



Web Services, CORBA and other Middleware

Dr. Seán Baker
IONA Technologies

Web Services For The Integrated Enterprise, OMG Workshop, Munich Feb 2003

Overview

- There are a number of different types of middleware
 - So what does **Web Services** offer?

Middleware

- Middleware enables integration, but there are multiple – competing – choices:
 - CORBA
 - J2EE
 - .NET
 - Various MoM & EAI proprietary middleware
 - Web Services – the new kid on the block.

There's lots of choice

- Some based on **technical grounds**, including:
 - RPC versus message passing
 - Java specific versus multi-language
 - Direct versus indirect communication
 - Permanent versus occasional connection
 - Platform versus integration middleware
- Some based on **personal choice**

Intra-enterprise *versus* inter-enterprise

- Most middleware has been designed for intra-enterprise
- Inter-enterprise adds at least two challenges
 - **Firewalls** (& inter-enterprise security in general)
 - **Different middleware** may be used at the two ends
 - As well as different operating system, languages, etc

Web Services

- Aims to address both of these issues
 - Its protocol is layered on HTTP
 - So it can flow through a firewall
 - This “cheat” raises security and other concerns, but ones that need to be addressed in any case
 - It uses XML to format messages
 - So both sides can accept this as a “neutral” formatting approach

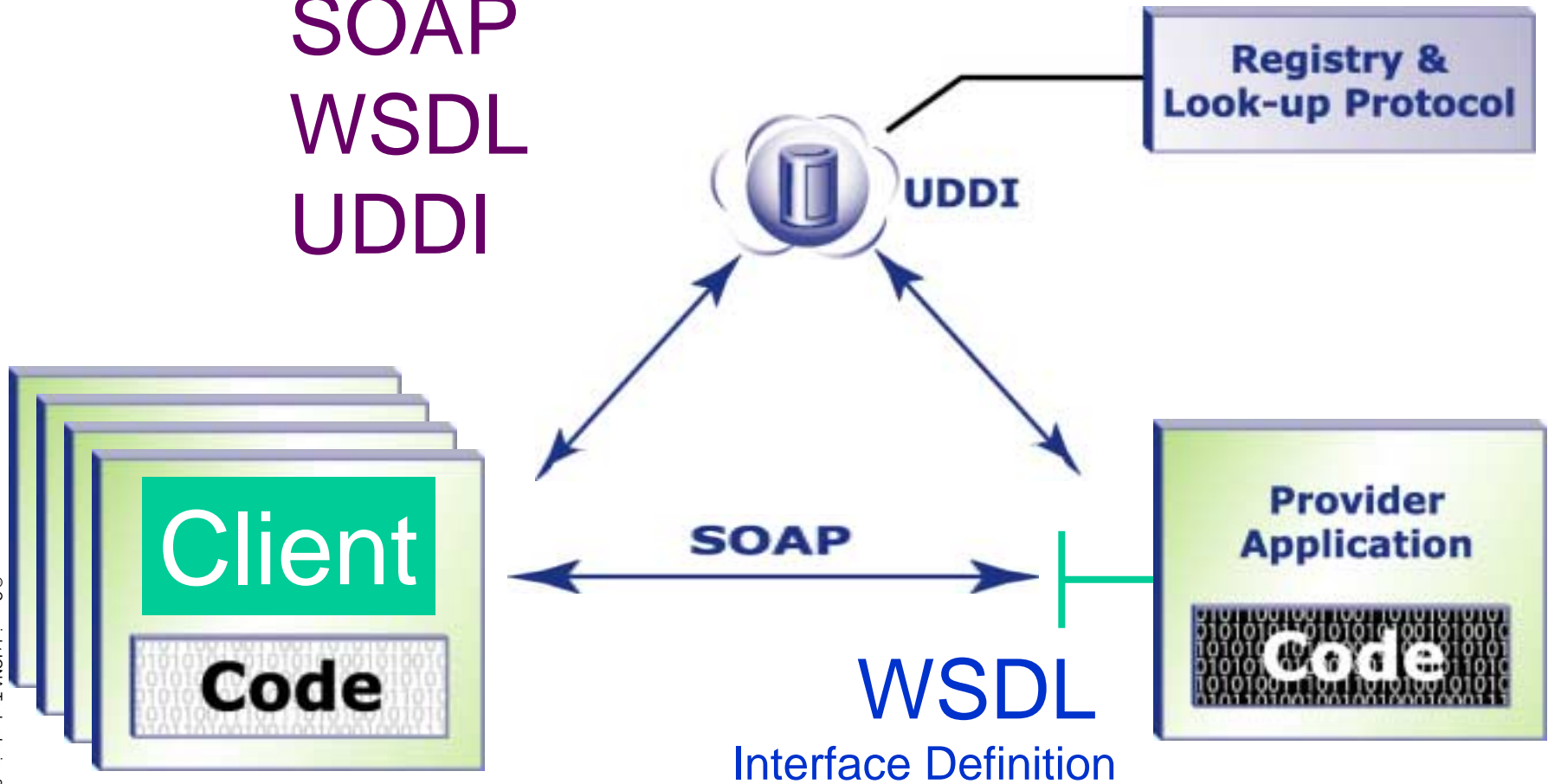
So what is a Web Service?

