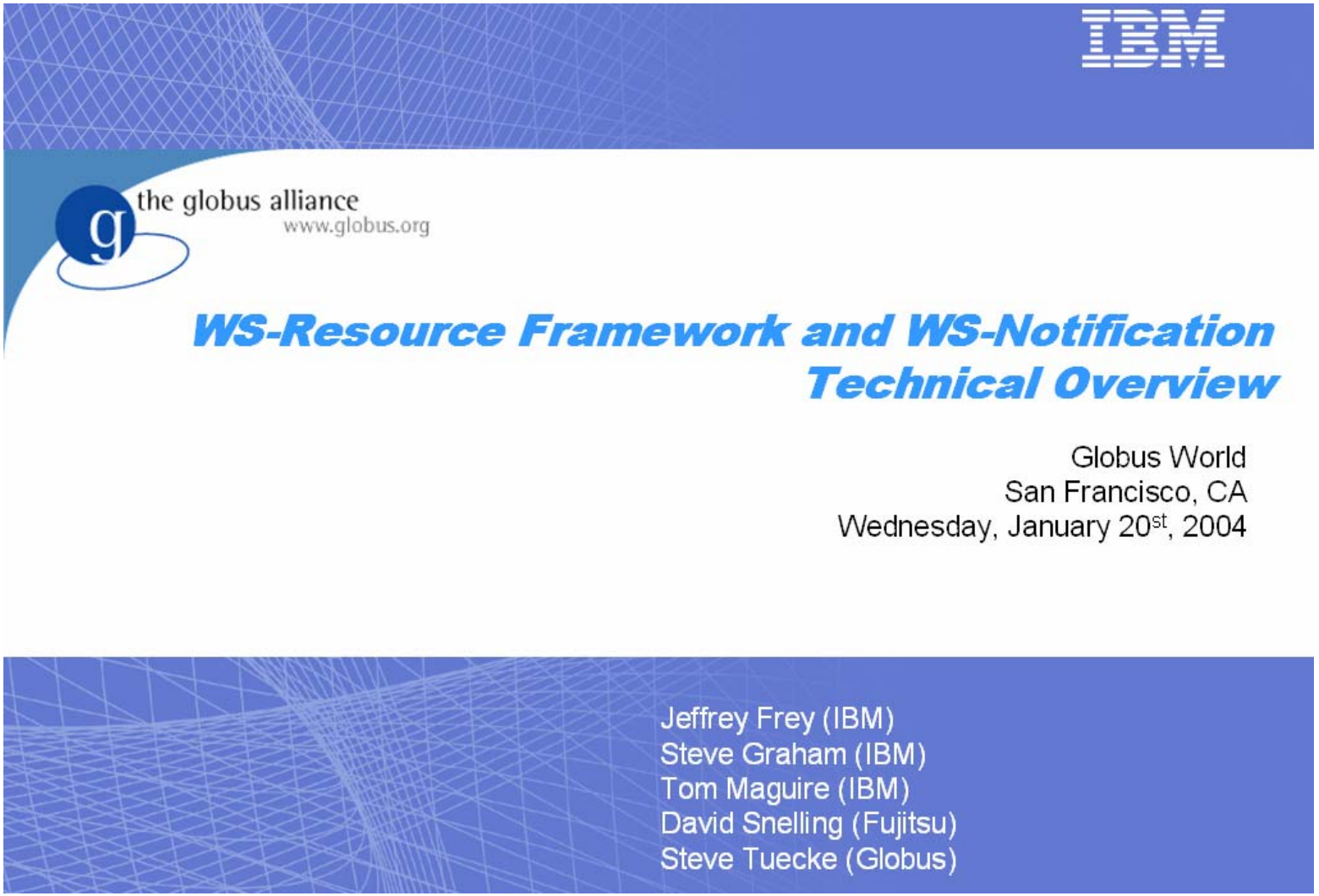
Web Services and Grid computing

Mike Mineter

National e-Science Centre, Edinburgh

Acknowledgements: some slides are from

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



Policy for re-use

- 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!!

Goal of presentation

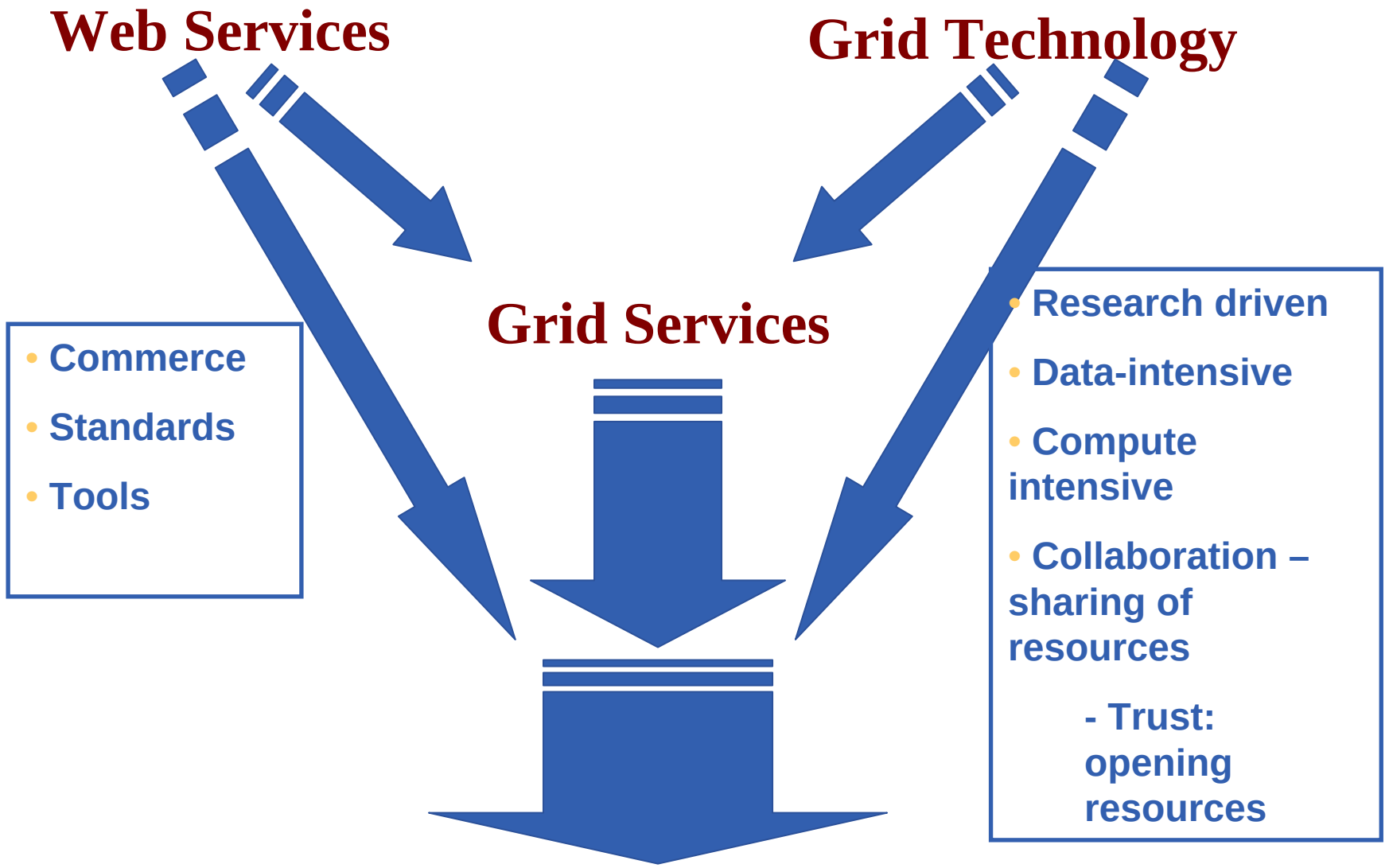
- An orientation to Web Services and to their role in Grid computing
- No prior knowledge assumed

Outline

- **“ 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?!**

Infrastructure for the industrial society: The Forth Bridges

October 2001 View



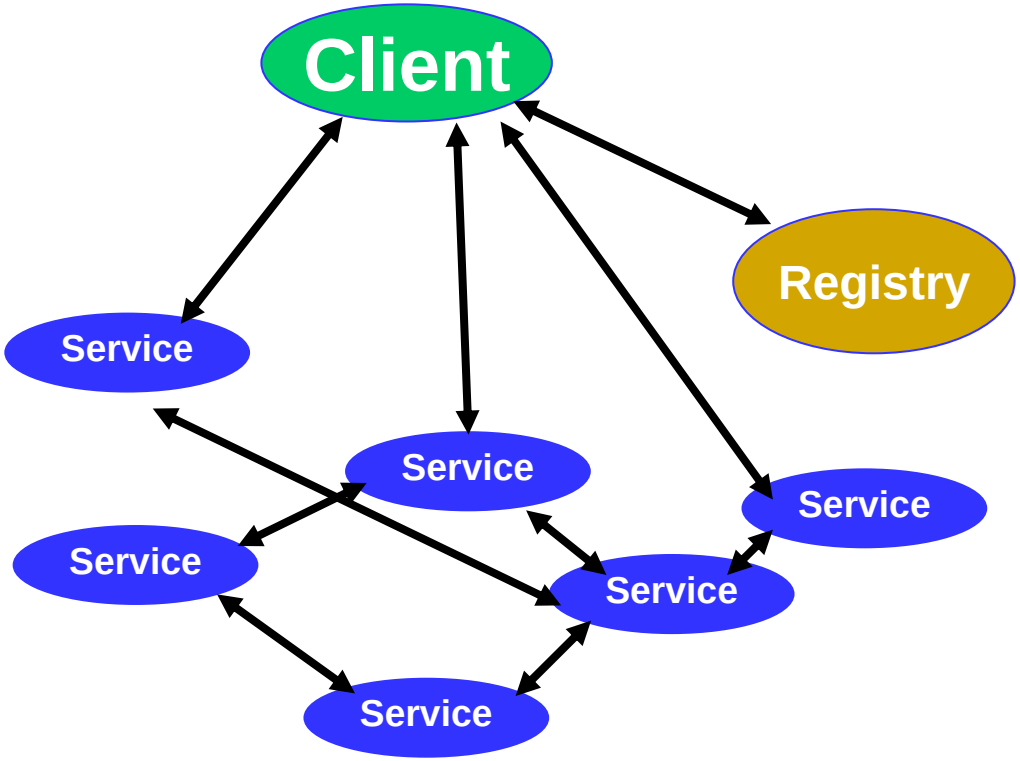
What are "Web Services?"

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

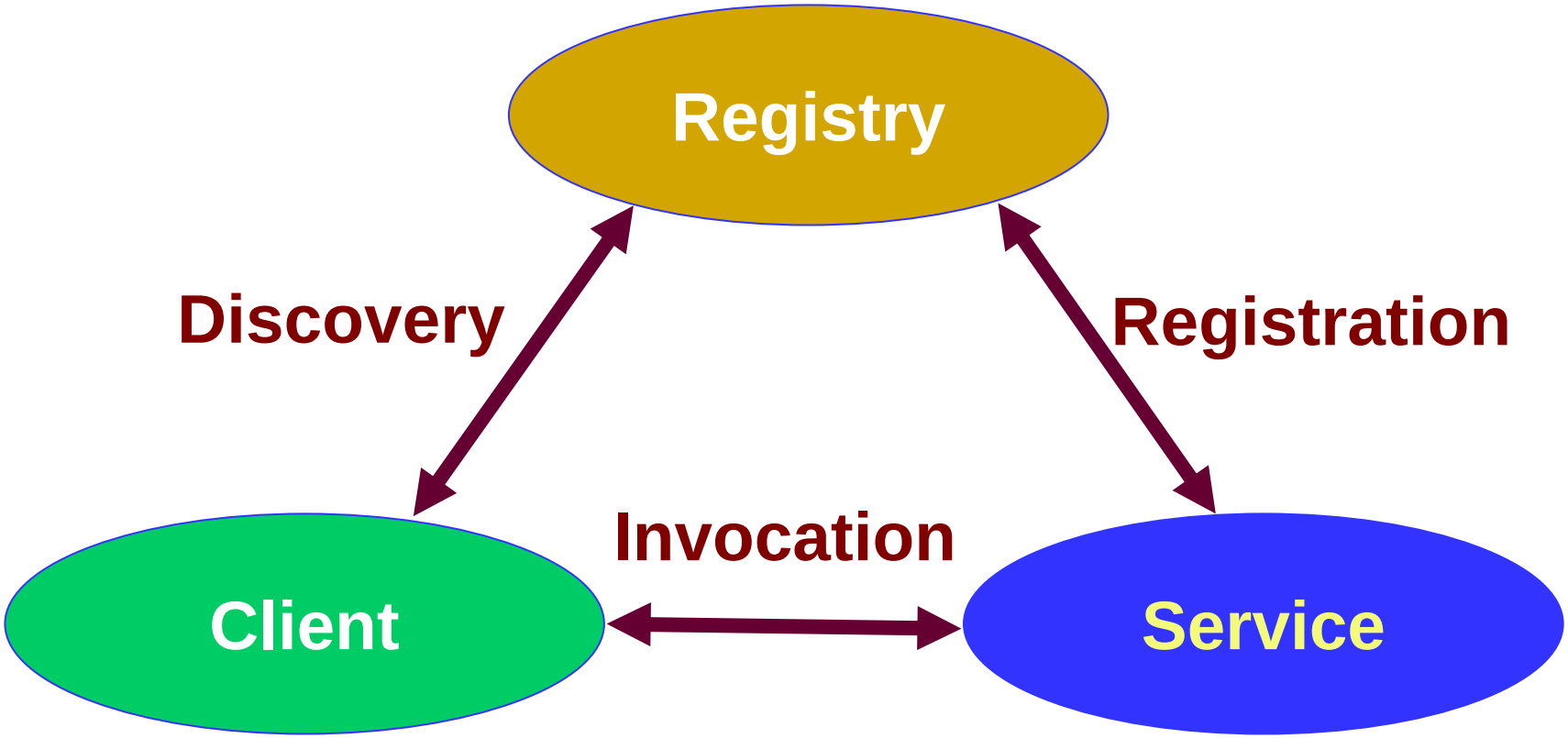
- **Web Services are software components that are..**
  - Accessible across a network
  - Loosely coupled
    - Defined by the messages they receive / send
    - Modular and self-contained
    - So can change service implementation without changing interfaces
  - Interoperable: each service has a description 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

