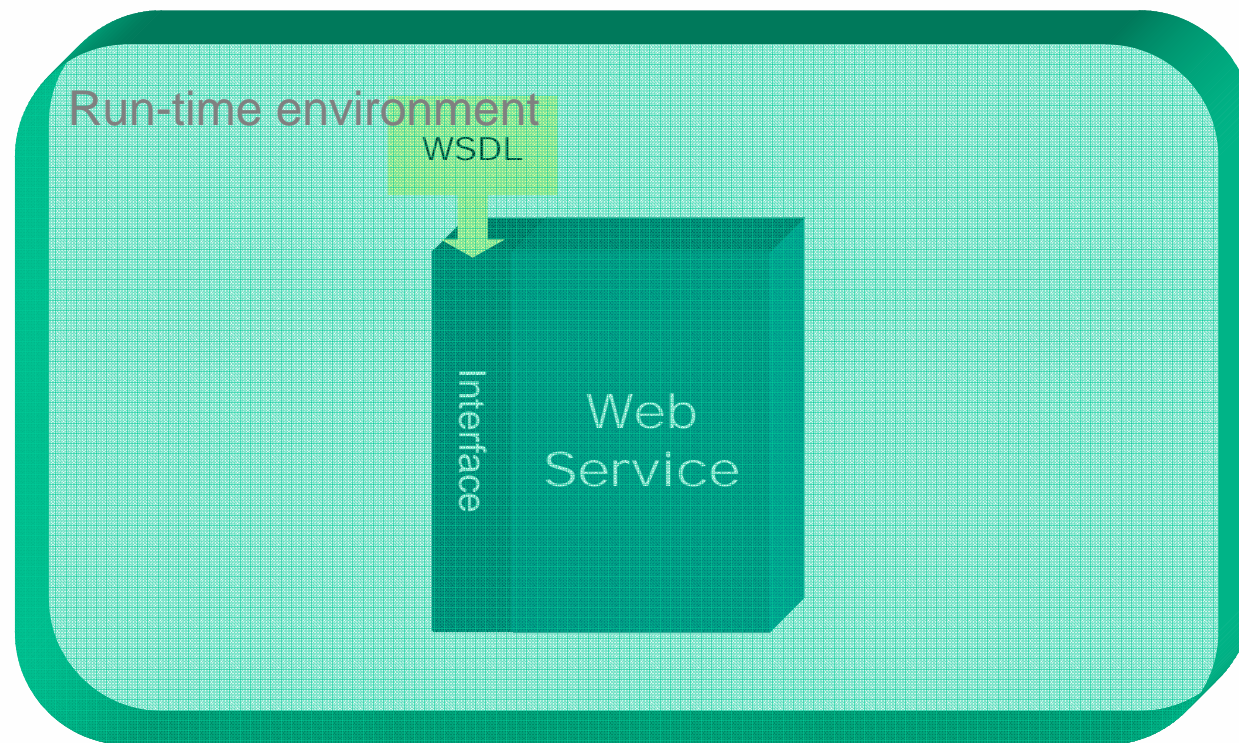**Resources**

Enabling Grids for E-sciencE

- **"Open grid services architecture" OGSA– proposed in 2001**
- **Open Grid Services Infrastructure**
  - Globus Toolkit 3 resulted
  - Specified in 2003
- **Then in January 2004**
  - OGSI to be replaced by emerging WS-RF (Web Services Resource Framework)
- **NOTE:**
  - OGSA still under development (GGF)
- **Imbalances in OGSI that are addressed by WS-RF (OASIS)**
  - WS community not engaged
  - Over O-O, megalithic

WS-Resource Framework Capabilities

★ **Specifies how to use XML to describe and access a resource's properties**

★ **Clarifies how stateful resources are addressed**

★ **Defines how a resource is created and messages to destroy resources**

★ **Provides a message subscription and notification mechanism for Web services**

○ **Outlines how to organize groups of resources and services**

○ **Adds a fault tolerance capability to WS-Addressing**

○ **Defines a standard, extensible format for Web services error messages**

The WS-Resource framework model

Web Service

The WS-Resource framework model

Invoking a Web Service

The WS-Resource framework model

- **What is a WS-Resource**

  - **Examples of WS-Resources:**
    - Physical entities (e.g.. processor, communication link, disk drive) or Logical construct (e.g.. agreement, running task, subscription)
    - Real or virtual
    - Static (long-lived, pre-existing) or Dynamic (created and destroyed as needed)
    - Simple (one), or Compound (collection)