- Here's a call to a Web Service:
 - get the status of an order

```
<SOAP-ENV:Body>  
  <s:GetOrderStatus  
    xmlns:s="www.xmlbus.com/OrderEntry">  
    <orderno>12345</orderno>  
  </s:GetOrderStatus>  
</SOAP-ENV:Body>
```

Basic Web Services Standards

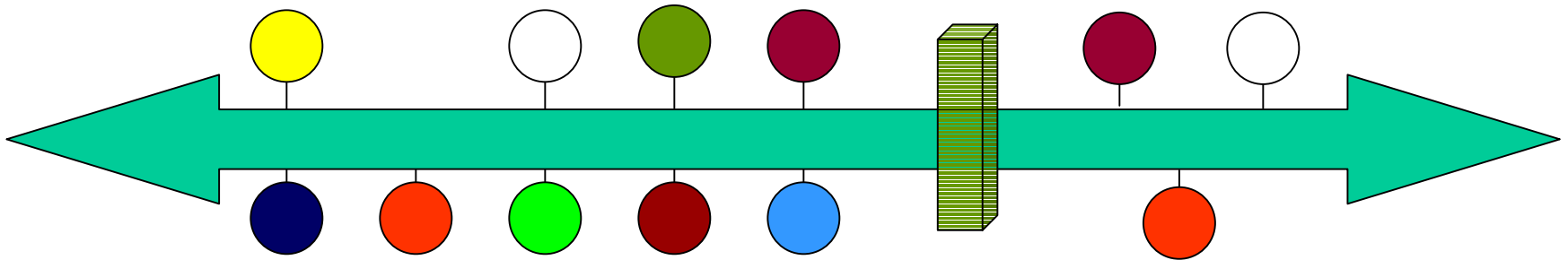
SOAP
WSDL
UDDI



© Copyright IONA Technologies 2002

Web Services – Higher Level View

- Service Oriented Architectures (SOA)
 - An IT architecture that exposes components, business functions, transactions and processes as well-defined services. SOAs are designed to allow reuse of services and rapid integration of applications with services.



Use of XML to format messages

- XML is a very general formatting tool
 - So use of XML doesn't ensure that both sides will understand the messages
 - XML isn't a panacea for interoperability
 - Also messages are larger (slower to marshall and send)
- However, it has advs for some uses (e.g. inter-enterprise)
 - “open” compared to proprietary / binary formats
 - XML has a powerful set of transformation tools

3 uses of Web Services

- Inter-enterprise communication
- Intra-enterprise (inter-middleware) communication
- Technology neutral interfaces to packages
 - e.g., ERP

Web Services won't replace existing middleware ...

