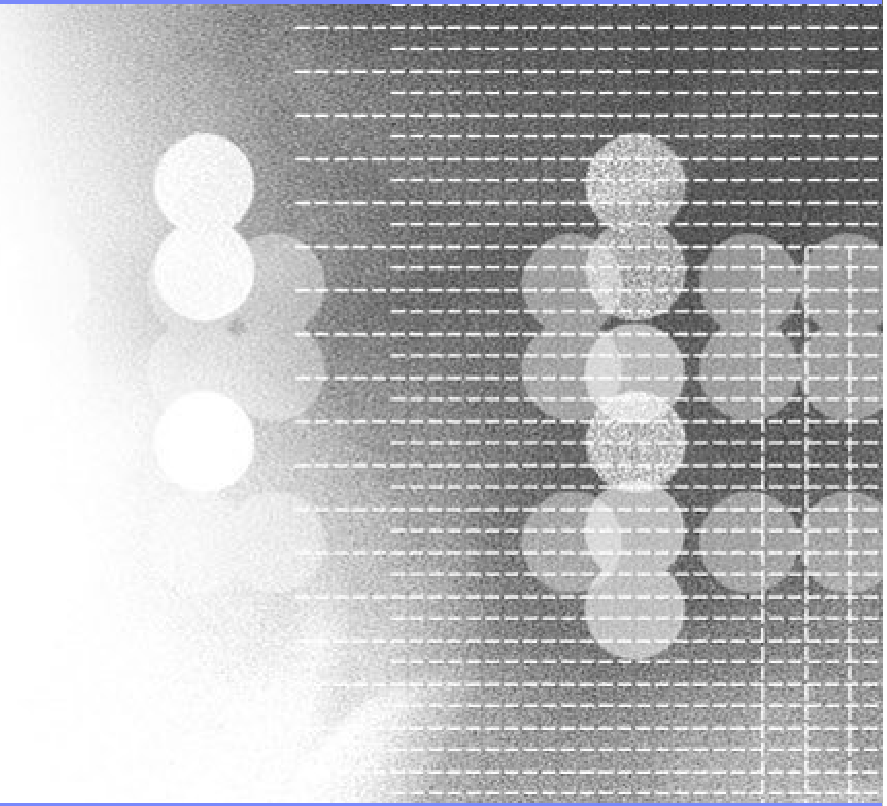
z/TPF EE V1.1
z/TPFDF V1.1
TPF Toolkit for WebSphere® Studio V3
TPF Operations Server V1.2

TPF Users Group Spring 2006

An SOA Primer



Name: Bill Cousins
Venue: Distributed Systems

AIM Enterprise Platform Software

IBM z/Transaction Processing Facility Enterprise Edition 1.1.0

© IBM Corporation 2006

Any references to future plans are for planning purposes only. IBM reserves the right to change those plans at its discretion. Any reliance on such a disclosure is solely at your own risk. IBM makes no commitment to provide additional information in the future.

Agenda

- SOA Overview
- SOA Programming Model
- SOA on TPF
- SOA Enablement Patterns
- Recap

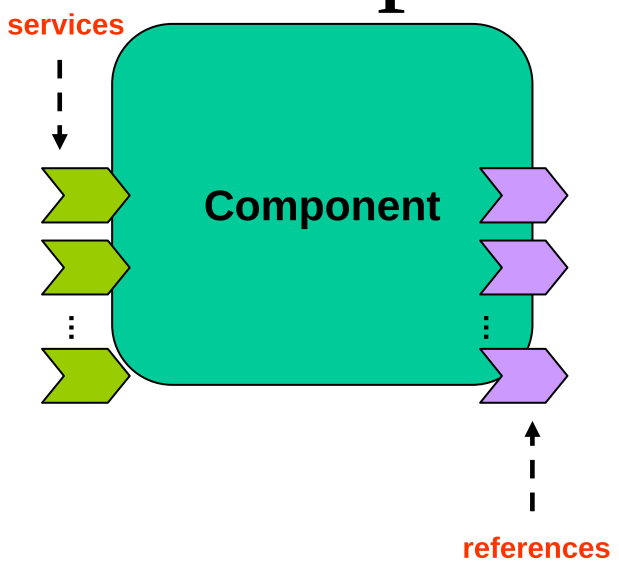
© IBM Corporation 2006

Service Oriented Architecture Overview

- What is a service?
  - A repeatable business task; a distinct business function
  - So what is that to my TPF application?
    - Application functions, implemented as transactions, can generally be transformed into services
    - Complex transactions may be implemented as several services
  - Why would I want to implement services?
    - Open up your business functions outside of TPF
    - Also make your customer data available
  - Services are implemented in *components*