  resource

  - **Unique - Has a distinguishable identity and lifetime**
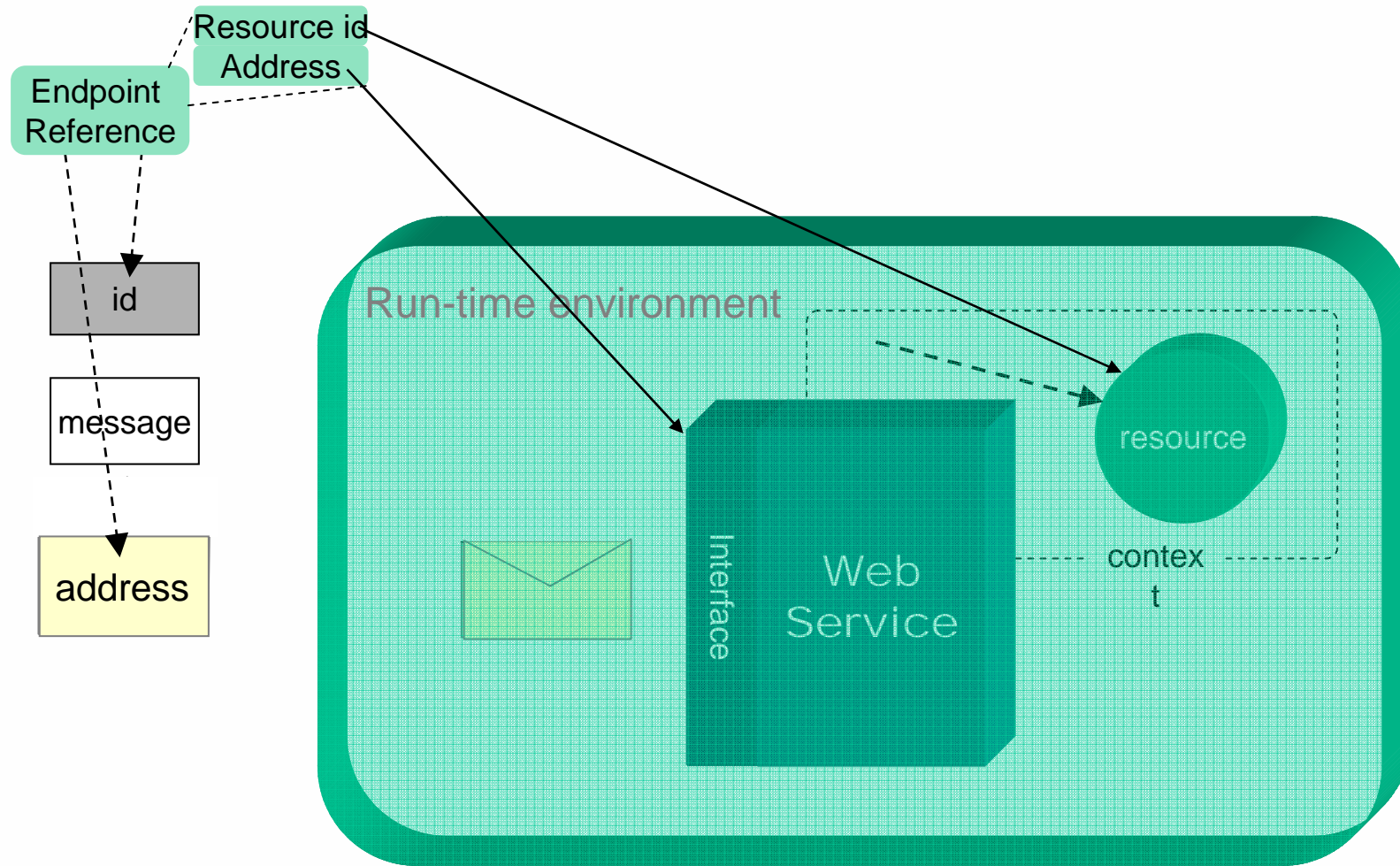
  - **Stateful - Maintains a specific state that can be materialized using XML**