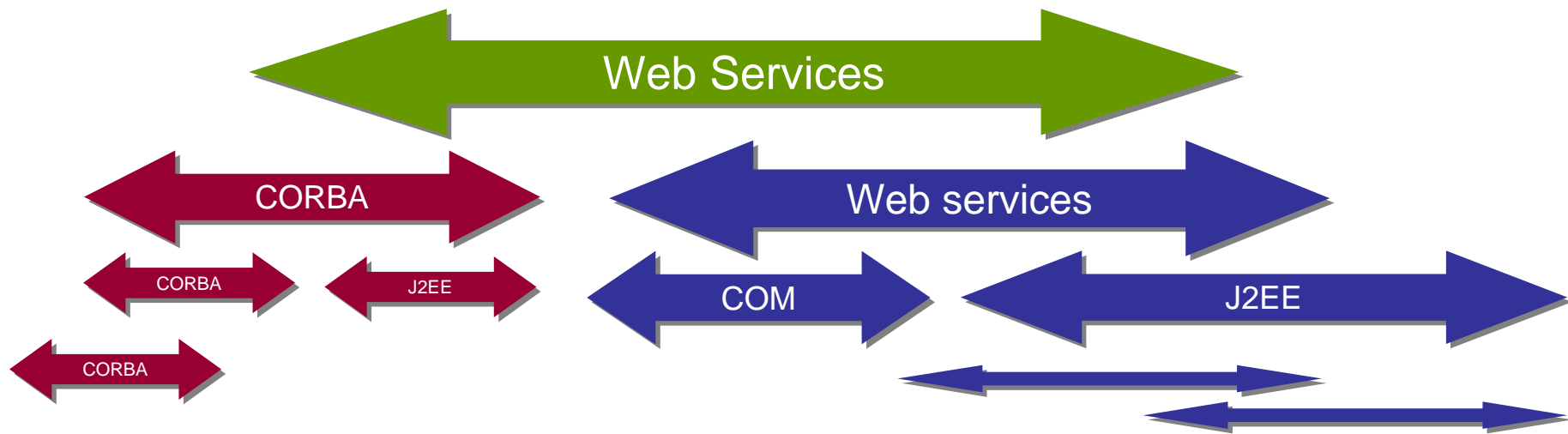
- Such as
 - CORBA
 - J2EE
 - MOM, EAI etc
- Why:
 - The incumbents are strong, well suited to their uses, and no one has money these days to fix what isn't broken
 - Web Services are “integration middleware” and not “platform middleware”
 - You can't implement services using the Web Services standards. CORBA, J2EE and .NET are the only platform middlewares.