© IBM Corporation 2006

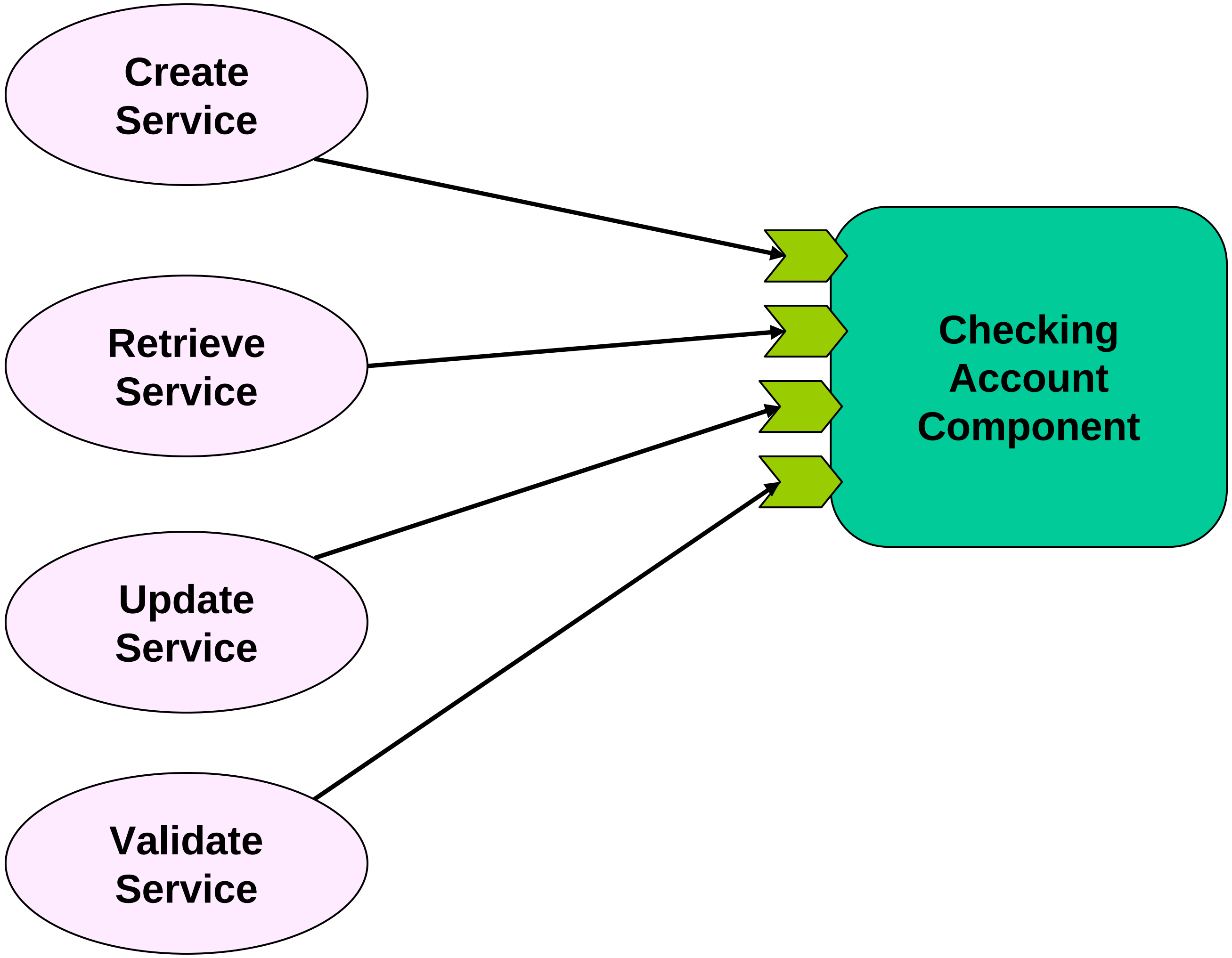
Components



- A component provides **Services** to other components
  - (The Service in SOA)
- Service **Interfaces** are defined *usually* with WSDL
  - (In the Web services implementation of SOA)
- Components are linked to create **composite applications**