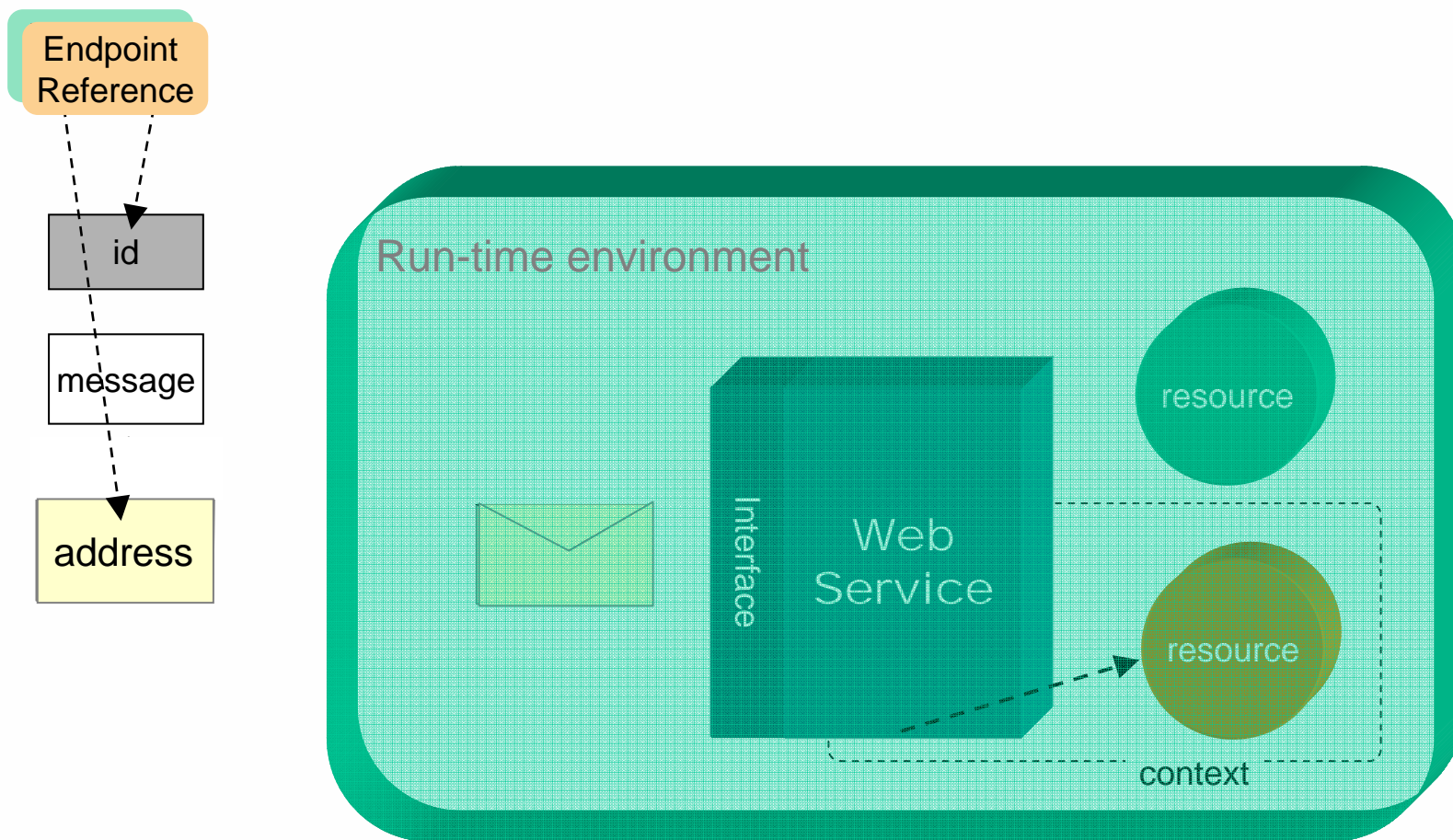 - **May be accessed through one or more Web Services**

The WS-Resource framework model

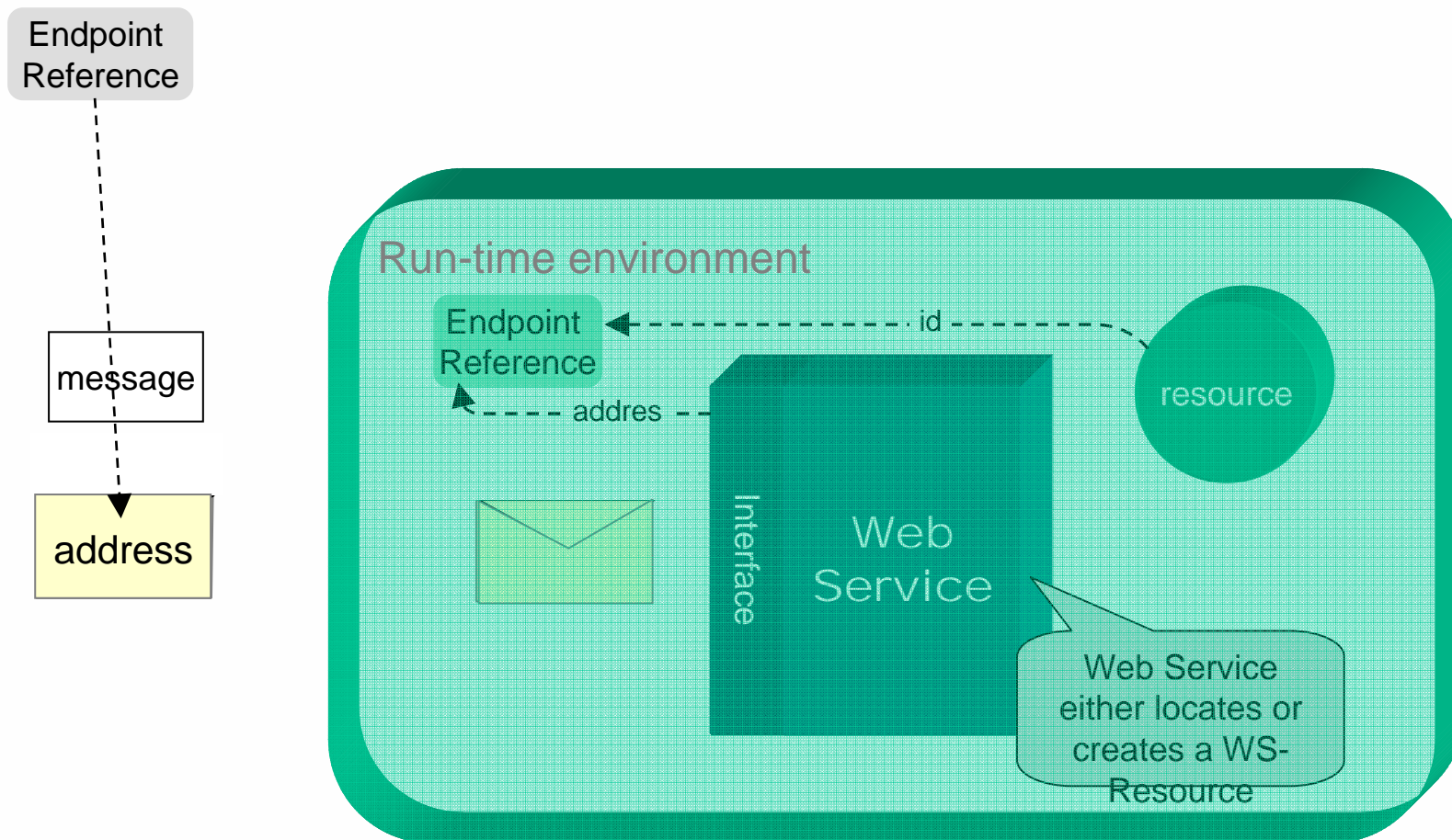## Using a Web service to access a WS-Resource