Service orientation – software components that are…

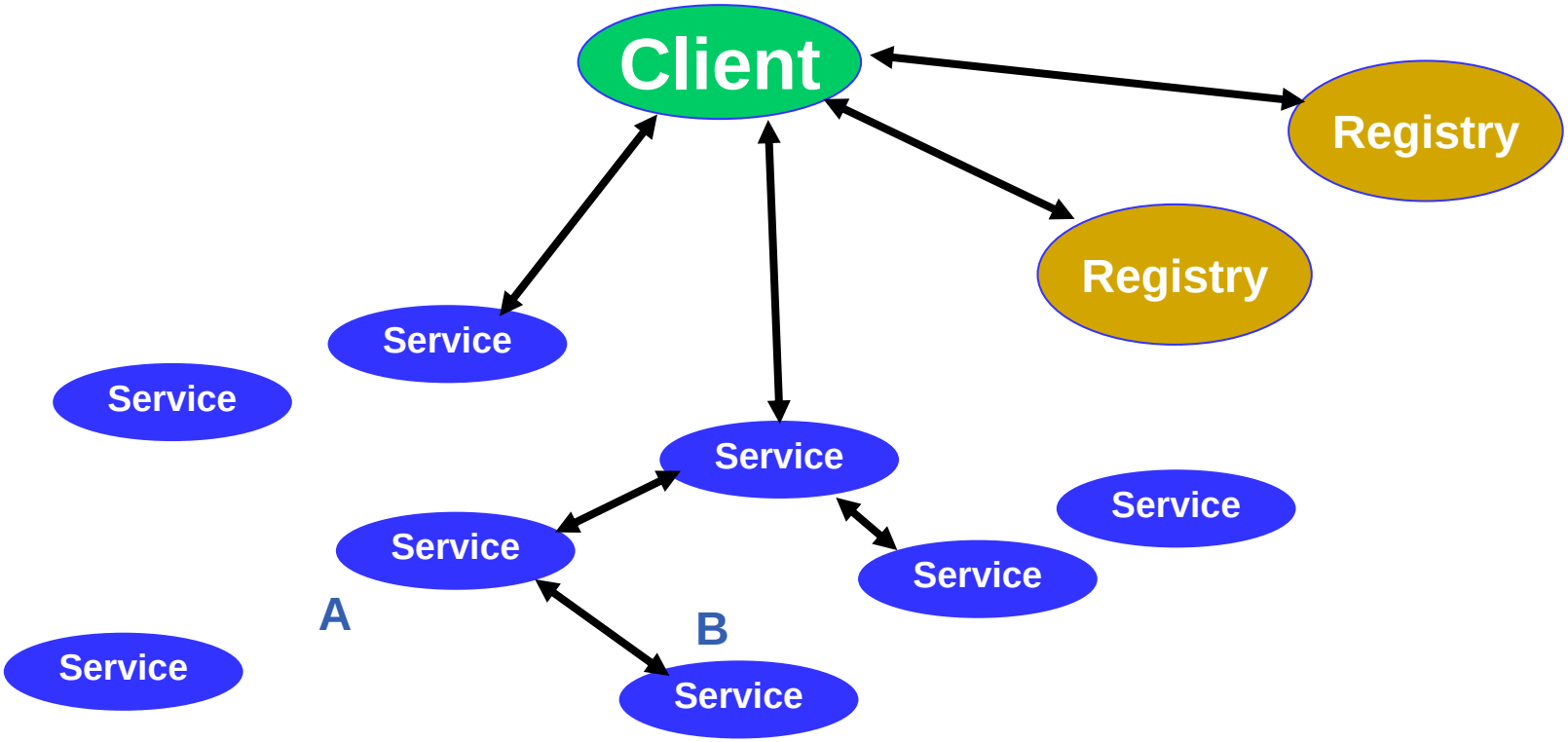
- Accessible across a network
- Loosely coupled, defined by the messages they receive / send
- Interoperable: each service has a description that is accessible and can be used to create software to invoke that service
- Based on standards (for which tools do / could exist)
- Developed in anticipation of new uses