© IBM Corporation 2006

Component Example



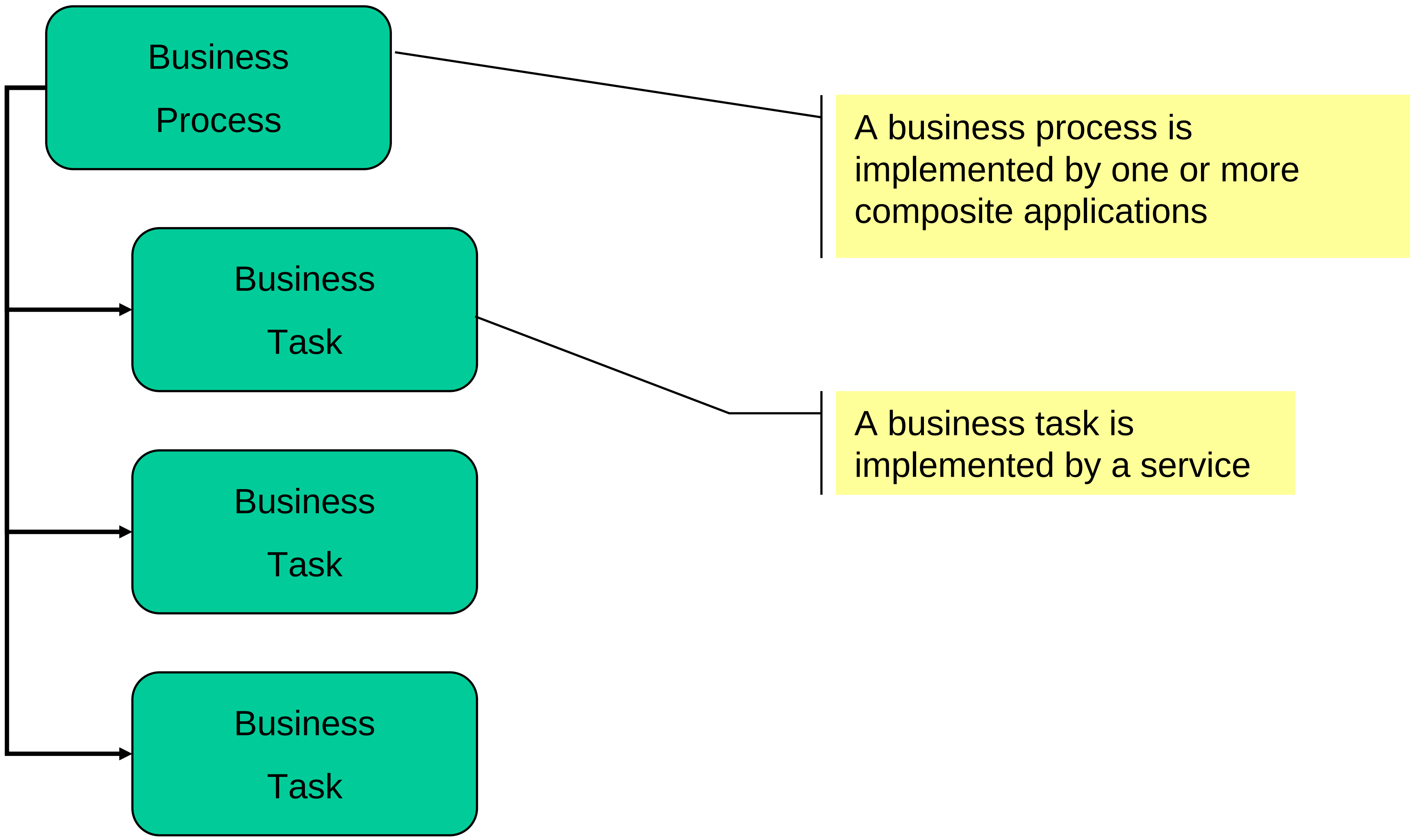
© IBM Corporation 2006

SOA Overview (cont’d)

- What is a composite application?
  - ➢A set of related and integrated services that support a business process
  - ➢This is how applications are built in an SOA environment
  - ➢Services can be deployed across a heterogeneous environment
    - 👍The full breadth of the enterprise can be leveraged to implement new business processes
    - 👍Reuse and increased flexibility provides quicker time to market