The WS-Resource framework model

Using a Web service to access a WS-Resource

The WS-Resource framework model
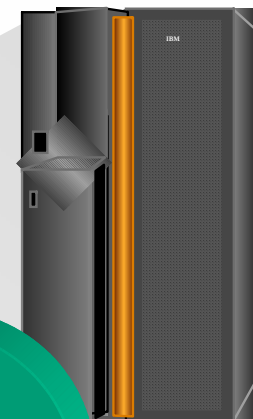
Creating / Locating a WS-Resource

The WS-Resource framework model

- ## 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
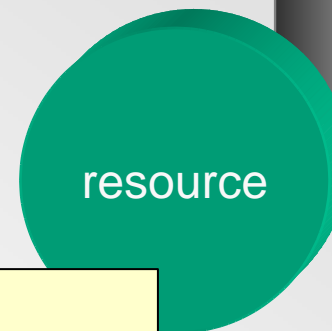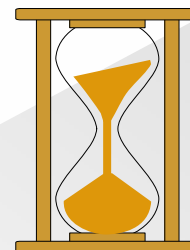
resource

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

</ProcessorProperties>
```

WS-Notification

- **Subscriber indicates interest in a particular "Topic" by issuing a "subscribe" request**

- **Broker (intermediary) permits decoupling Publisher and Subscriber**

- **"Subscriptions" are WS-Resources**

  - **Various subscriptions are possible**
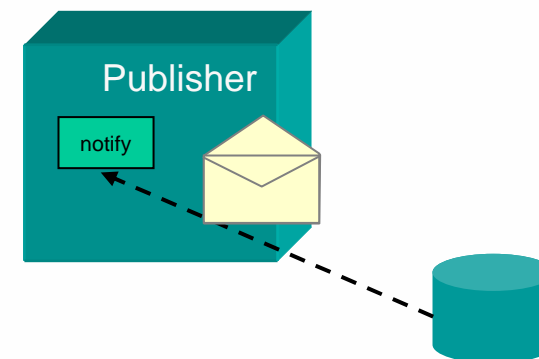
- **Publisher need NOT be a Web Service**

- **Notification may be "triggered" by:**

  - **WS Resource Property value changes**
  - **Other "situations"**

- **Broker examines current subscriptions**

- **Brokers may**

  - **"Transform" or "interpret" topics**
  - **Federate to provide scalability**

Subscriber

subscribe

Broker

notify

S S S

Publisher

notify
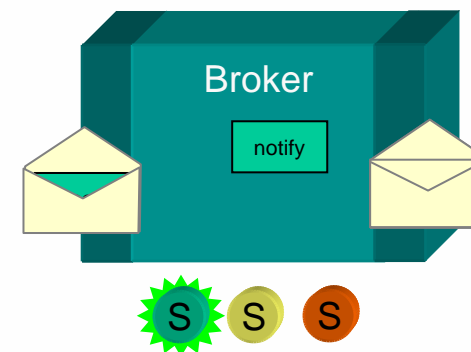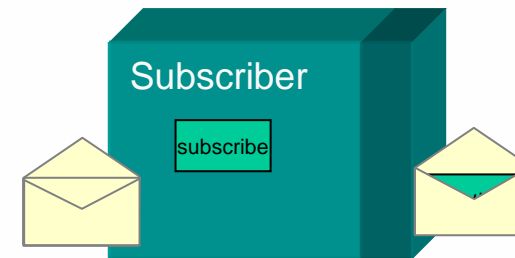
Bringing it All Together

Scenario: Resource management & scheduling

Local processor manager is "f...ded" with A W...vice interface

Grid "Jobs" and "..." are also modeled... WS-Resource... Resource Properties

Grid Scheduler is a Web Service

Service Level Agreement is modeled as a WS-Resource

Notification...WS-R.......sed