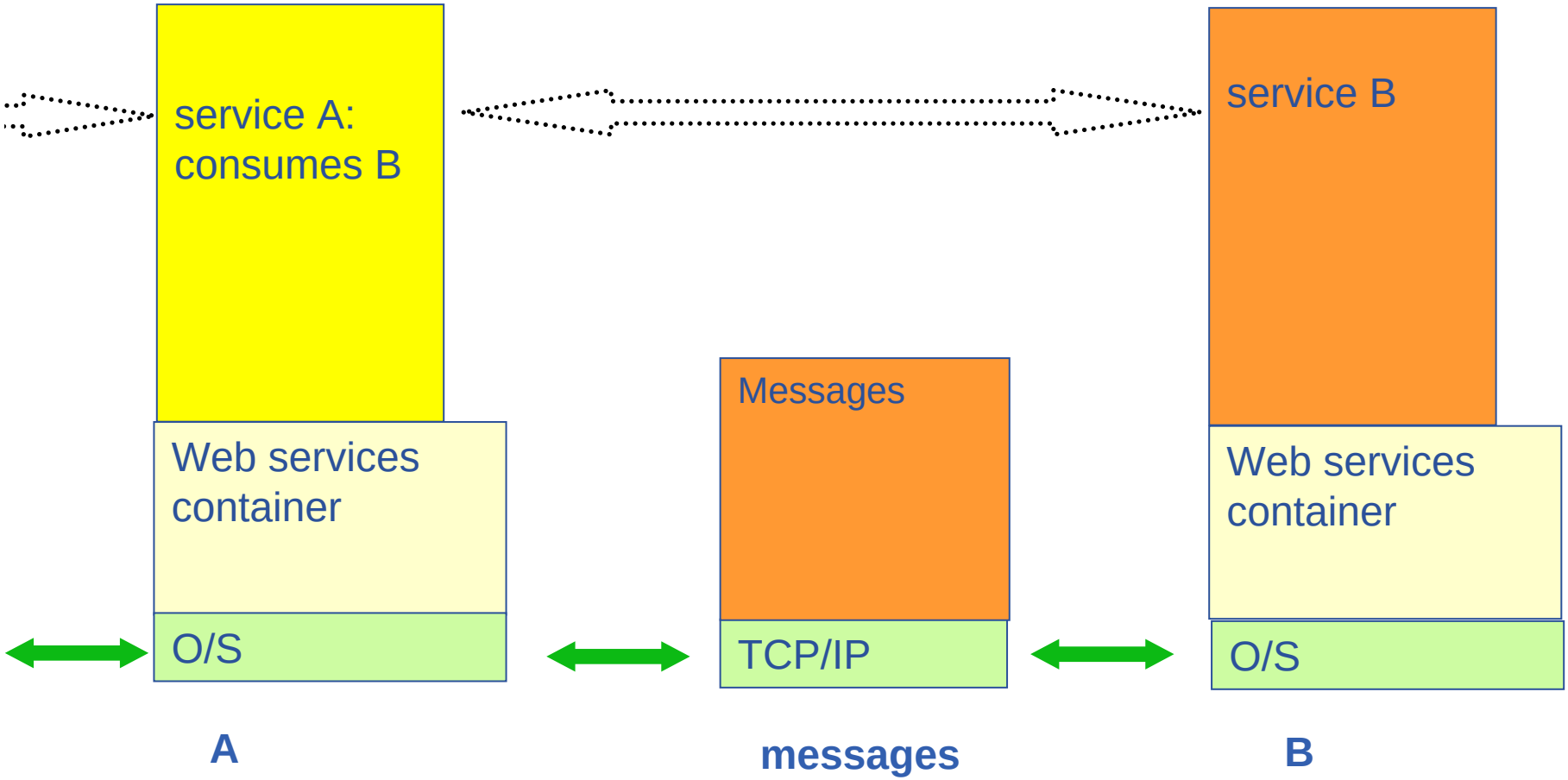
Web Services



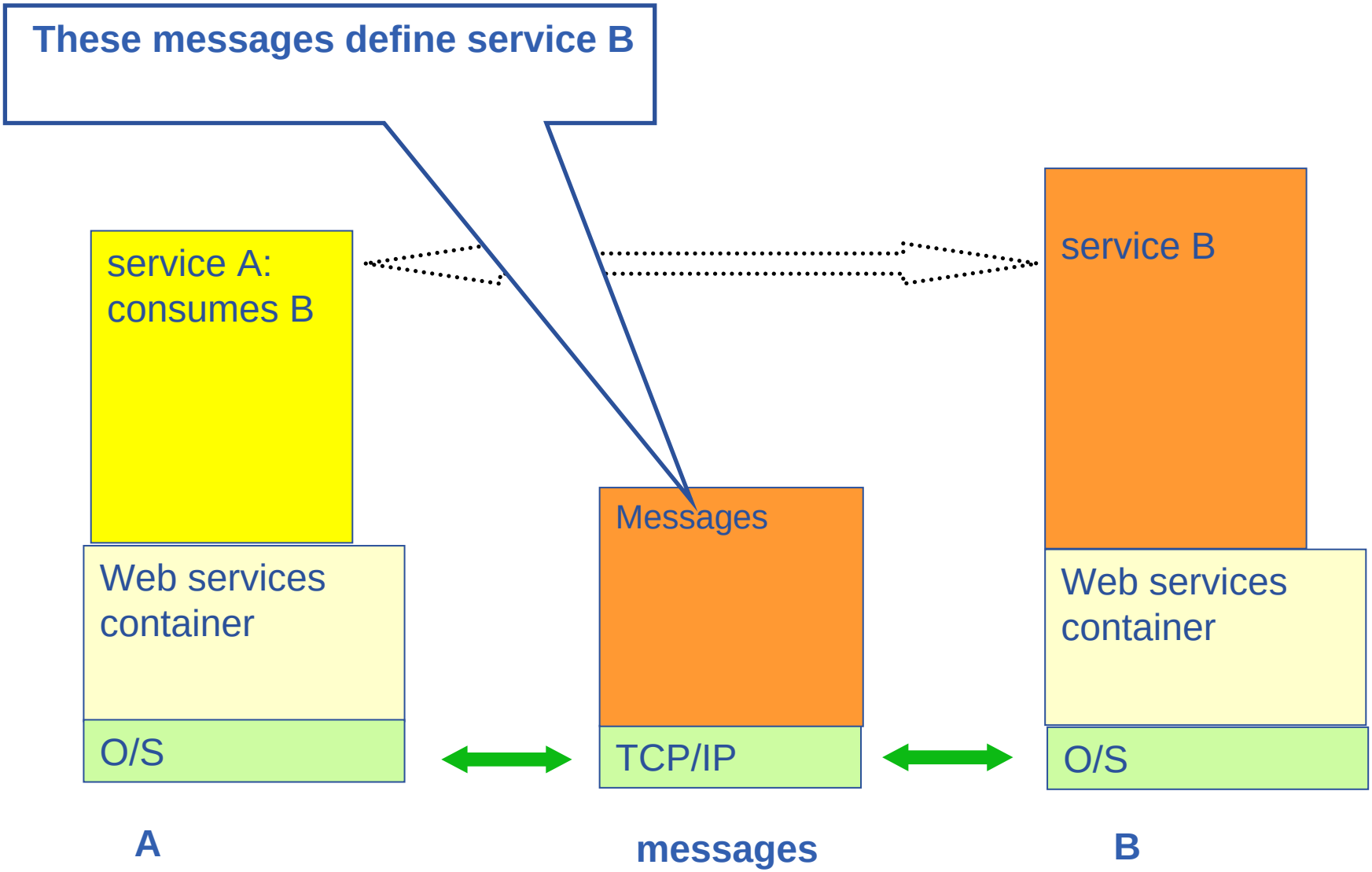
Dynamic composition



Using service B from service A

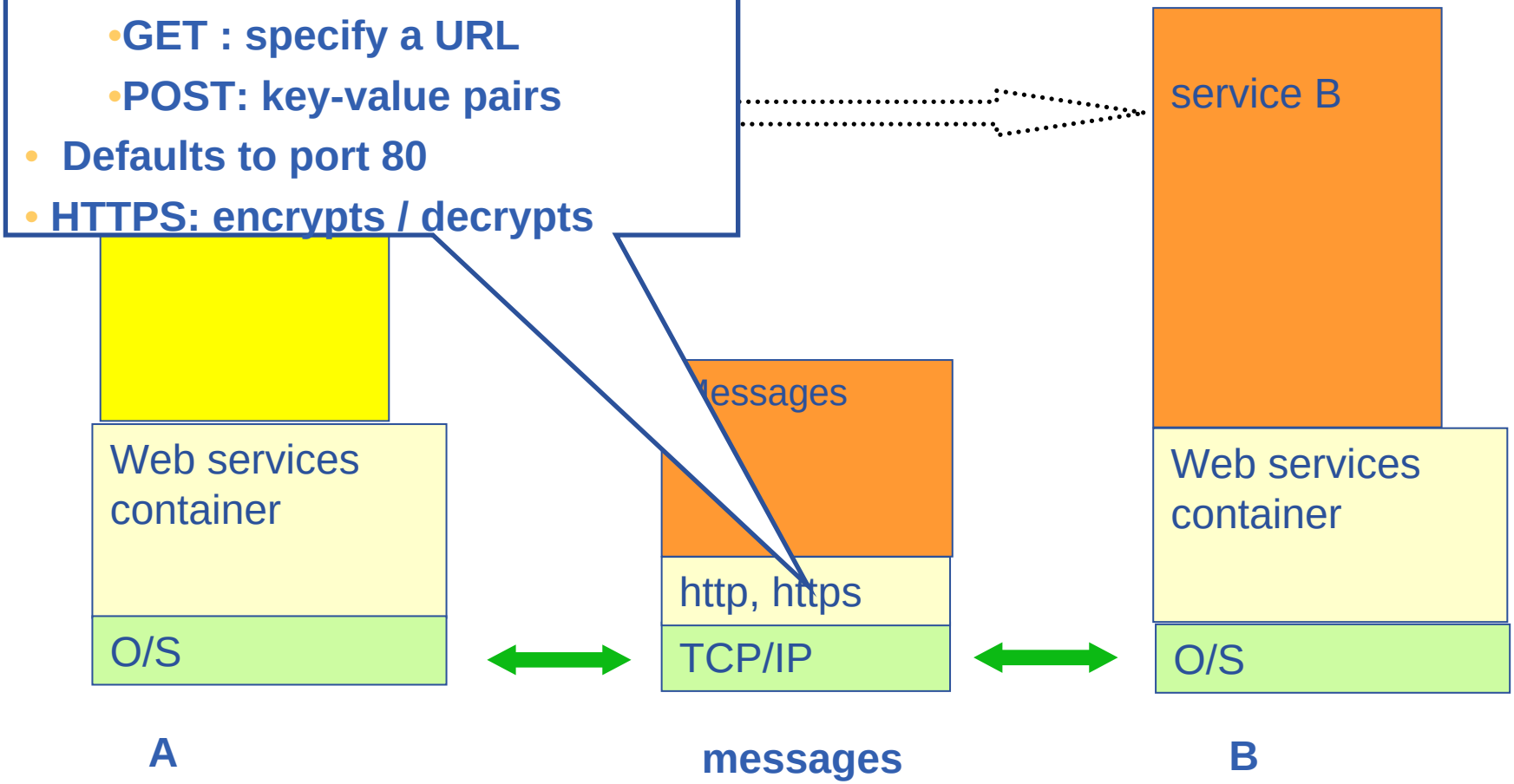


Using service B from service A



HTTP and HTTPS

- 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

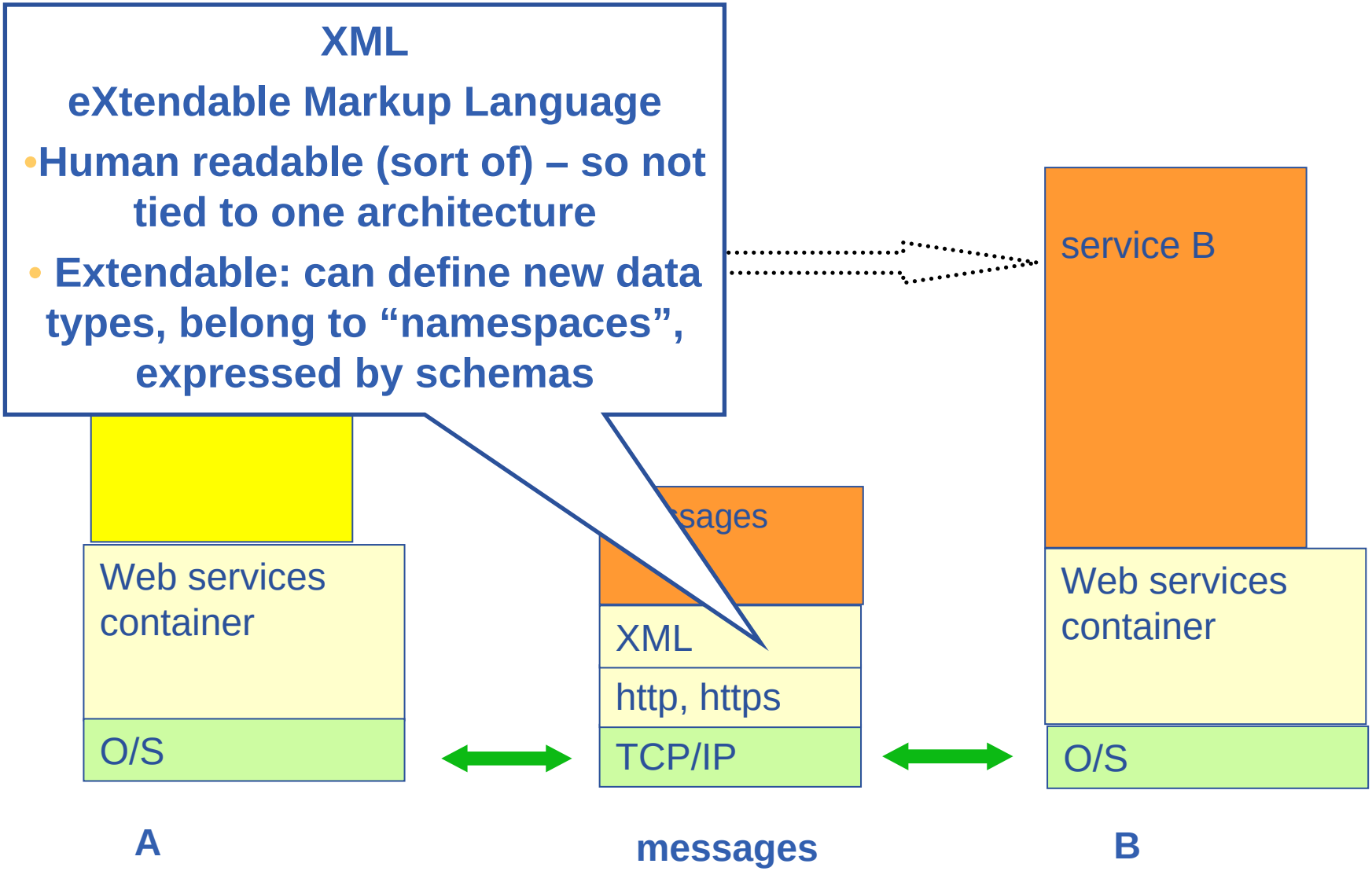


XML – usual basis for messages

XML

eXtendable Markup Language

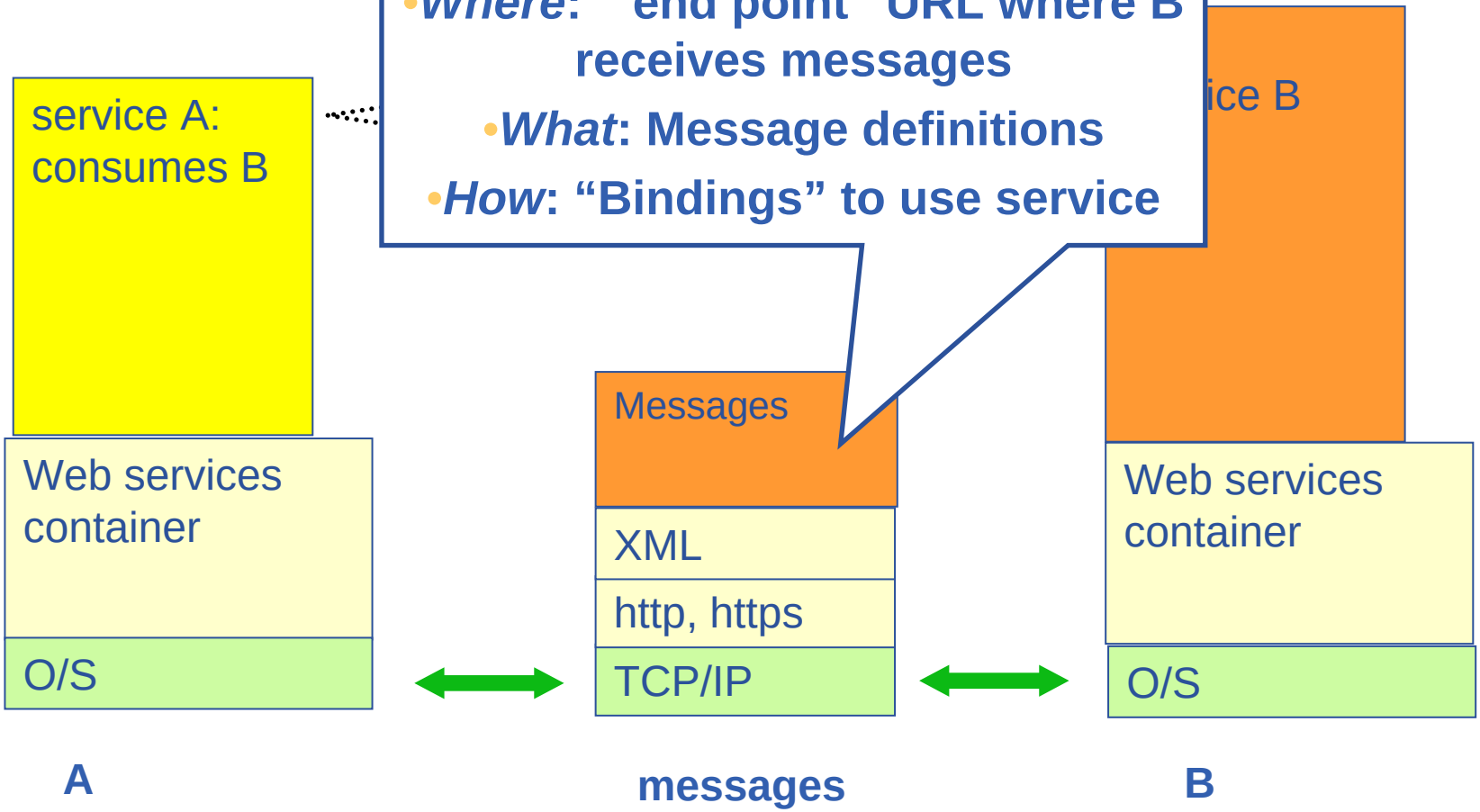
- **Human readable (sort of) – so not tied to one architecture**
- **Extendable: can define new data types, belong to "namespaces", expressed by schemas**



WSDL -1

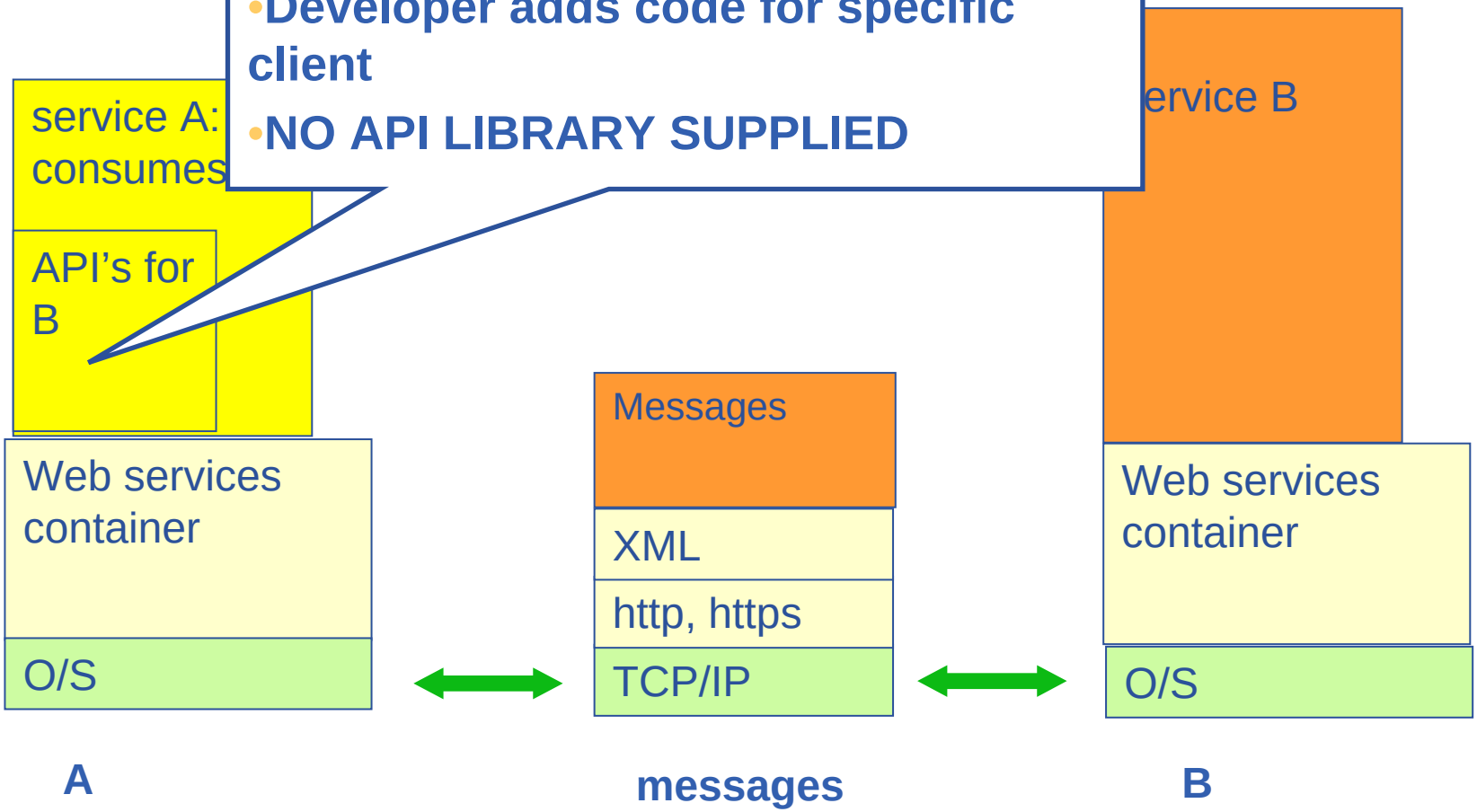
Service B is described by WSDL, "Web Service Description Language". Includes:

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



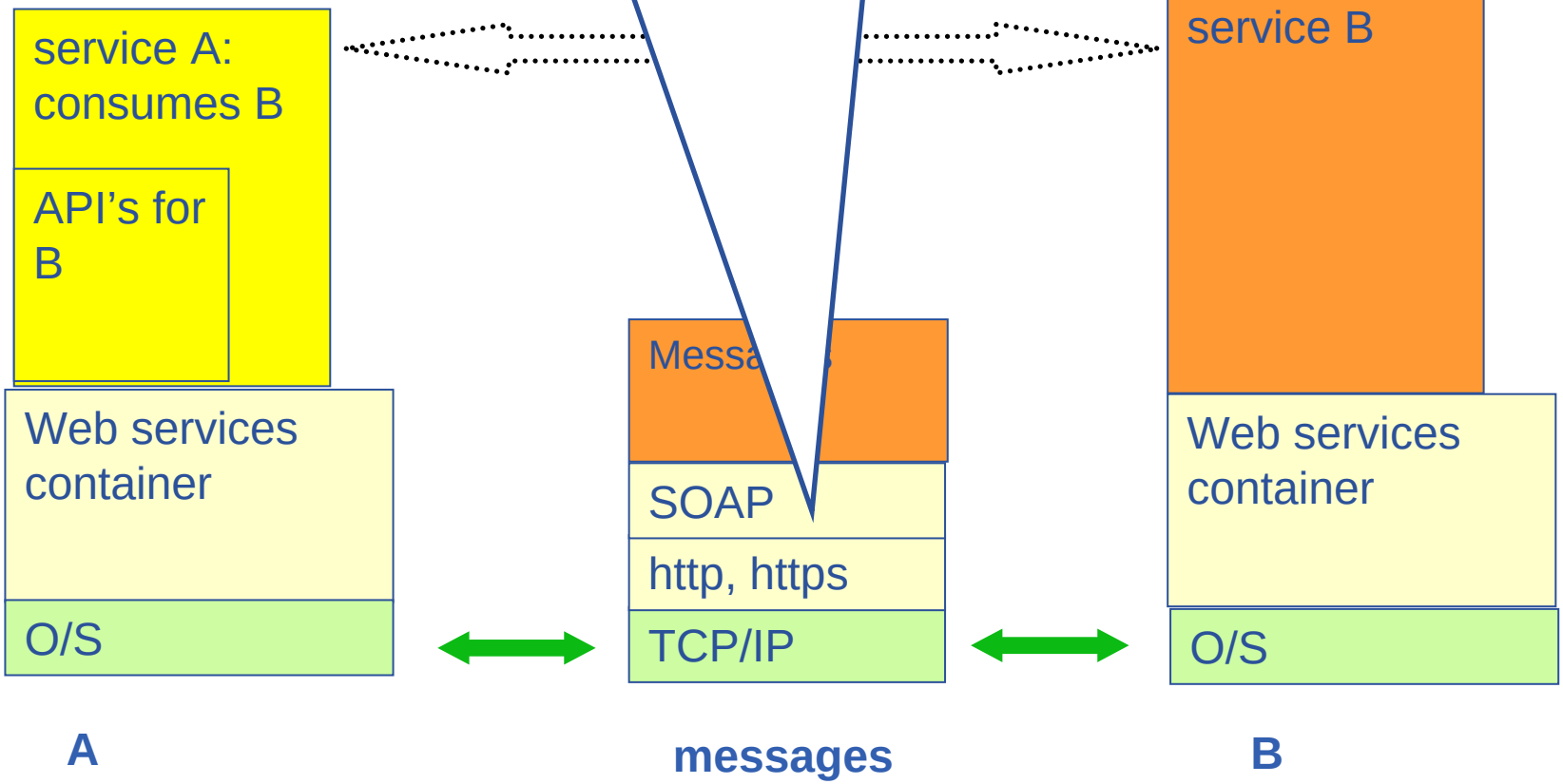
WSDL -2

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

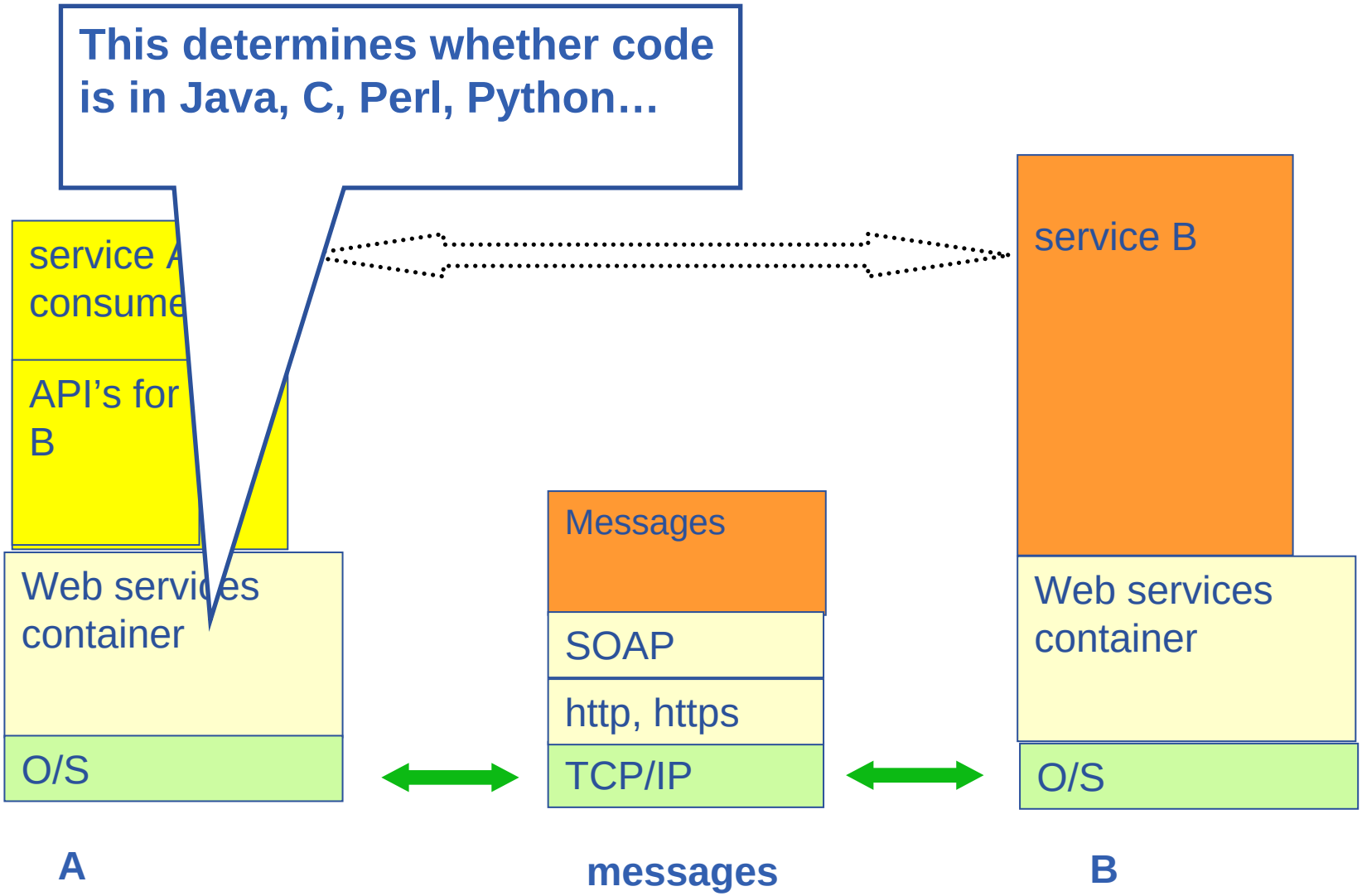


SOAP

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



Code languages



SOAP RPC Request

- **This is a request to a service called `averager` which takes two numbers and returns the average**

```
<SOAP-ENV:Envelope
xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
SOAP-ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <SOAP-ENV:Body>
    <m:getAverage xmlns:m="http://ella:8080/axis/services/averager">
      <in0 xsi:type="xsd:int">5</in0>
      <in1 xsi:type="xsd:int">7</in1>
    </m:getAverage>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

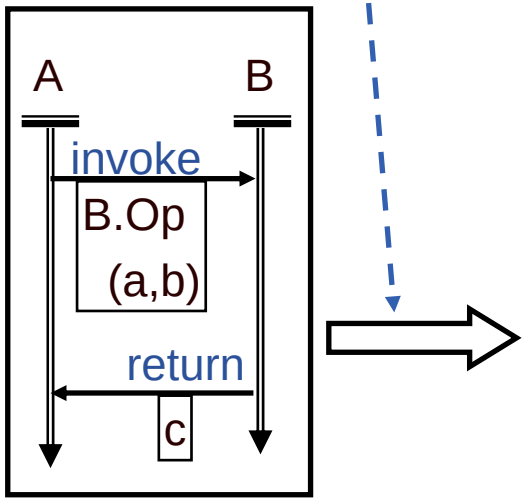
SOAP RPC Response

- **Here is the response from the `averager` service**

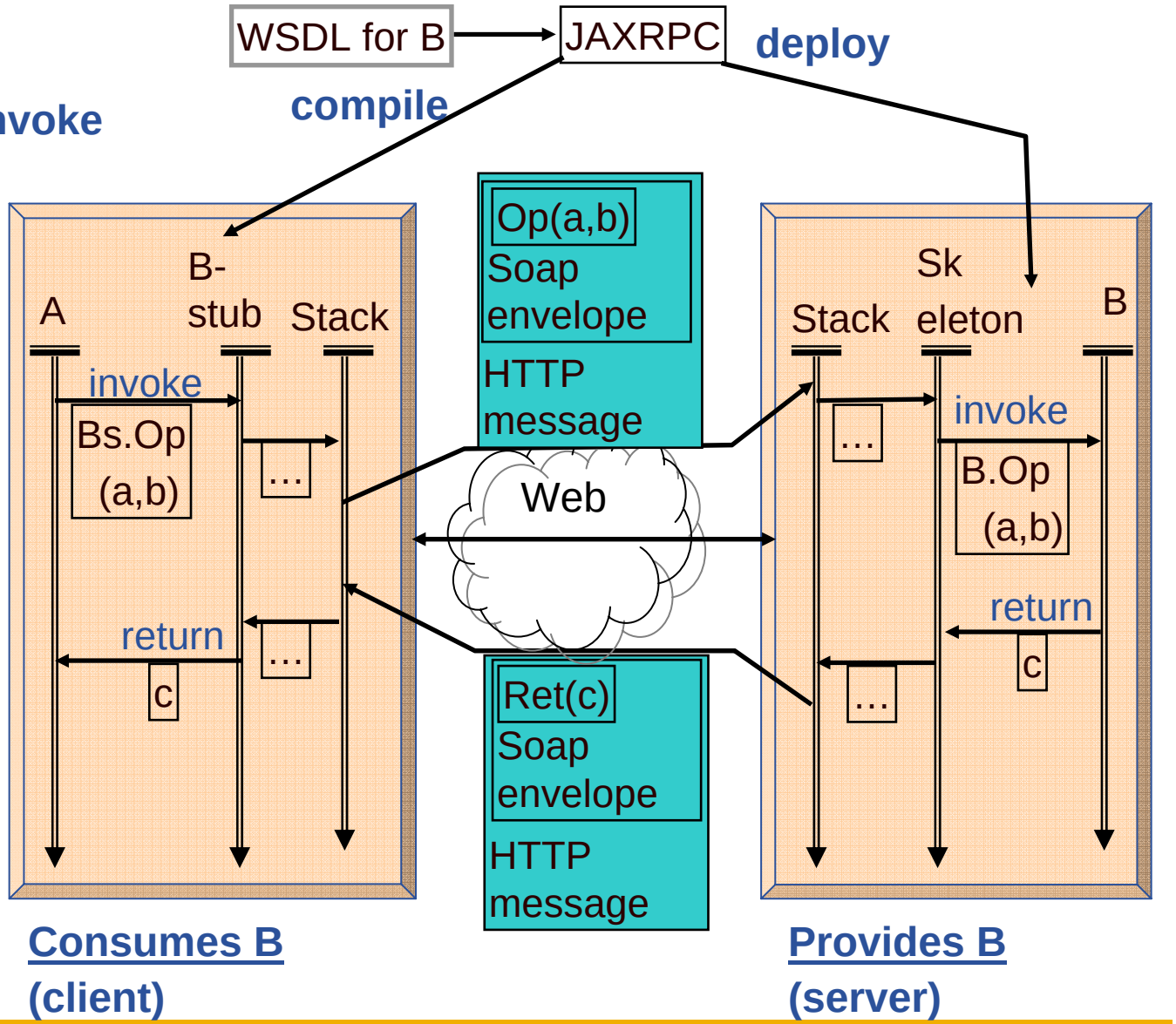
```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <soapenv:Body>
    <ns1:getAverageResponse
soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:ns1="http://ella:8080/axis/services/averager">
      <getAverageReturn xsi:type="xsd:double">
         6.0
      </getAverageReturn>
    </ns1:getAverageResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

(JAVA) Web Services

Insert the web into the invoke and into the return



Java program with 2 classes



Consumes B (client)

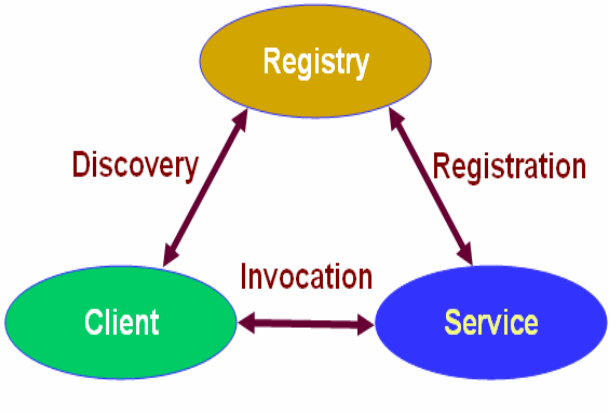
Provides B (server)

WS-I core of Web Services

- 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



Outline

- “ 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 ?