although MOM & EAI will have to adapt

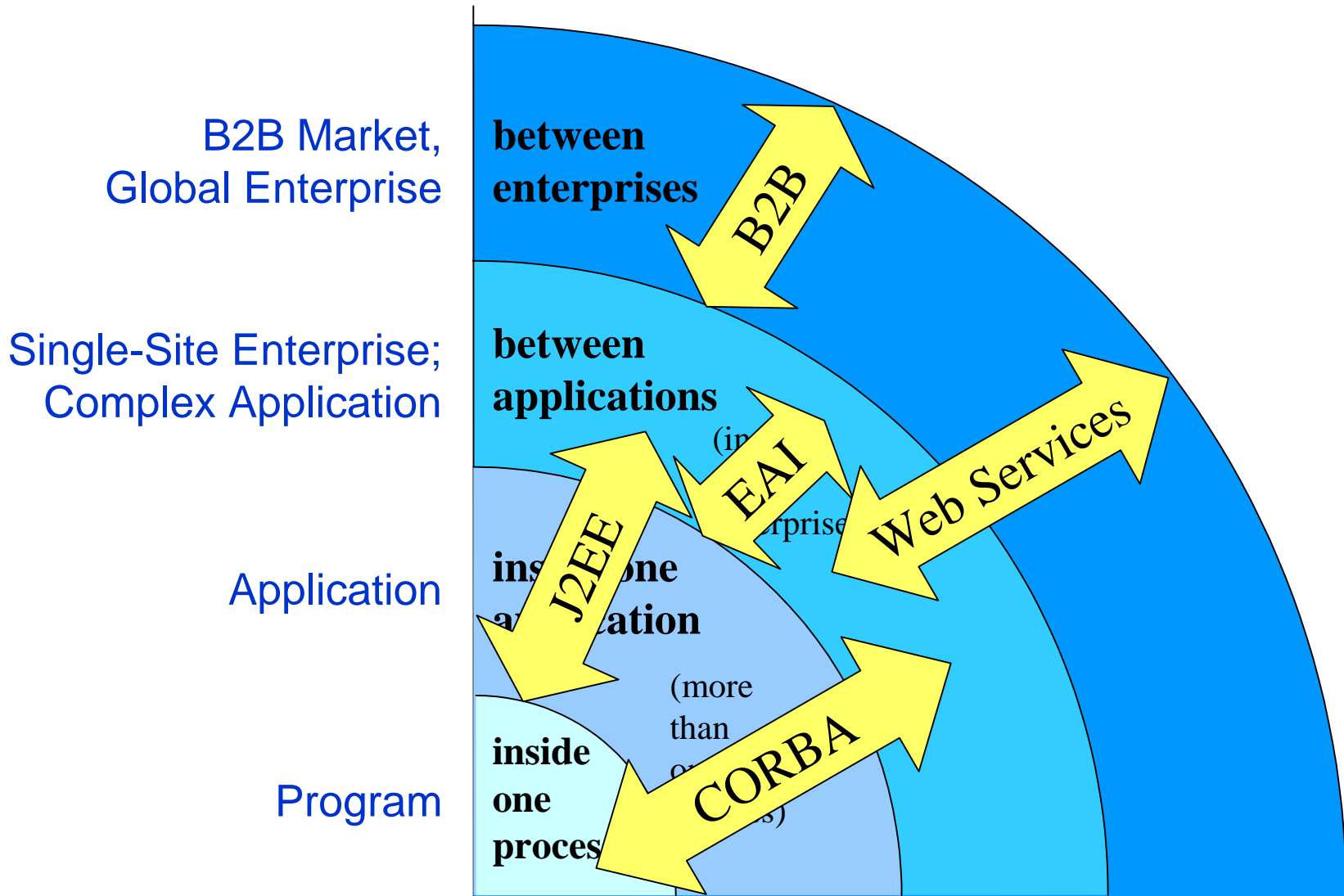


Web Services – relationship to other middleware

- Use Web Services
 - to integrate between different enterprises
 - and to integrate between different “islands” inside an enterprise