...fication can be ..."inform" the... scheduler when processor utilizatio... changes

Lifetime of SLA Resource tied to the duration of the agreement

utilization)

Blades

Storage

J J J

S

A

R R R • • • • •

Enabling Grids for E-sciencE

Web service itself
  is stateless

Front end to multiple instances of
  back-end for each resource

Maintains state in a back-end

Service request identifies the specific
  resource

Enabling Grids for E-sciencE

- **WSRF builds on**
  - WS-Addressing – W3C submission Aug 2004
  - WS-Notification
    - WS-BaseNotification
    - WS-BrockeredNotification
    - WS-Topics

- **WSRF comprises standards**
  - WS-ResourceLifetime
  - WS-ResourceProperties
  - WS-RenewableReferences
  - WS-ServiceGroup
  - WS-BaseFaults

Enabling Grids for E-sciencE

- **Standards are emerging… some near acceptance and some being discarded**
  - Standards bodies:
    - W3C   http://www.w3c.org/
    - GGF   http://www.ggf.org/
    - OASIS   http://www.oasis-open.org/home/index.php
    - IETF   http://www.ietf.org/
  - For a summary see http://www.innoq.com/soa/ws-standards/poster/

- **Production grids are based on de-facto standards at present**
  - Inevitably!
  - GT2 especially
  - But locks a grid into one middleware stack unable to benefit from the diverse developments of new services

- **Some confusion remains after the OGSI era**
  - Many projects sidestepped this by using "pure" WS

- **Globus Toolkit 4 has been released**

- **HP-IBM-Intel-Microsoft Roadmap**
- **Globus comments:**
- **http://www.globus.org/wsrf/convergence.php**