Grids need….

- **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!!!!!**

WS & Grid Goals

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

WS & Grid Usage

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

Operational status

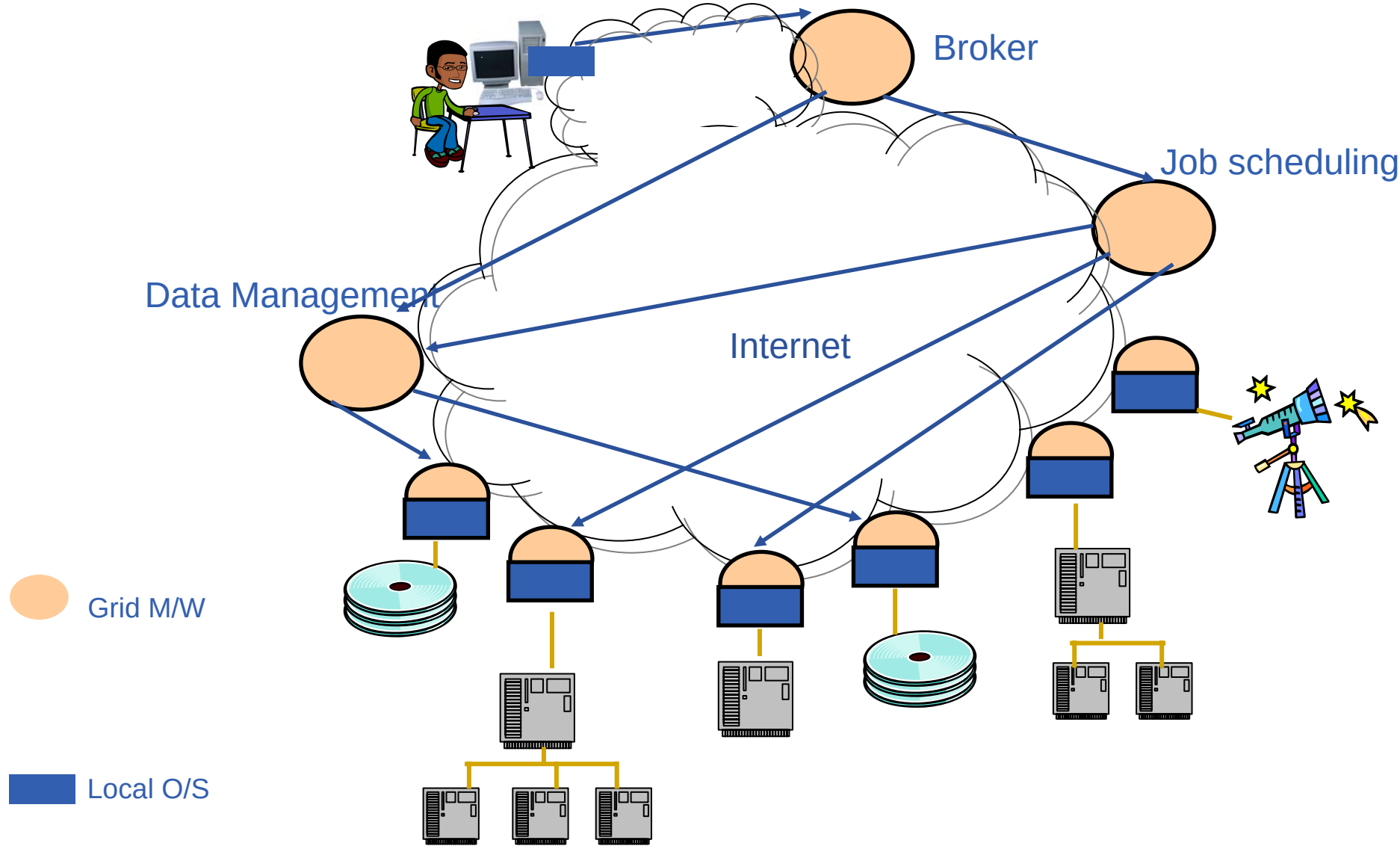
Web Services

- **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

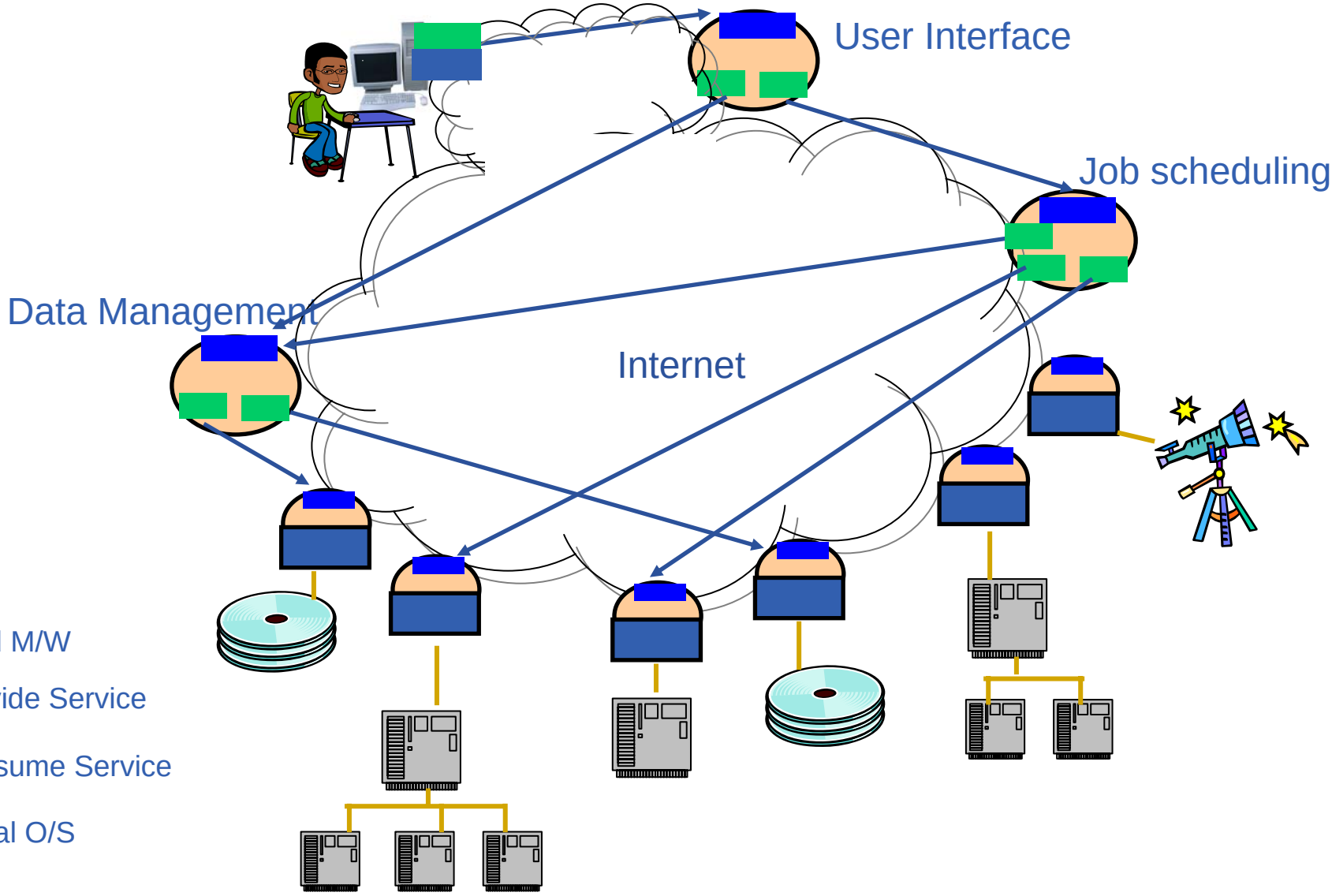
Grids

- **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**

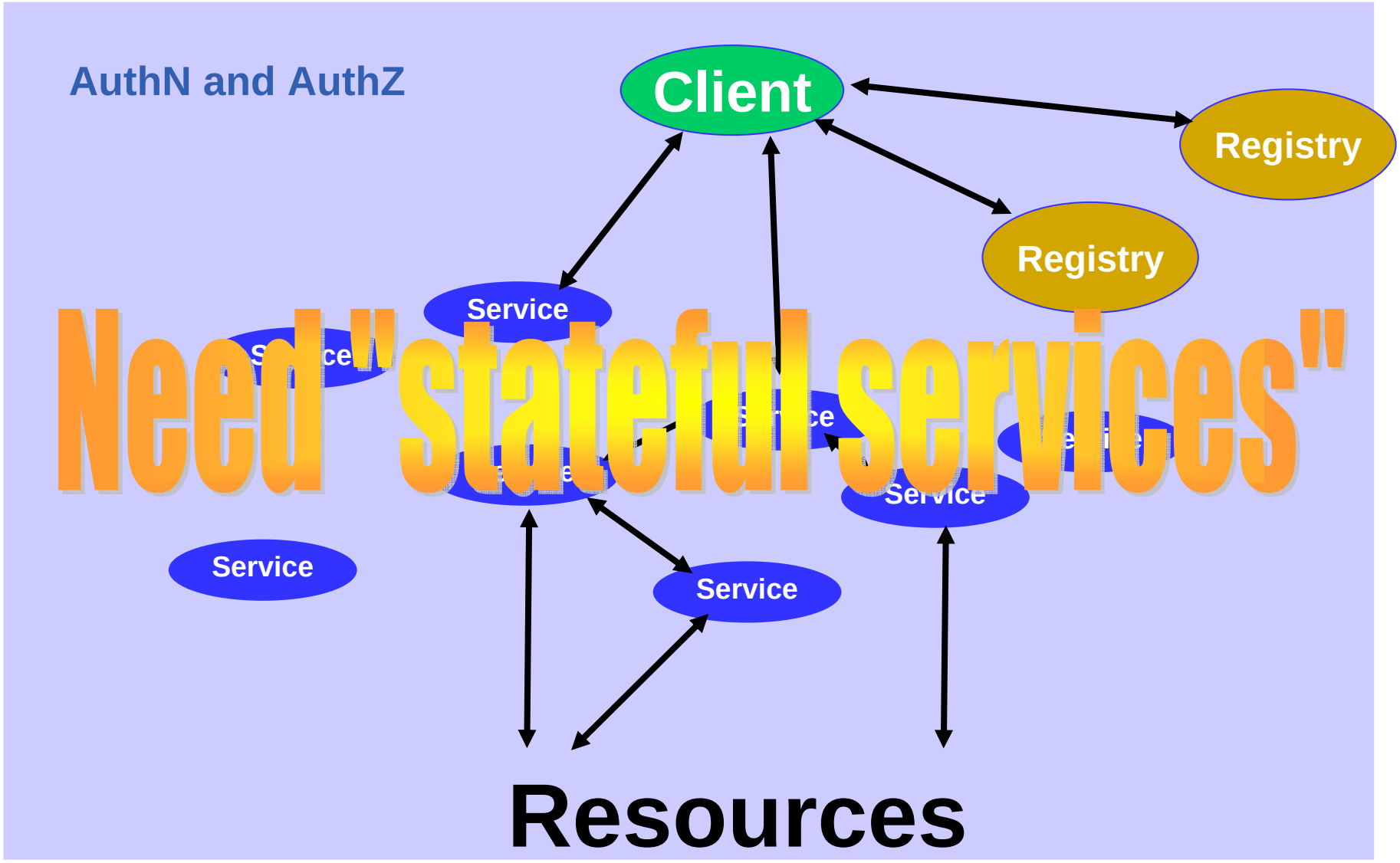
Re-Package Grid Middleware: from..