© IBM Corporation 2006

SOA and Business Process

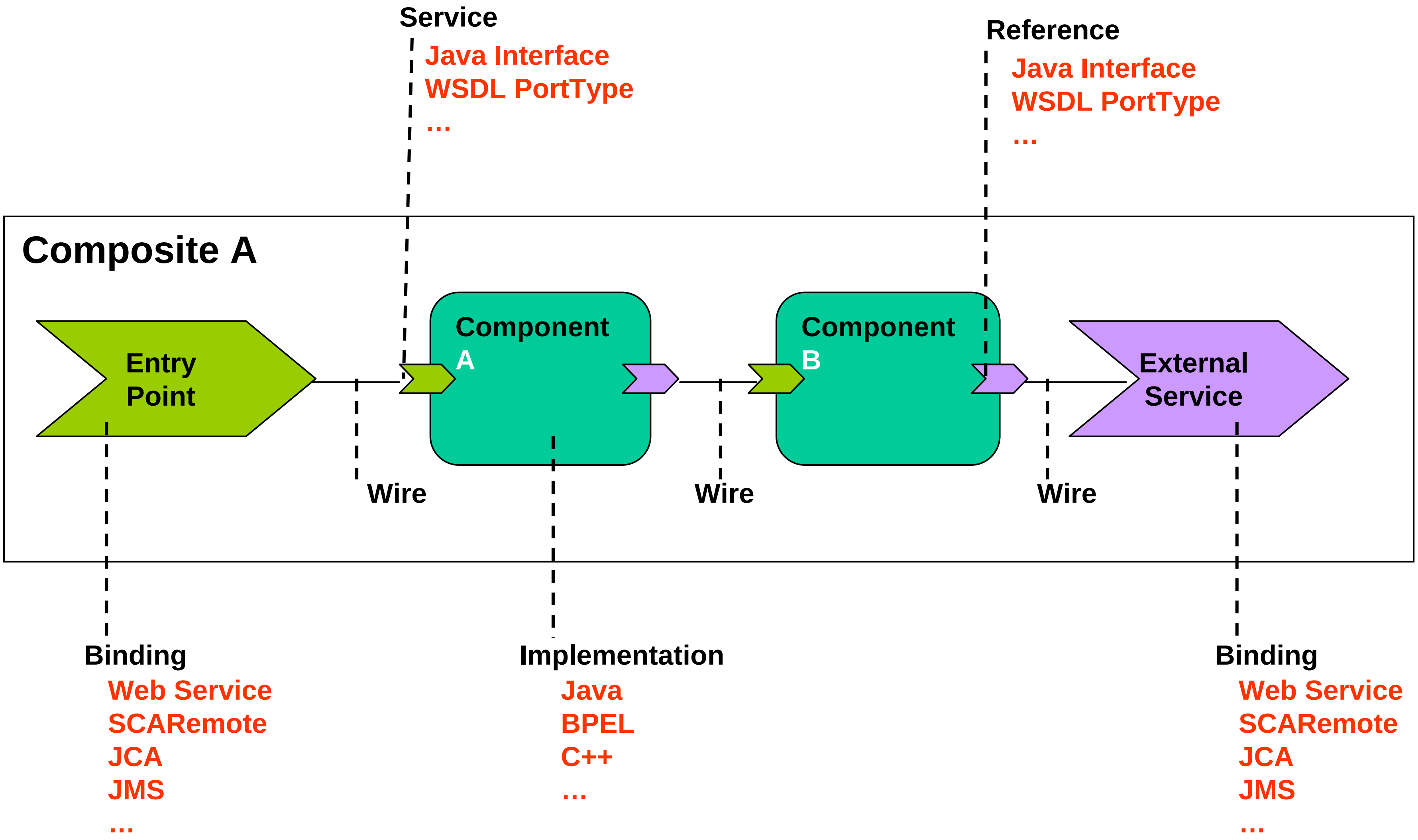


A business process is implemented by one or more composite applications

A business task is implemented by a service

© IBM Corporation 2006

Composite Application



Service

Java Interface
WSDL PortType
…

Reference

Java Interface
WSDL PortType
…

Binding

Web Service
SCARemote
JCA
JMS
…

Implementation

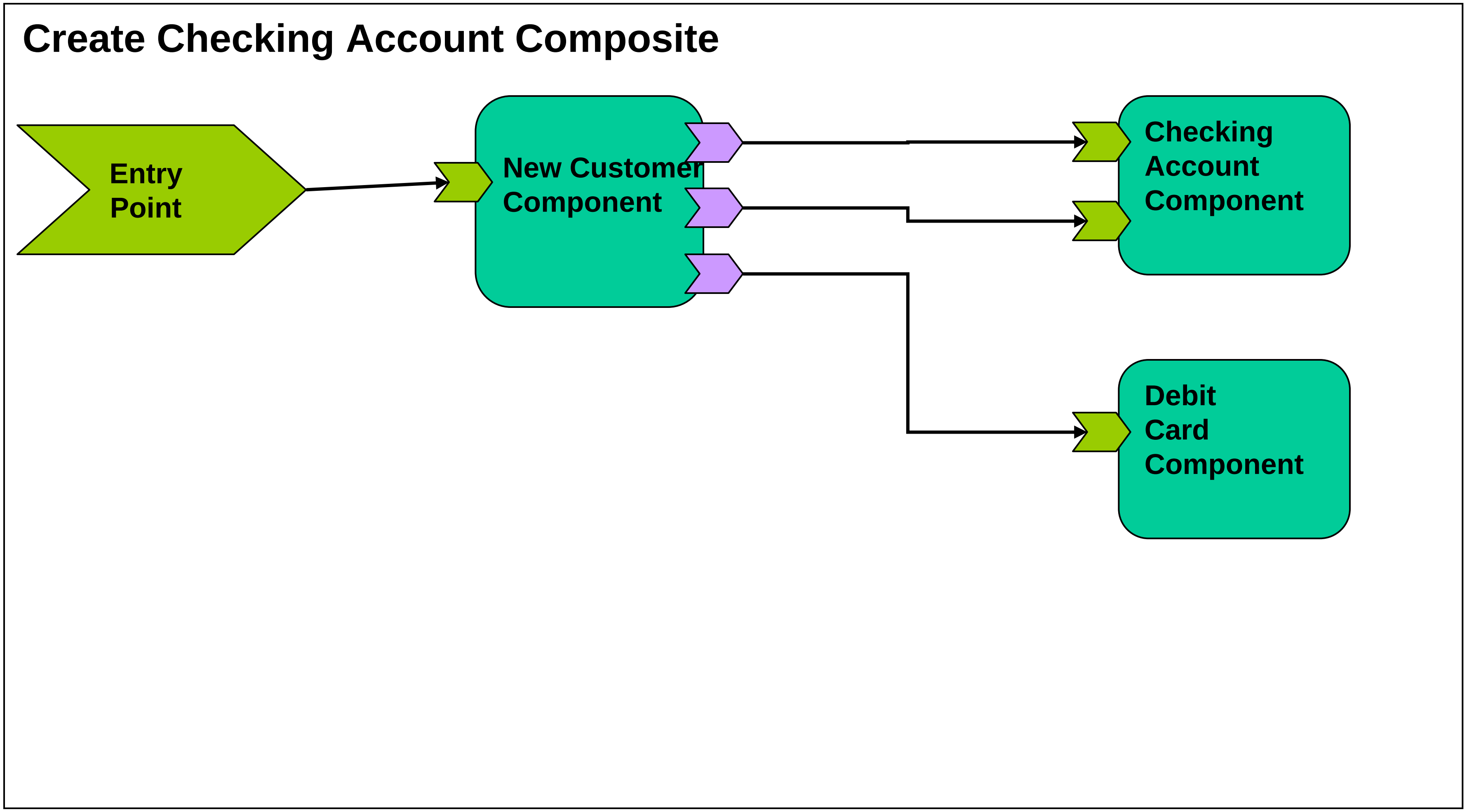
Java
BPEL
C++
…

Binding

Web Service
SCARemote
JCA
JMS
…

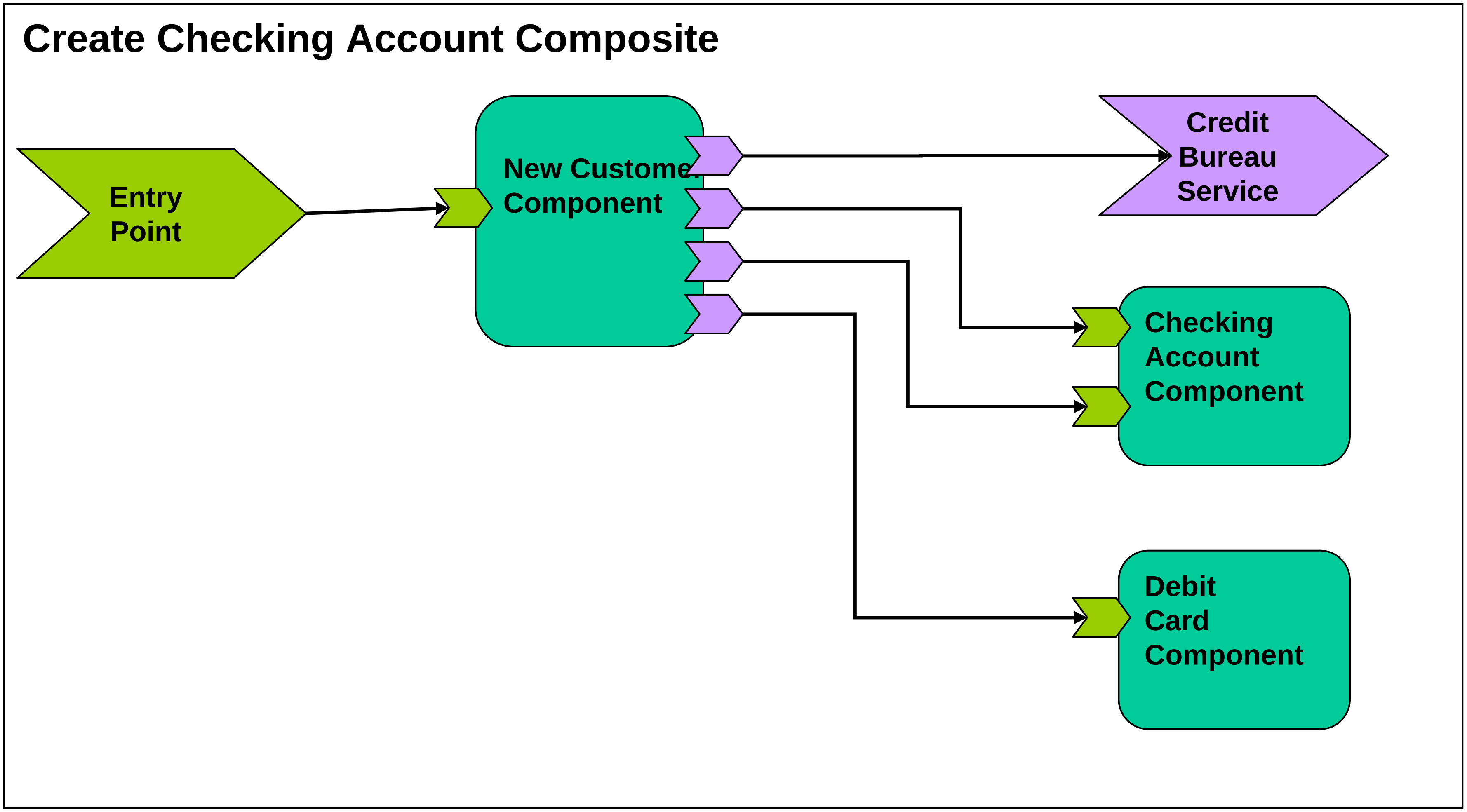
© IBM Corporation 2006