  - "reconciling two similar but competing approaches"
  - the Web Services Distributed Management (WSDM) family of specifications (including Web Services Resource Framework (WSRF) and WS-Notification (WS-N))
    - IBM, HP, and others
  - WS-Management family of specifications (including WS-Transfer, WS-Eventing, and WS- Enumeration)
    - Microsoft, Intel, and others
  - "Globus will also work to provide a painless migration path for GT4-based services and clients".
  - "While detailed specifications are not yet available, we are confident, based on knowledge of the existing specifications that are to be reconciled, and the published roadmap, that such a migration path will be easy to achieve".

egee

Grid applications are based on the high-level services defined by OGSA (i.e. not implemented from scratch using WSRF)

Standards in the works (GGF)
- VO management
- Security
- Resource management
- Job Management
- Data services
- etc.

GT4 includes many of the services required by OGSA

Applications

OGSA

WSRF

Web Services

Standardized (Oasis) and implemented (GT4)

Standardized (W3C) and implemented (e.g. Apache Axis)
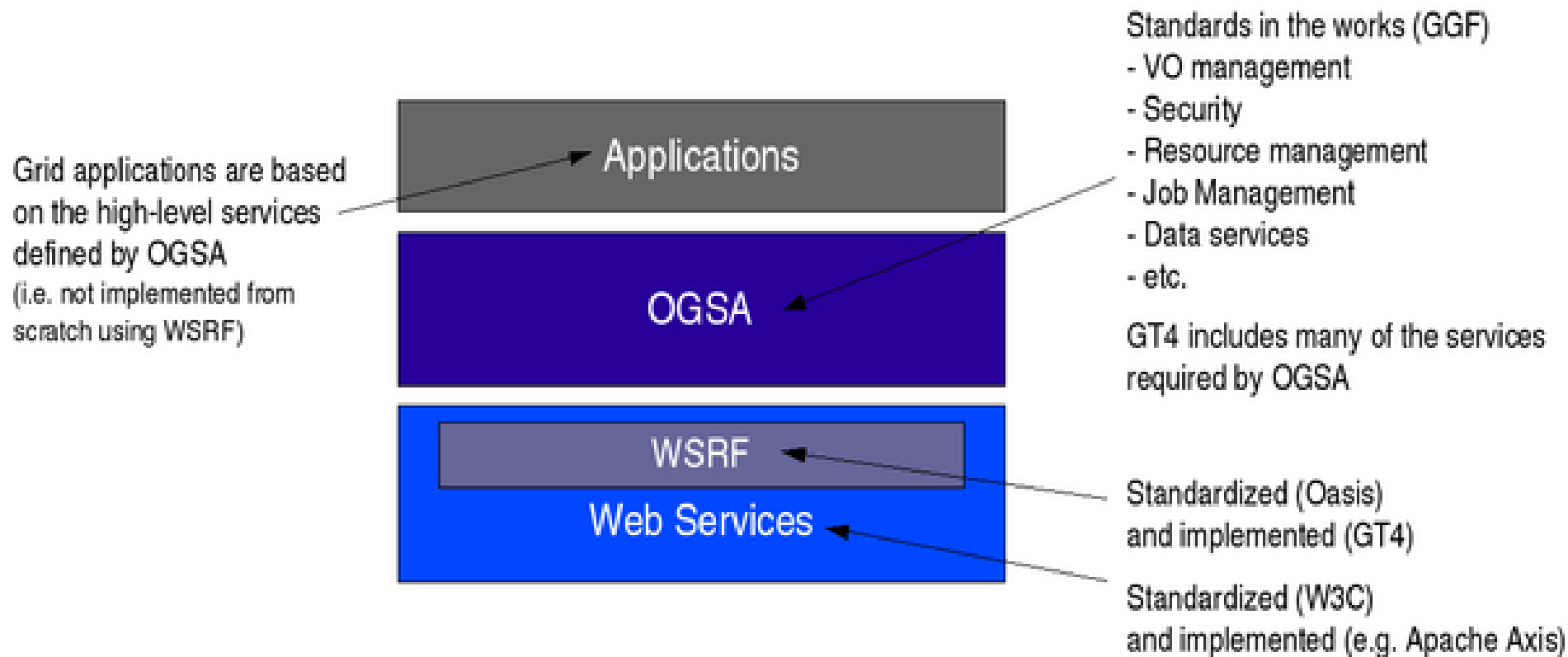
Diagram from Globus Alliance

Diagram from Globus Alliance

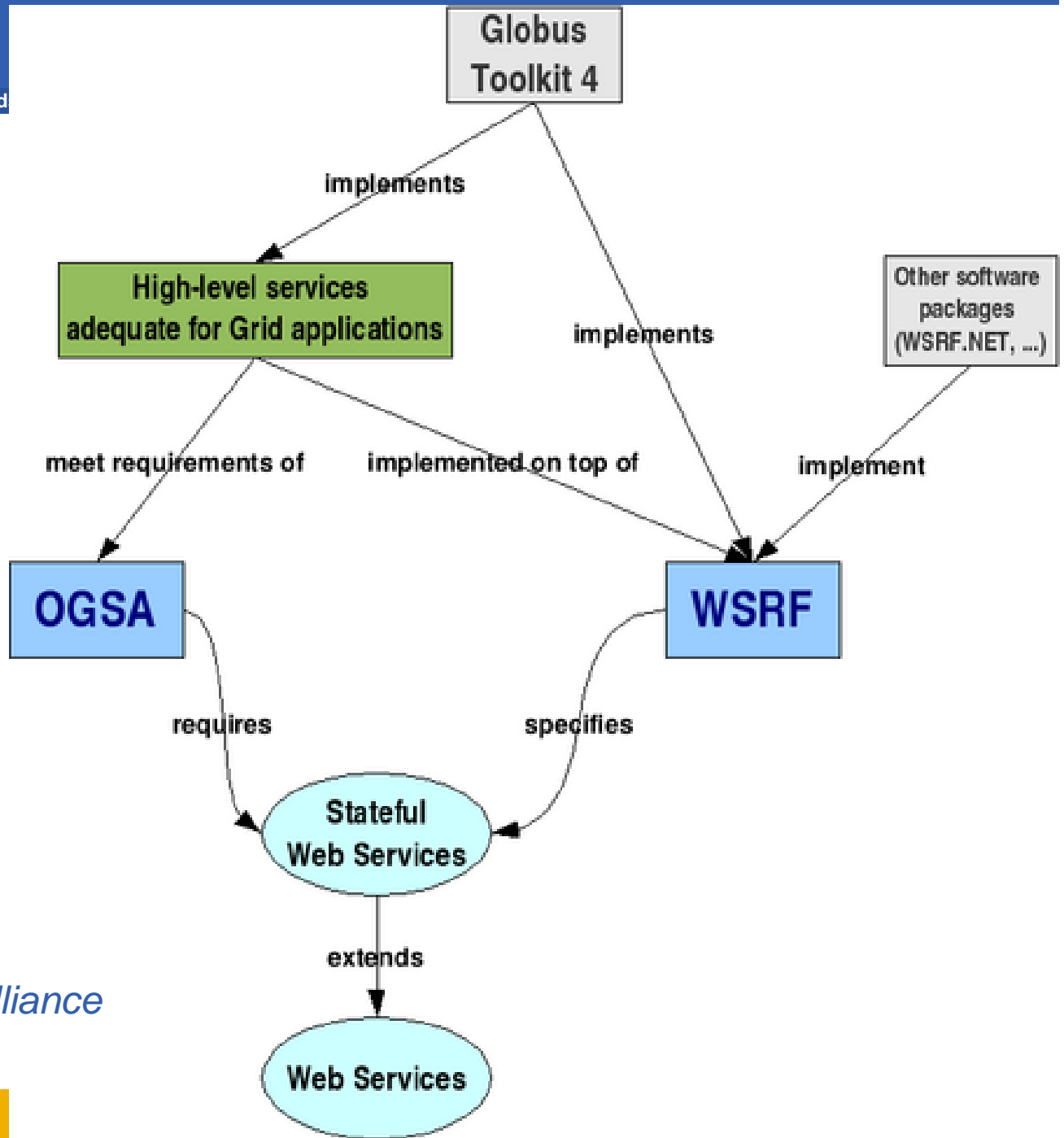**Enabling Grids for E-sciencE**

- **VDT: Virtual Data Toolkit**
  - ensemble of grid middleware that can be easily installed and configured
  - Been used by LCG and EGEE with GT2, Condor, MyProxy,…

- **Pre-requisite for using GT4 in gLite and other production grids has been achieved:**