Re-Package Grid Middleware: to…



Grids – and resources



A bit of history

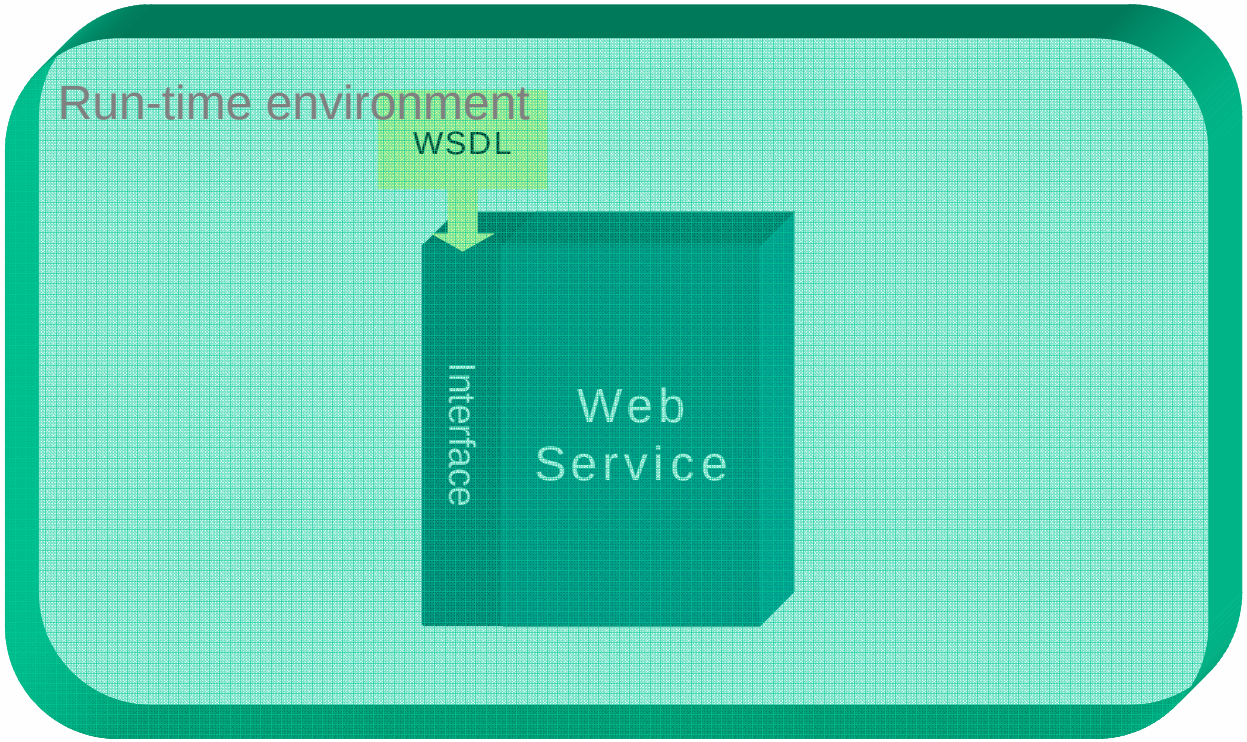
- **“Open grid services architecture” OGSA– proposed in 2001**
- **Open Grid Services Infrastructure**
  - Globus Toolkit 3 resulted
  - Specified in 2003
- **Then in January 2004**
  - OGSI to be replaced by emerging WS-RF (Web Services Resource Framework)
- **NOTE:**
  - OGSA still under development (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**

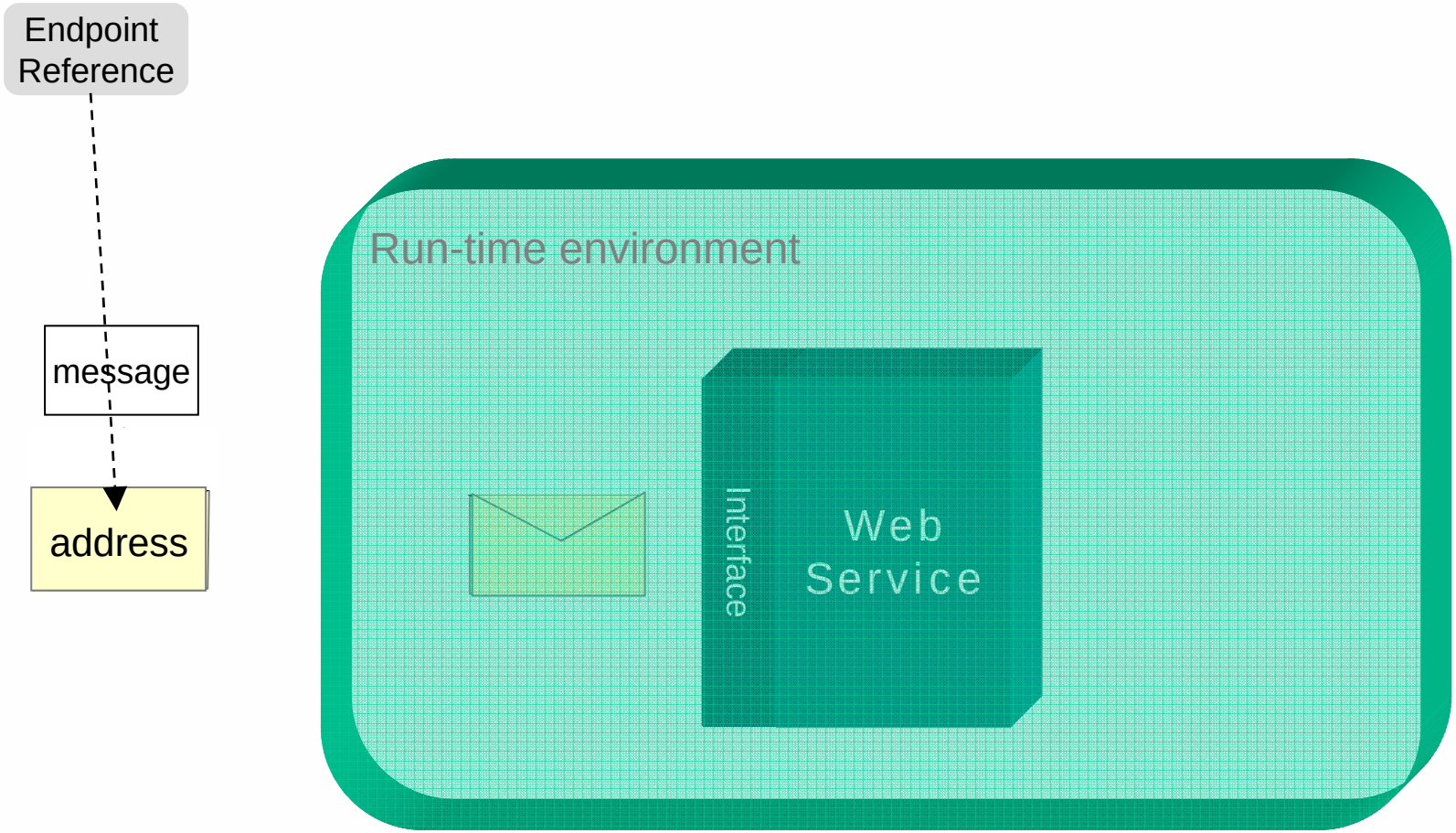
The WS-Resource framework model

Web Service



The WS-Resource framework model

Invoking a Web Service



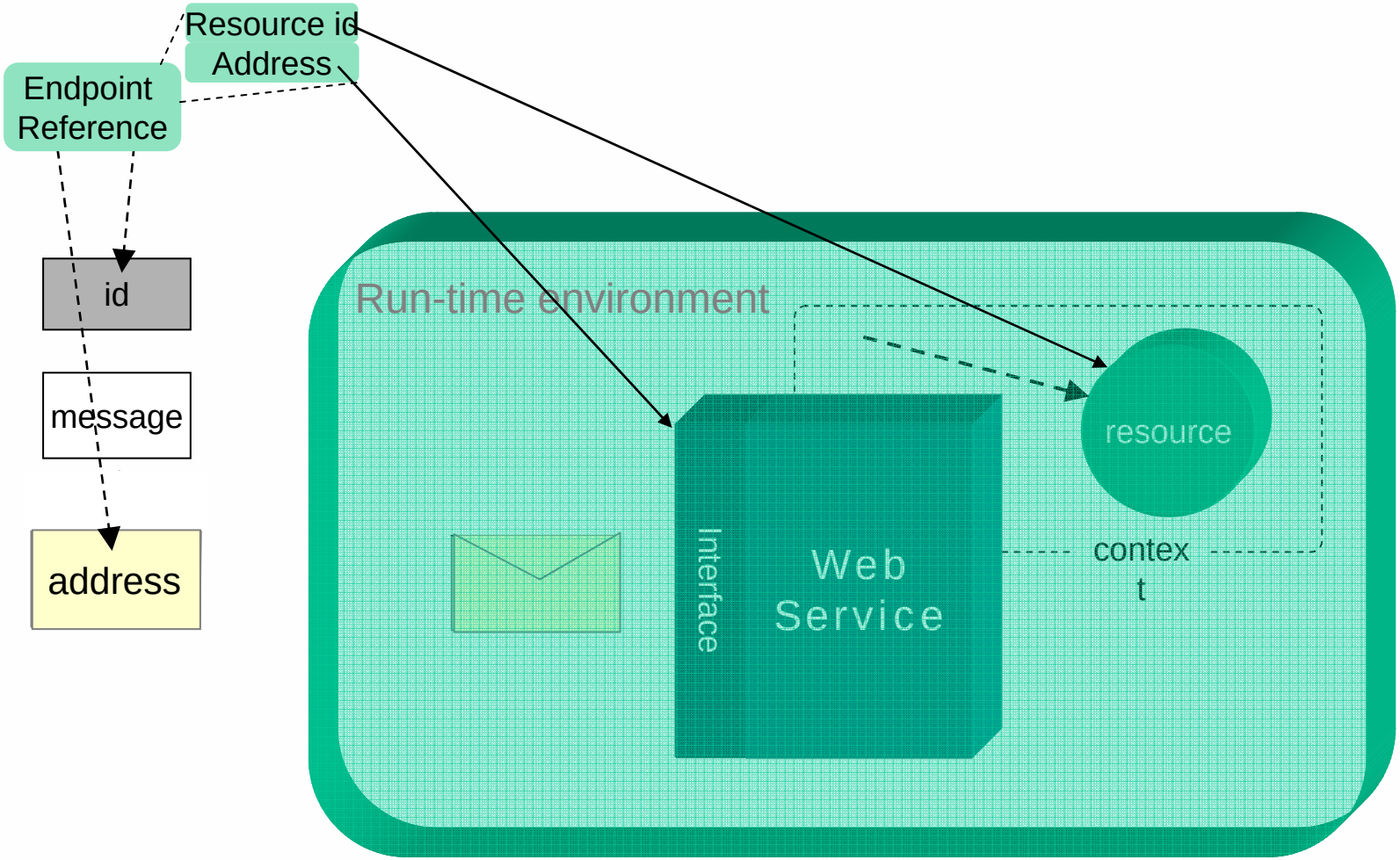
The WS-Resource framework model

- **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)**

  - **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**

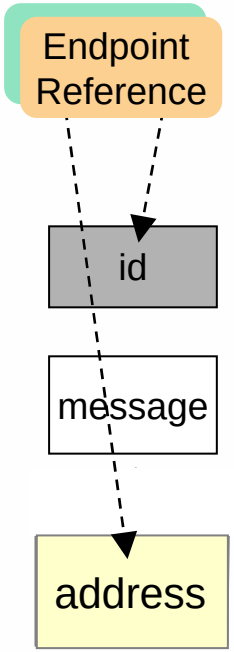
The WS-Resource framework model

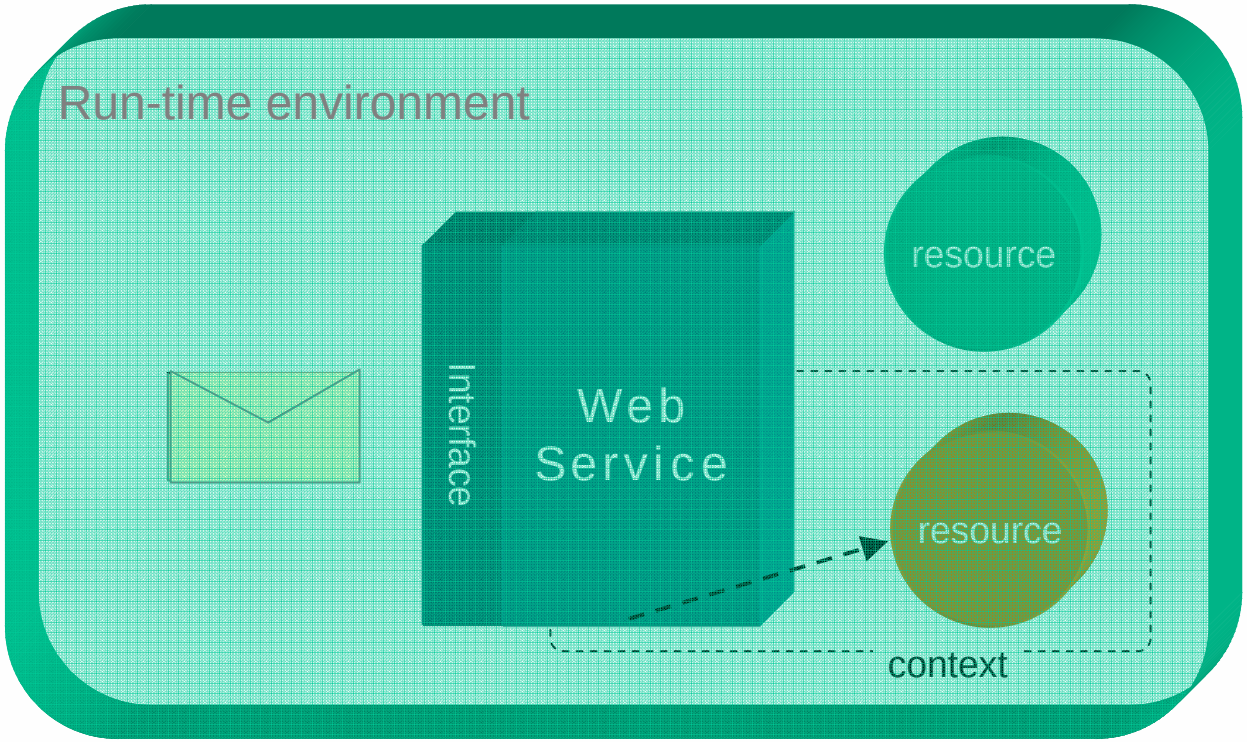
Using a Web service to access a WS-Resource