Composite Application Example



Create Checking Account Composite

Composite Application Example



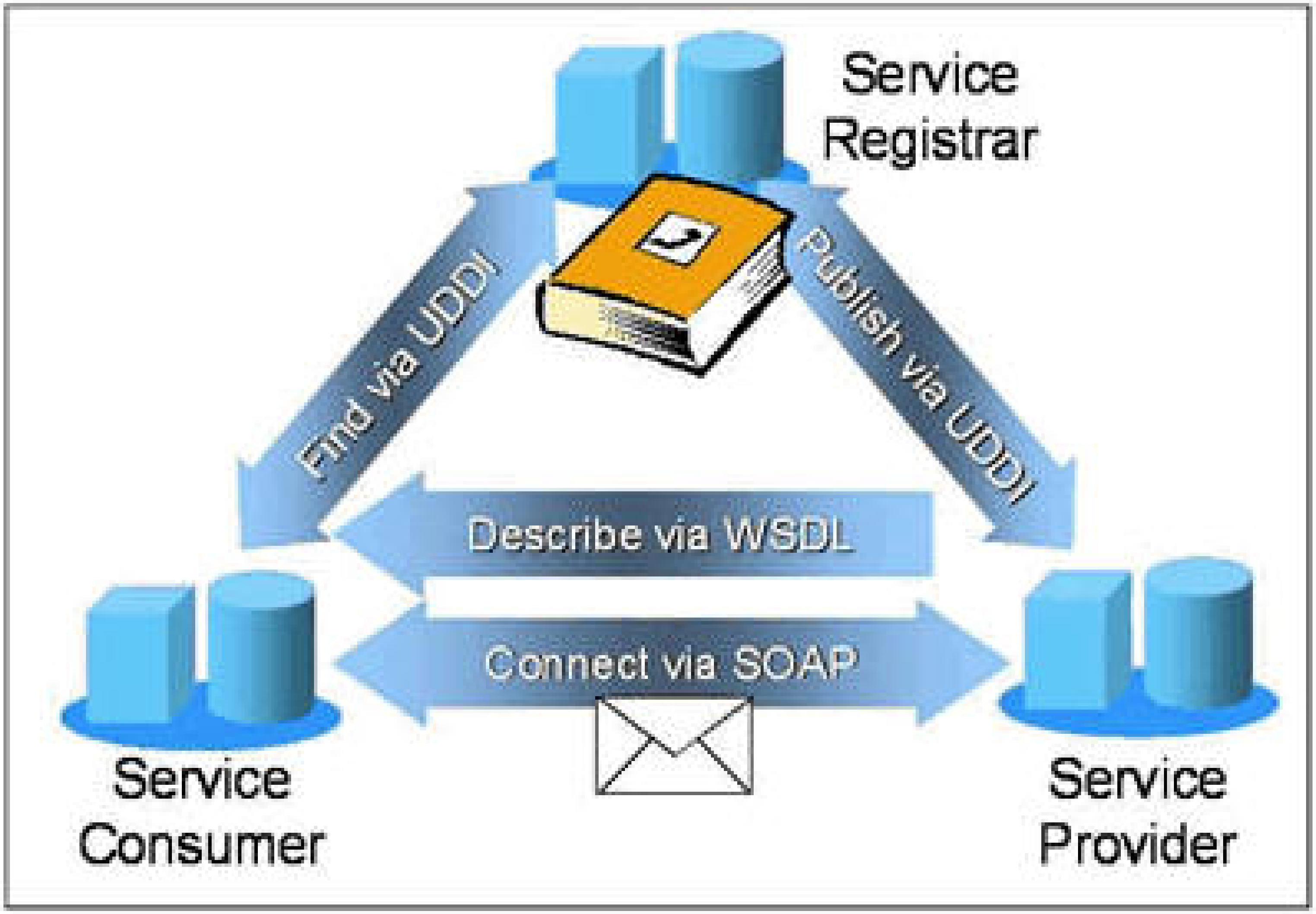
© IBM Corporation 2006

SOA Overview

- What is SOA?
  - An IT architectural style that supports service orientation
  - Creates new applications by coordinating services
  - New terms to be aware of:
    - Service consumer
    - Service provider
    - Service registrar

© IBM Corporation 2006

SOA Roles and Operations



© IBM Corporation 2006

What does SOA mean to…

- a business analyst
  - ➢a set of business services that depict the business process to be exposed to customers and clients
- a software architect
  - ➢an architectural style that requires a service provider, service consumer, and service description
- a software developer
  - ➢a programming model with standards, tools, and methodologies to implement services

© IBM Corporation 2006

SOA Programming Model

- Service Component Architecture
  - ➢Provides ability to integrate services into composite applications
- Service Data Objects
  - ➢Standardizes data access across heterogeneous data sources
  - ➢Data Access Service
    - Handles connectivity to a data source
- Enterprise Service Bus
  - ➢Provides connectivity between services and composite applications

© IBM Corporation 2006

SOA on TPF

- The SOA implementation on TPF is with Web services
  - Web services is becoming the primary implementation of SOA (XML, SOAP, HTTP (Apache), …)
- Allows TPF to participate in an enterprise-wide SOA environment
  - Business analysts and software architects can include TPF in their plans
- Opens TPF applications and data to outside systems
  - Using standard, open methods