- **"VDT 1.3.7 introduces the Globus Toolkit 4.0 (GT4) series – both pre-web services and some web services."**

http://vdt.cs.wisc.edu/globus_3.2_vs_4.0.html

Enabling Grids for E-sciencE

- " Web Services are _the_ way to build Grids"

- Web Services

- Relevance of Web Services to Grids

- Extending WS for grids

- So where are we now ?

- **Where might we be going?!**

Decomposition Enables Separation of **Concerns** & **Roles**

the globus alliance
www.globus.org

User

"Provide access to data D at S1, S2, S3 with performance P"

Service Provider

"Provide storage with performance P1, network with P2, ..."

Resource Provider

Ian Foster GGF16

D

S1

S2

S3

D

S1

S2

S3

Replica catalog, User-level multicast, ...
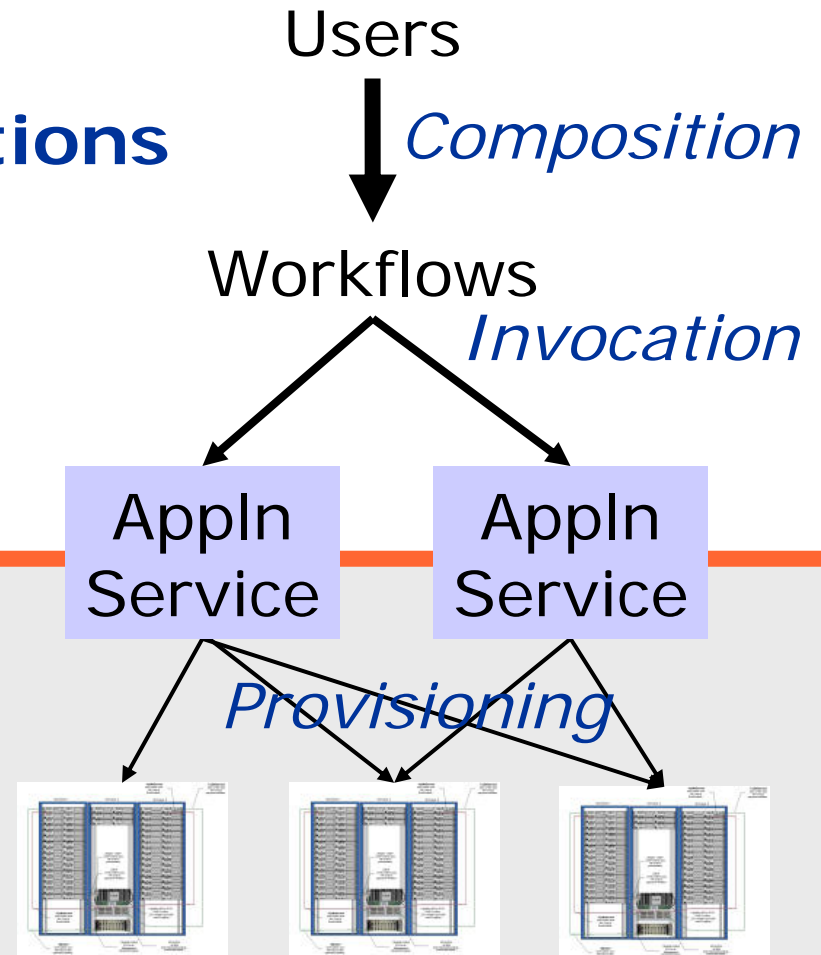
D

S1

S2

S3

the globus alliance
www.globus.org
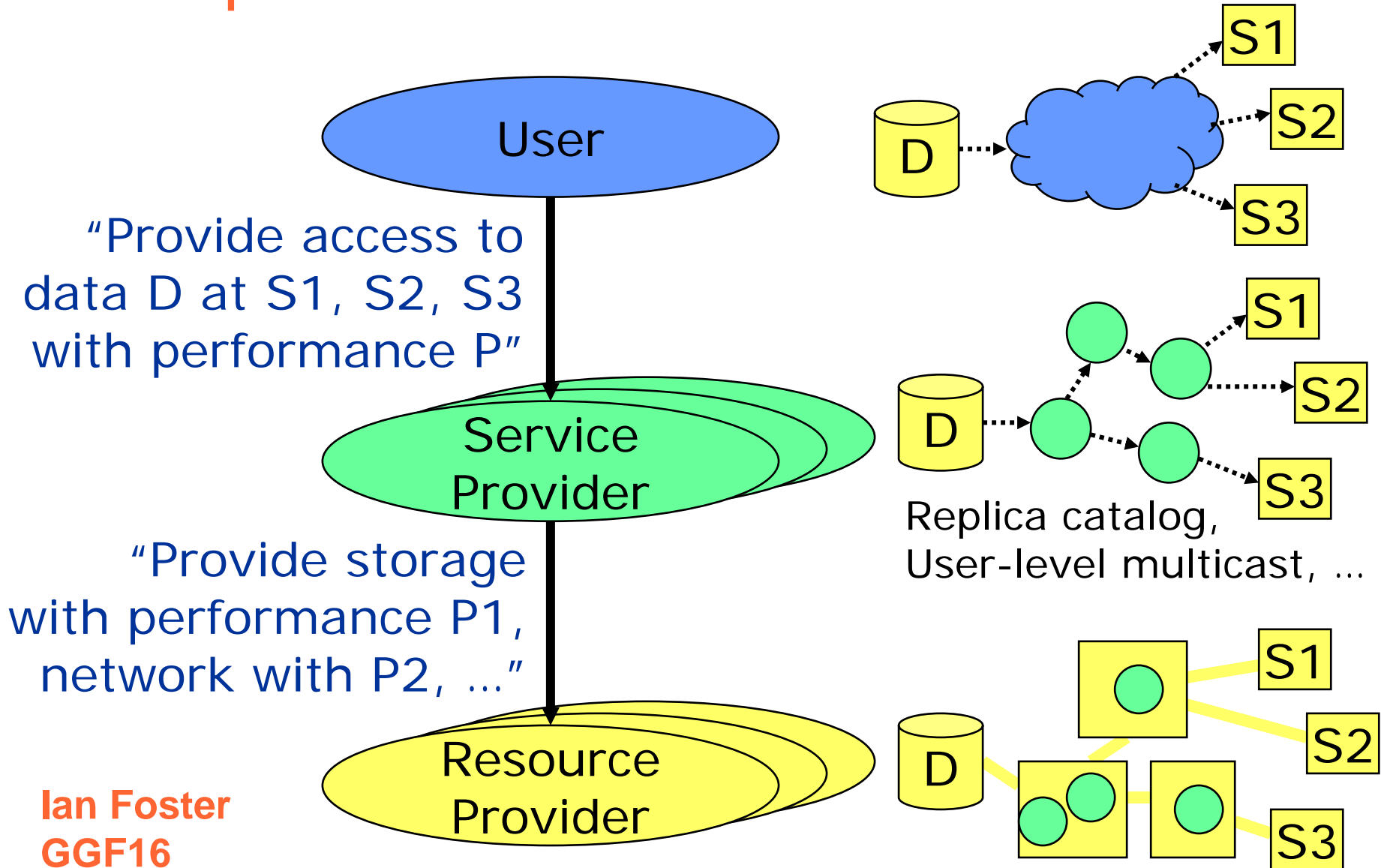
Service-Oriented Systems: The Role of Grid Infrastructure

- **Service-oriented applications**
  - Wrap applications as services
  - Compose applications into workflows

- **Service-oriented Grid infrastructure**
  - Provision physical resources to support application workloads

Users

Composition

Workflows

Invocation

Appln Service   Appln Service

Provisioning

"The Many Faces of IT as Service", ACM Queue, Foster, Tuecke, 2005

- "potential to increase individual and collective scientific productivity by making powerful information tools available to all"

- "Ultimately, we can imagine a future in which a community's shared understanding … is documented also in the various databases and programs that represent—and automatically maintain and evolve—a collective knowledge base. "