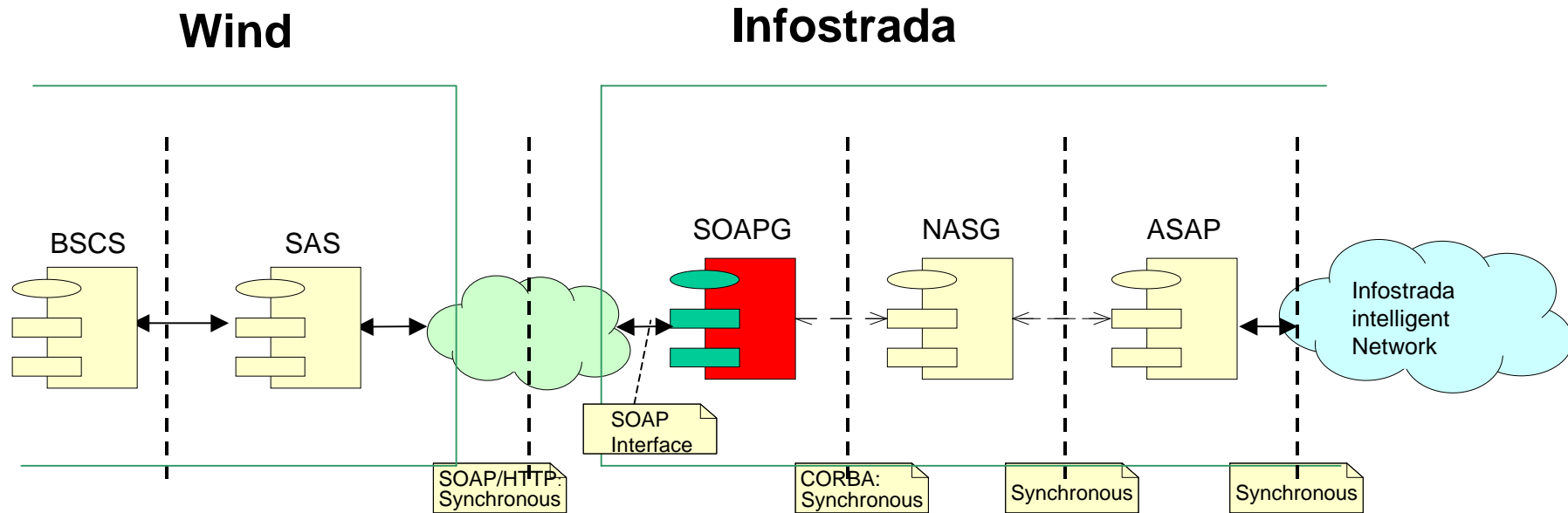


The Middleware Space



© Copyright IONA Technologies 2002

Example: InfoStrada Architecture

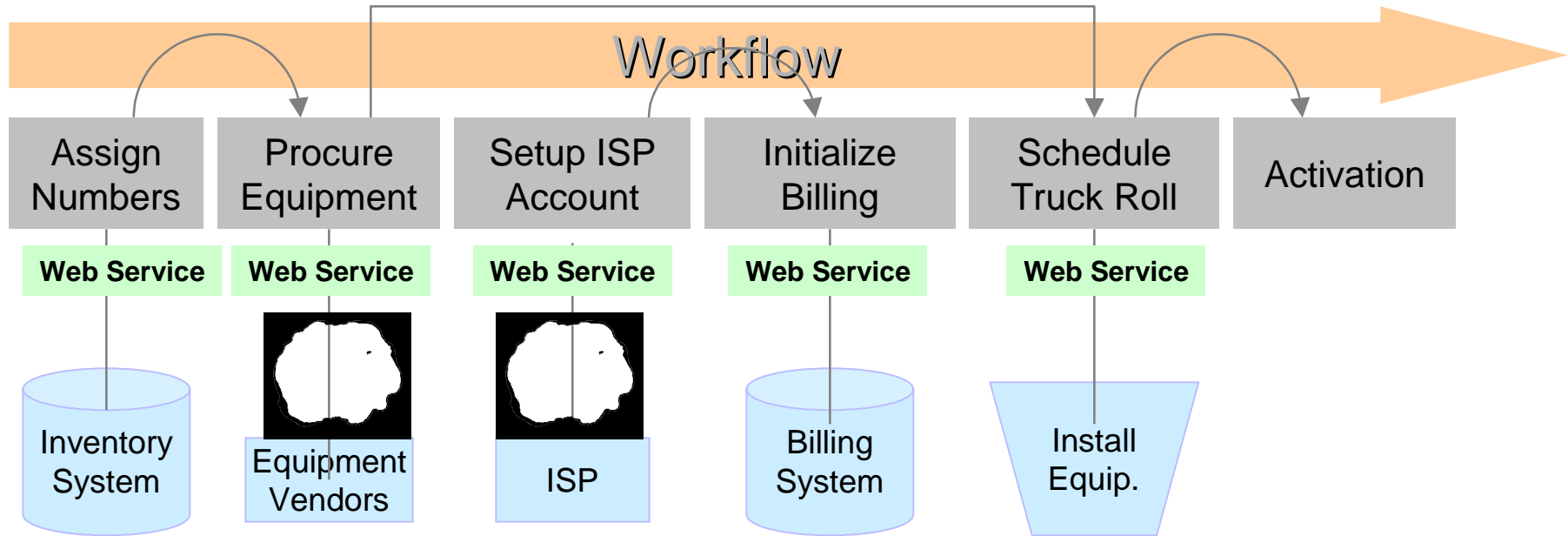


© Copyright IONA Technologies 2002

Web Services standardisation will continue for many years

- Security
 - Process Flow
 - “Transactions”
 - Unreliable connections
 - Store and Forward
 - Publish and subscribe
 - Management
 - ... and so on
- Concentration on the “sweet spots” for Web Services:
 - Inter-enterprise
 - Inter-middleware
 - Will the industry agree on a common set of standards?