The WS-Resource framework model

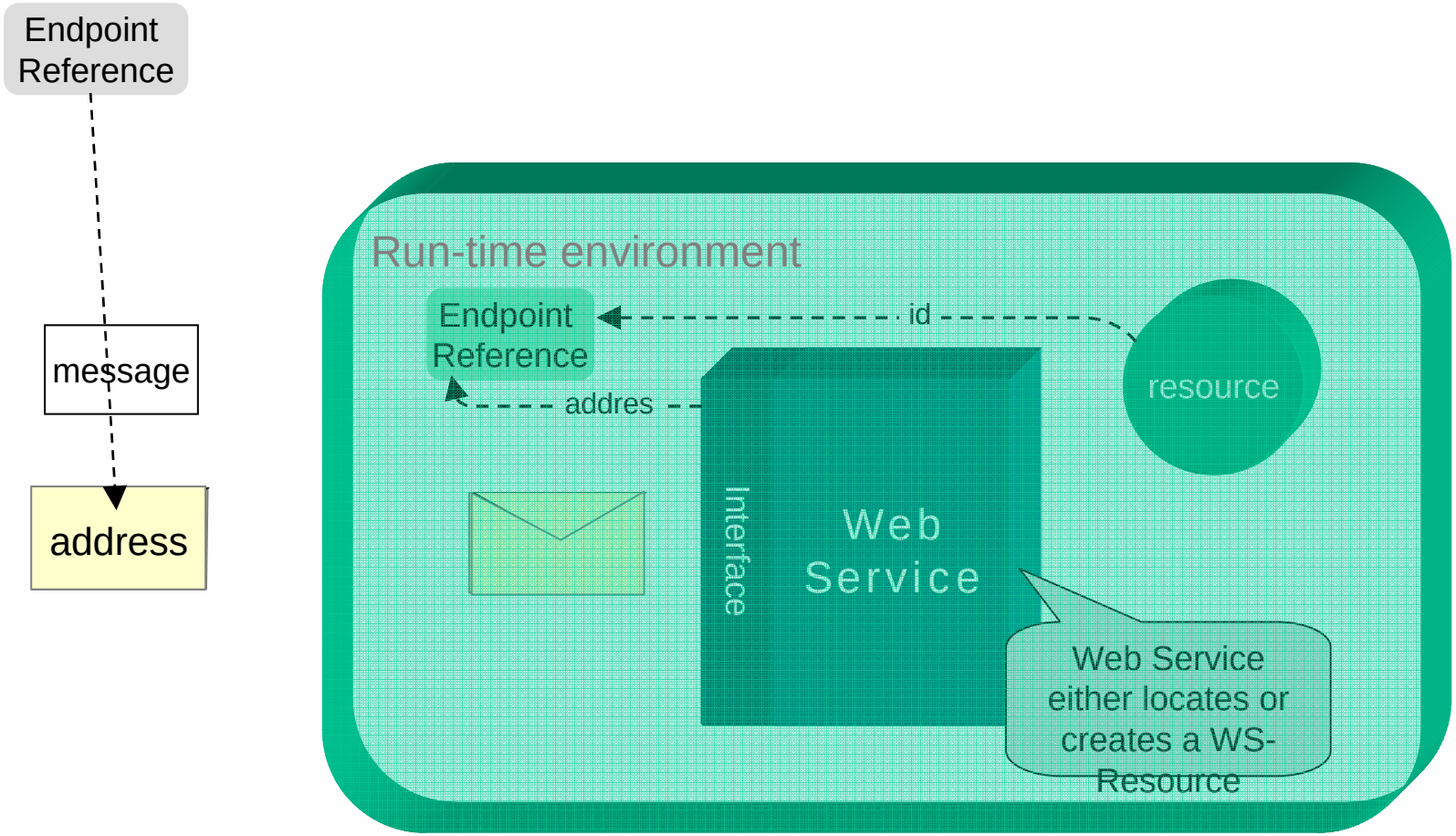
Using a Web service to access a WS-Resource





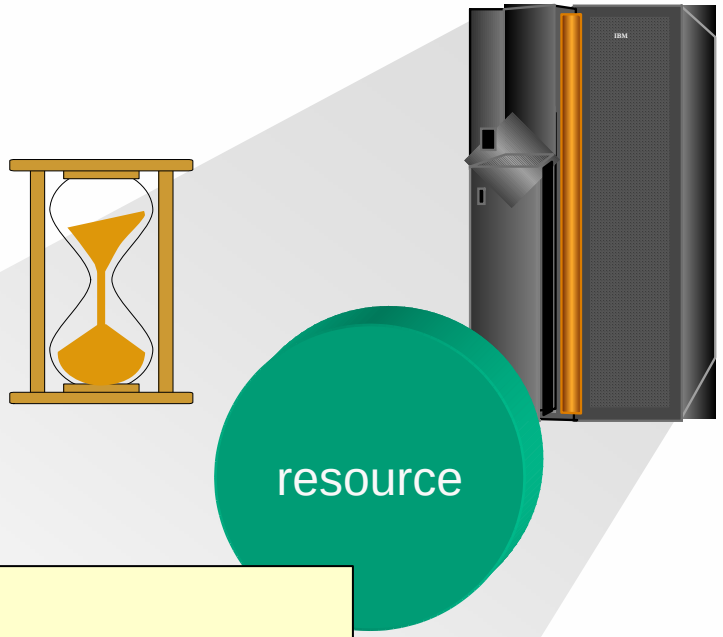
The WS-Resource framework model

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**

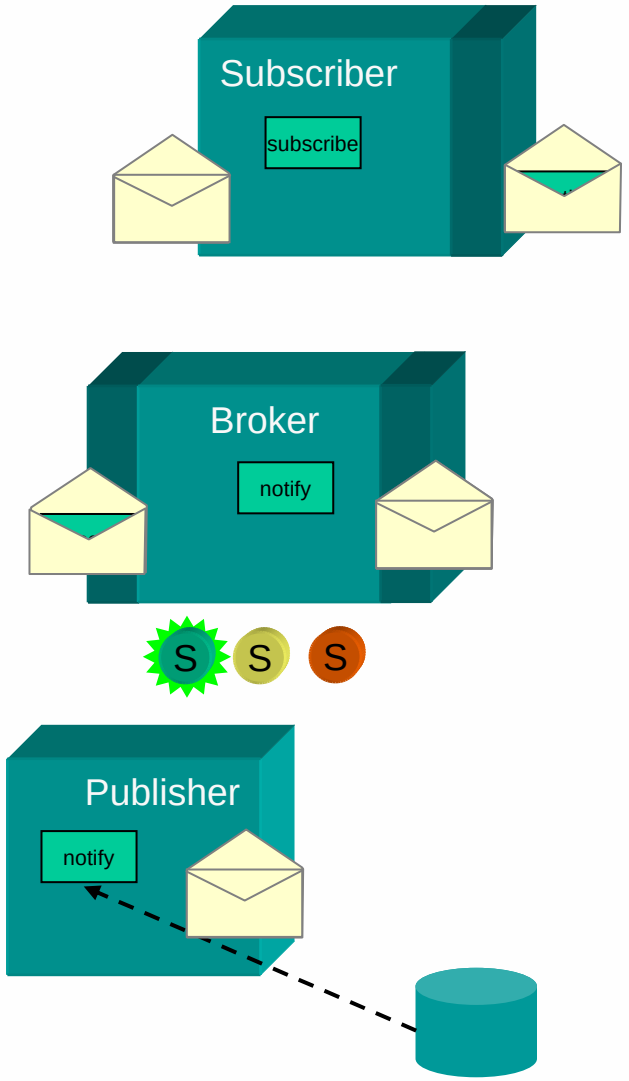



```
<ProcessorProperties>
    <ProcID>5A34C1DE03</ProcID>
    <ProcArchitecture>Power6.2</ProcArchitecture>
    <ProcSpeedMIPS>400</ProcSpeed>
    <ProcCacheMB>256<ProcCache>
    <ProcRunning>1</ProcRunning>

</ProcessorProperties>
```

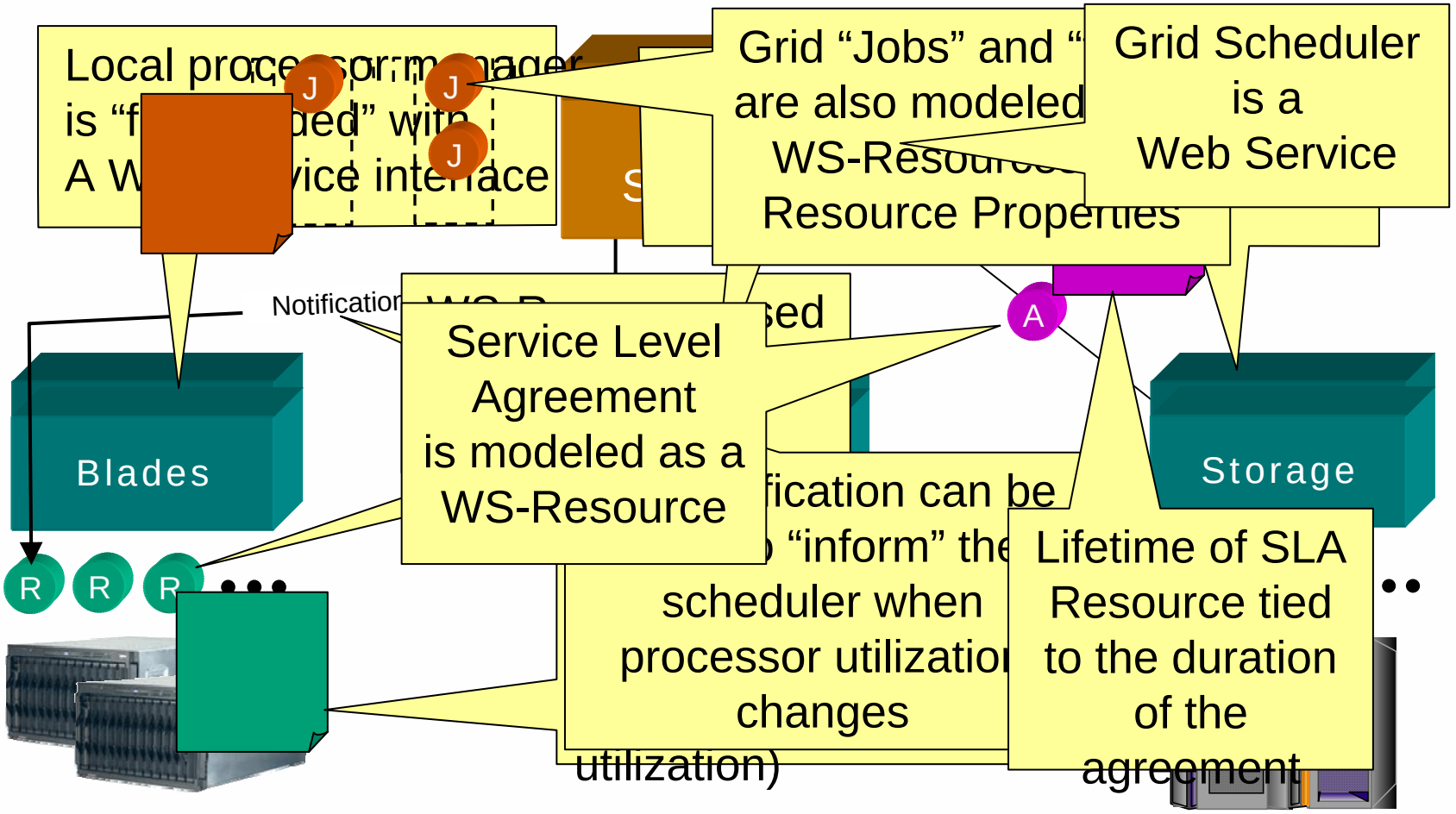
WS-Notification

- **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**



Bringing it All Together

Scenario: Resource management & scheduling



Stateful Resources

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

Component Standards

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

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

Where are we now?!

- **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/
- **Production grids are based on de-facto standards at present**
  - Inevitably!
  - Globus Toolkit 2 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**

Recent progress!

- **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".