Ian Foster, http://www.sciencemag.org/cgi/content/full/308/5723/814?ijkey=aqCCmCFix8LI.&keytype=ref&siteid=sci

Science 6 May 2005

- **Early grids**
  - Resource utilisation
  - A few big-science VOs
    - Trivial parallelism – many concurrent independent jobs
    - Data management – files only

- **Grid-enabling databases**
  - Pre-existing databases accessible from grids
  - Data integration

- **Service-oriented grid: possibilities for**
  - any collaborative research
  - International / national / university resources become accessible
    - With control and AA (authorisation and authentication)

- **The Grid Core Technologies, Maozhen Li and Mark Baker, Wiley, 2005**

- **The Globus Toolkit 4 Programmer's Tutorial Borja Sotomayor, Globus Alliance, http://gdp.globus.org/gt4-tutorial/multiplehtml/index.html**

- **The Web Services Grid Architecture (WSGA) www.nesc.ac.uk/technical_papers/UKeS-2004-05.pdf**

- **http://java.sun.com/xml/webservices.pdf**

- **http://www.globus.org/wsrf/**

Enabling Grids for E-sciencE

- **Current way people try to create grid middleware is using Service Oriented Architectures based on WS**

- **An abundance of standards is en route**
  - Extensions to manage resources are in WS-RF framework
  - Workflow – service composition
  - Also portals/portlets to expose services

- **Initial implementation based on WS-RF and OGSA is in Globus Toolkit 4**

- **Service-orientation is transforming business and research!**