Process Flow



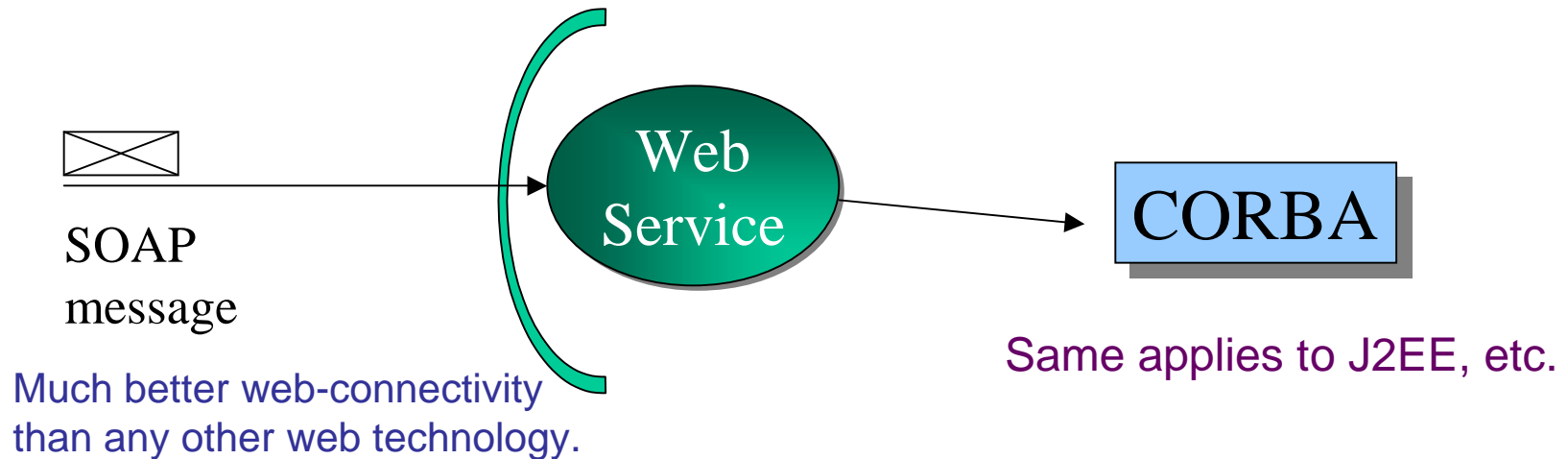
© Copyright IONA Technologies 2002

Web Services and CORBA

- CORBA is platform middleware
 - Multi-language, multi-platform, object/component oriented, robust, high-performance, reliable, transactional, secure,
 - And very well proven: banking, telecommunications, utilities, government, defence, computer animation, astronomy, weather simulation, and so on
- So what's missing then?
 - ... why would you need Web Services ?

What can Web Services offer CORBA?