GT4-view of OGSA and WSRF -1

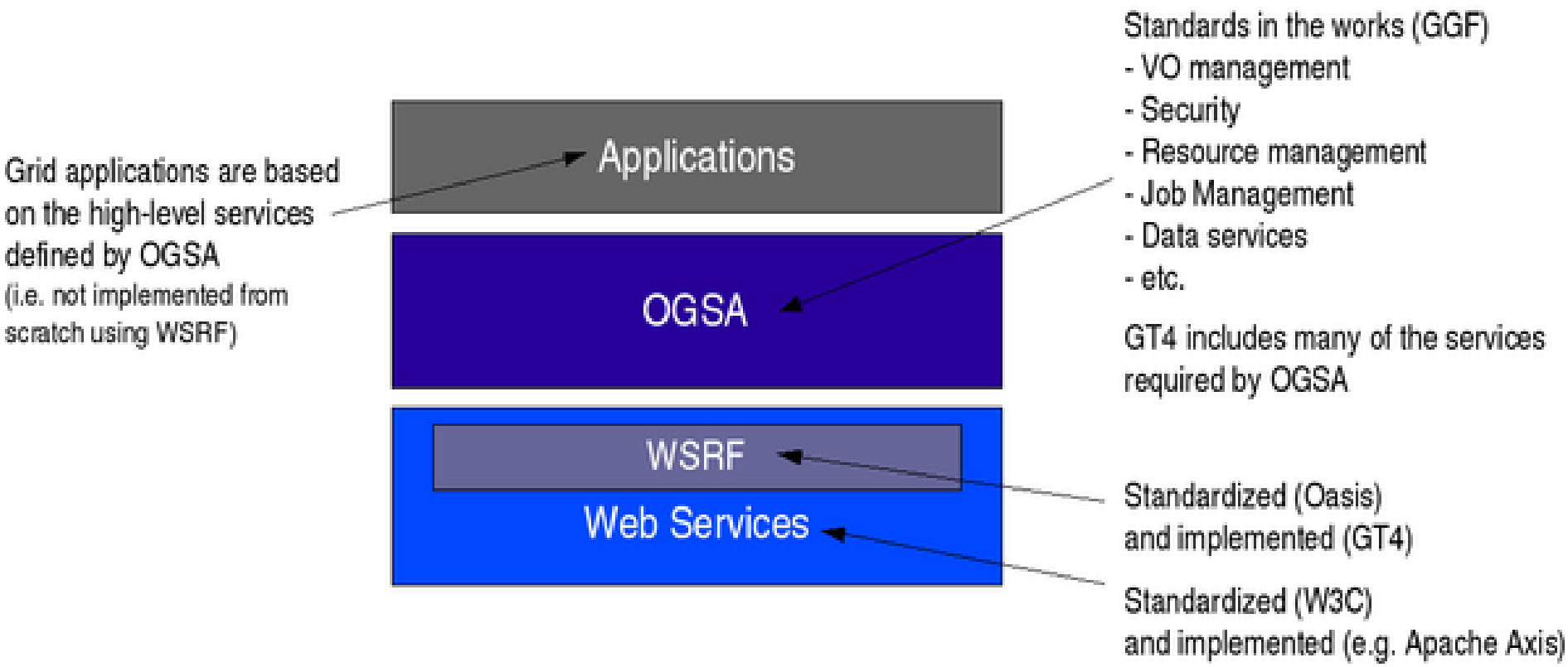


Diagram from Globus Alliance

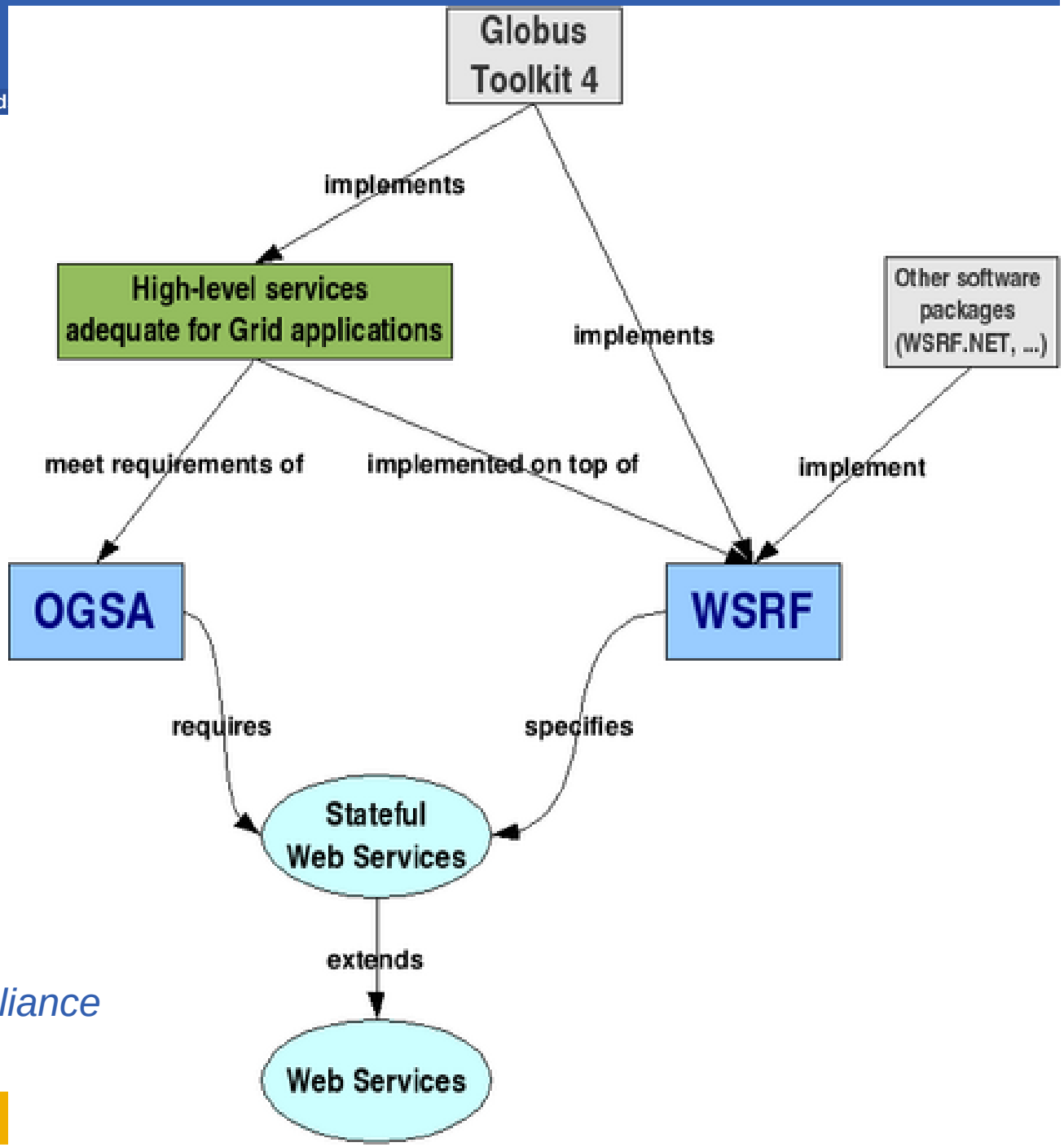


Diagram from Globus Alliance

GT4 and VDT

- **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.”**

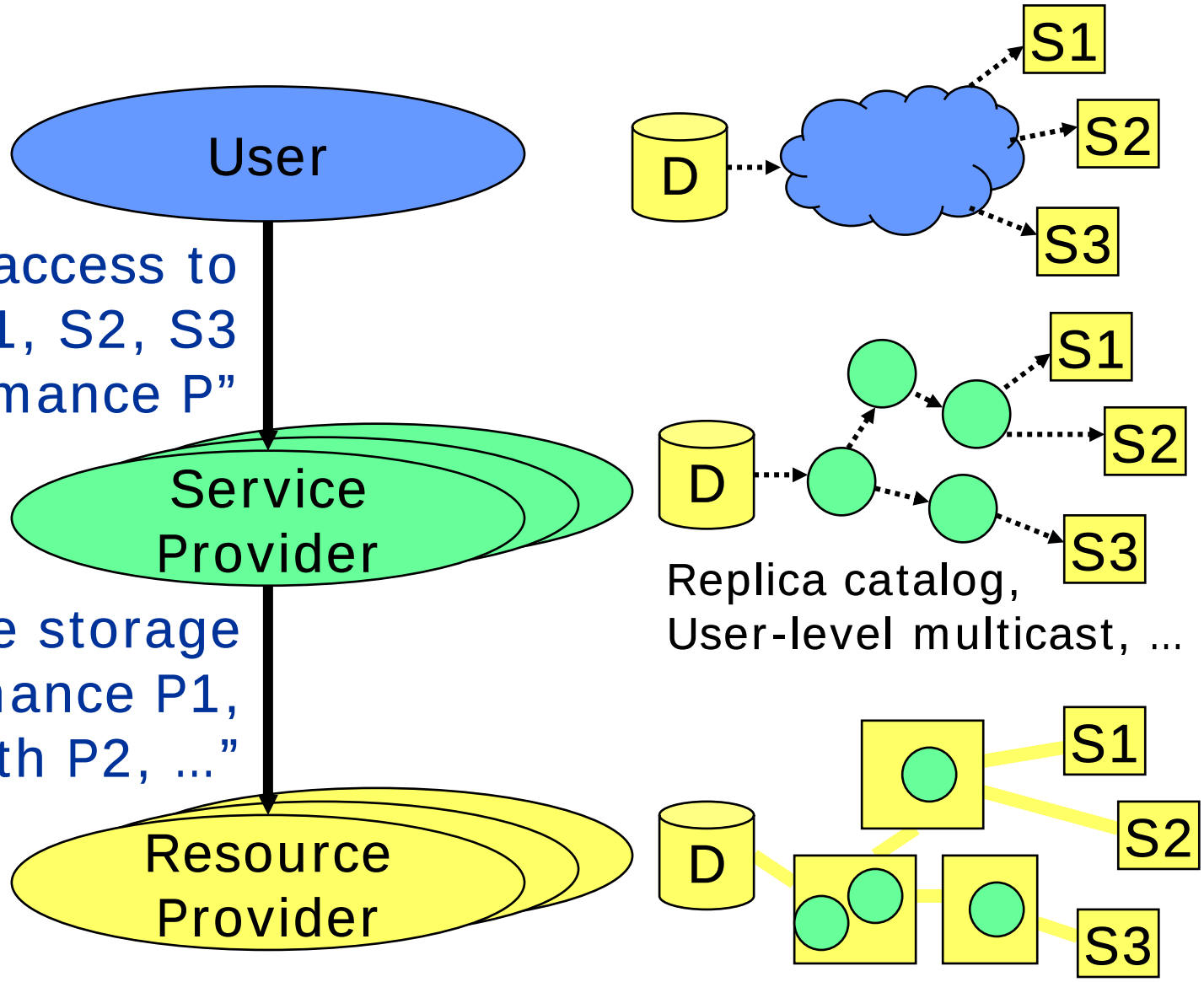
http://vdt.cs.wisc.edu/globus_3.2_vs_4.0.html

Outline

- " 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?!**

the globus alliance
www.globus.org

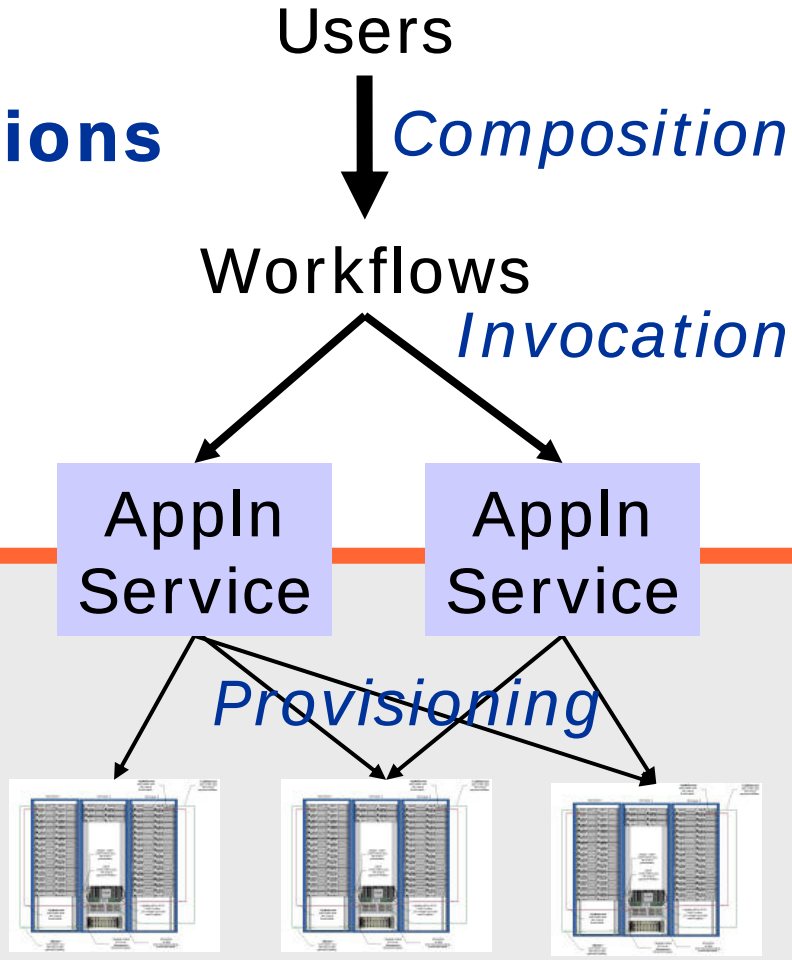
Decomposition Enables Separation of **Concerns** & **Roles**



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



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

Service-oriented research??

- **“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**

Further reading

- 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/

Summary

- 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 WS-I to manage stateful resources are in WS-RF framework
- Initial implementation based on WS-RF and OGSA is in Globus Toolkit 4
- New horizons opening for Service-orientation in research!