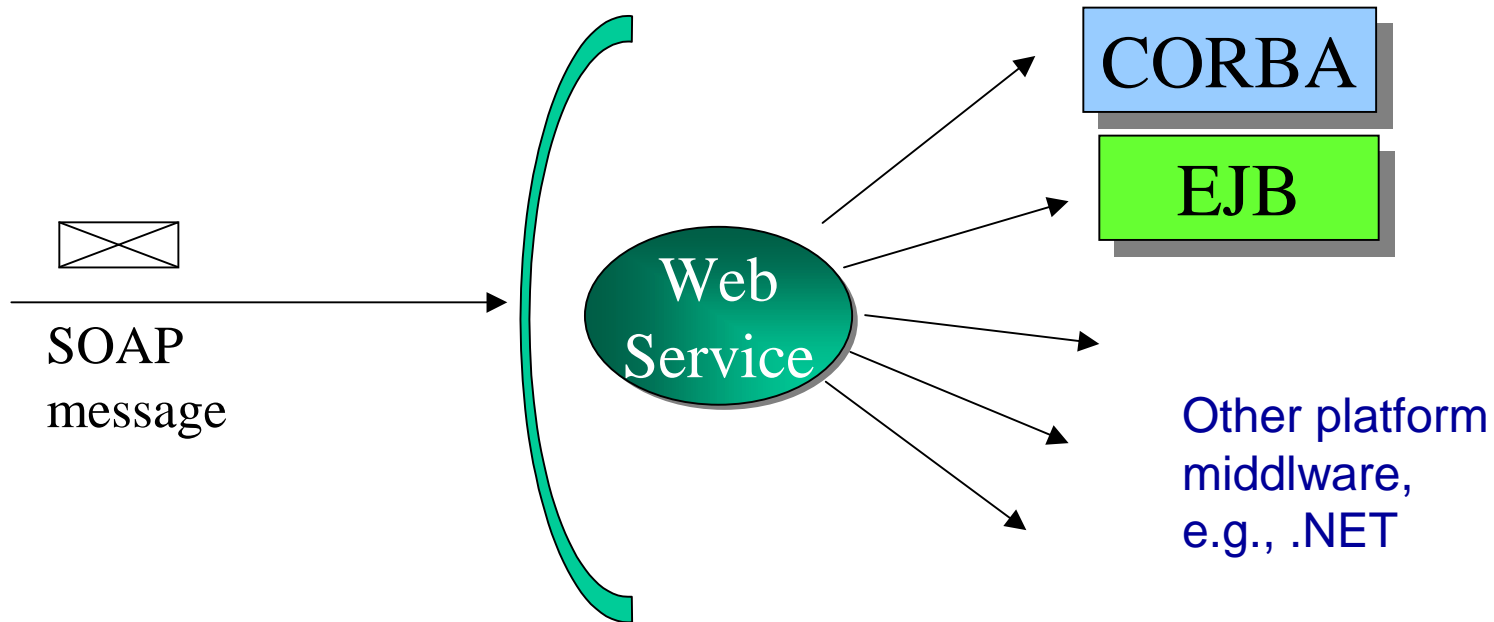
- Integration across the web
 - Not just across firewalls, but integration with the web (technology and acceptance)



- Integration across “islands” within an enterprise

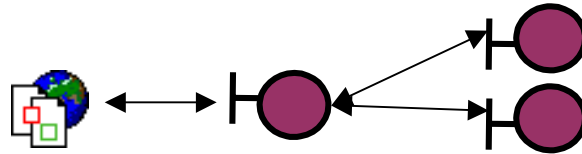
What can CORBA offer Web Services?

An implementation platform!

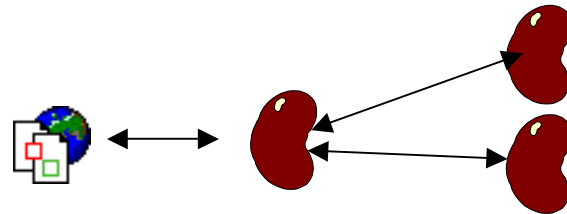


Aggregating backend services

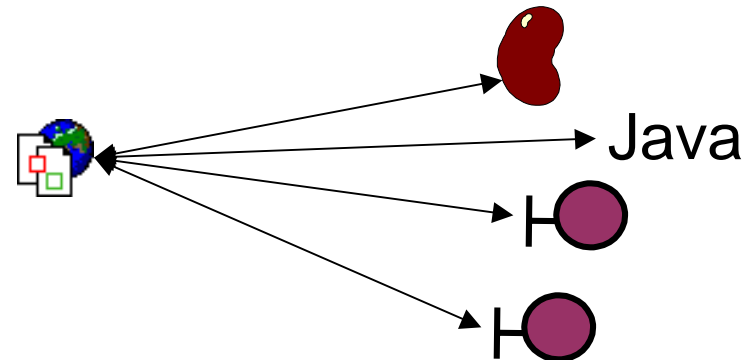
Aggregating
CORBA
services



Aggregating
J2EE
services

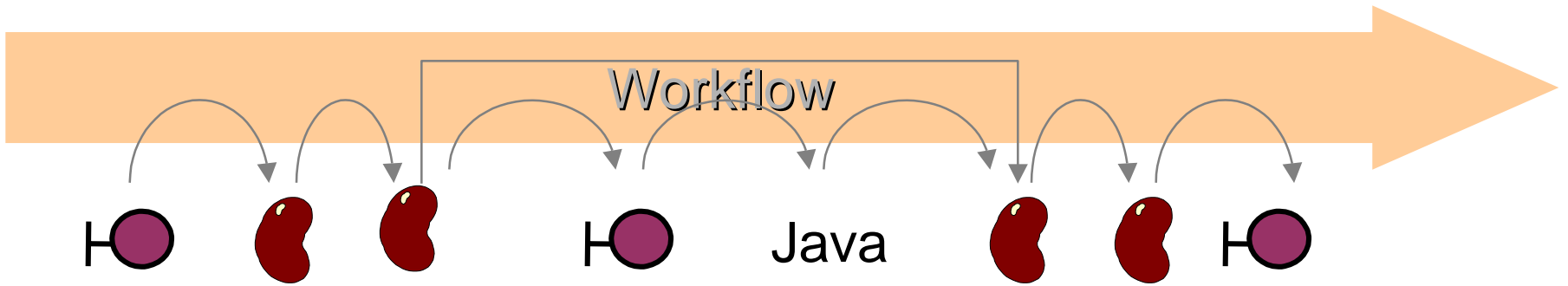


Using
XMLBus's
*Operation
Flows*



No coding!

... or you may need to use Business Processes



© Copyright IONA Technologies 2002

Other issues

- Other reasons why existing interfaces can't be translated directly into WSDL:
 - Latency – any 10x needs to be taken seriously
 - Some CORBA interfaces break the “don't trust the clients” rule
 - Some CORBA interfaces are “fragile”
 - Standard WSDL interfaces, or XML schemas

3 uses of Web Services

- Inter-enterprise communication
 - Previous web-based technology is very inappropriate
- Intra-enterprise (inter-middleware) communication
 - forget the idea of a single middleware
- Technology neutral interfaces to packages
 - e.g., ERP

Web Services ...

- ... won't replace existing middleware !
 - CORBA
 - J2EE
 - MOM, EAI etc

although MOM & EAI will have to adapt

Special support for integration of existing middleware with Web Services will be very valuable.

Web Services...

- ... is the basis for Middleware to Middleware Integration (M2M)
- But it's not the full story for this
 - M2M isn't trival
 - No middleware can subsume others
 - Too big. Compromises. Awkward translations. Wrong QoS.
 - Runtime translation costs can sometimes be too high
 - Many standards required – especially QoS issues



Discussion – Wednesday panel session (or during the coffee breaks)

Follow up – some of the later talks.