© IBM Corporation 2006

Web Services

- Standards for documenting service interfaces and invoking services
- Focuses on distinct services and business operations
- Exploits Internet standards
  - ➢For example, HTTP/S, XML
- Flexible aggregation of services into new, modified business processes and solutions

© IBM Corporation 2006

SOA Enablement Patterns

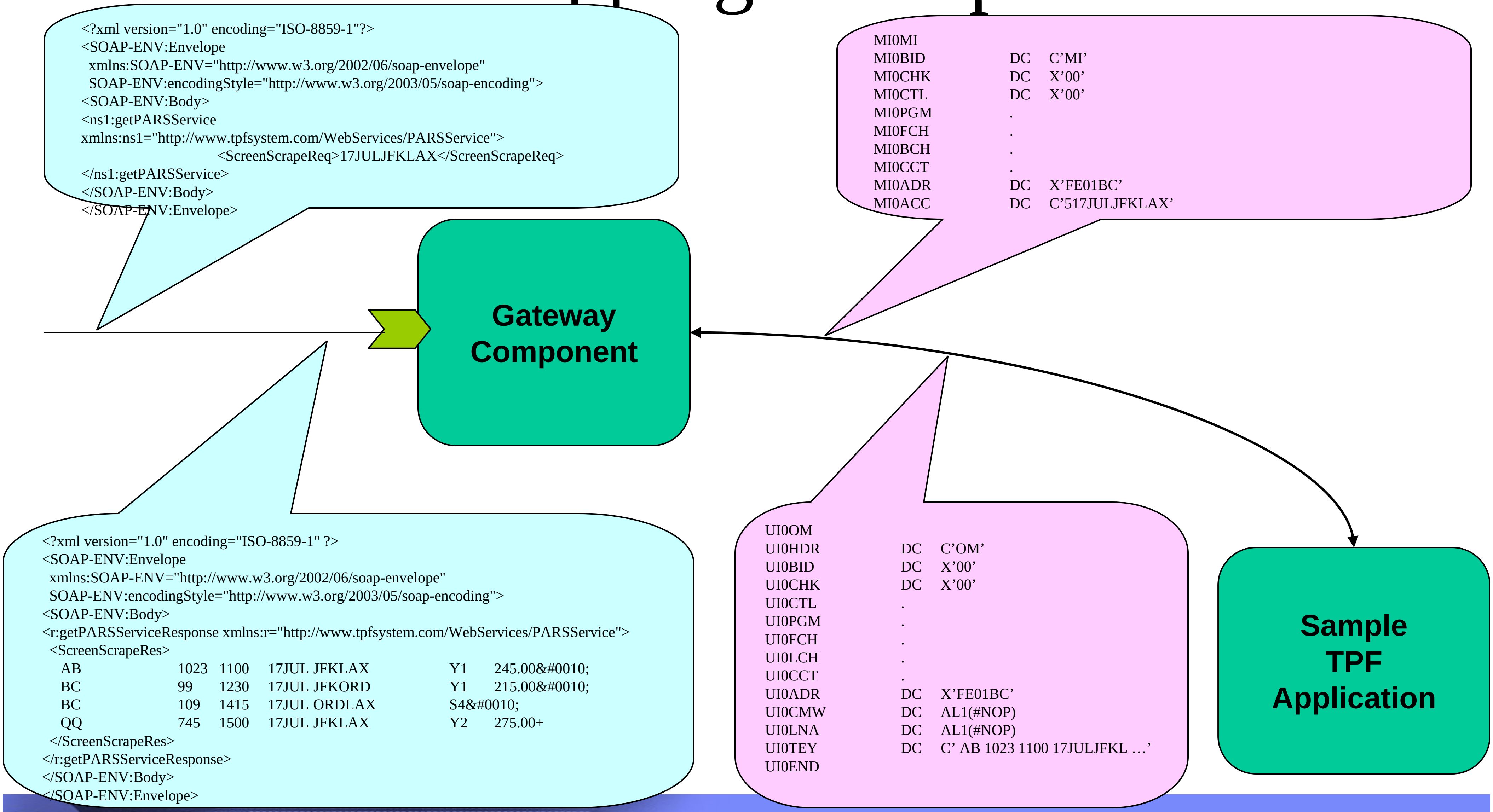
© IBM Corporation 2006

SOA Enablement Patterns (cont’d)

- Wrapping
  - An existing application has a gateway attached to manipulate input and output messages to provide an SOA interface
  - Changes are made to the application to return to the gateway instead of sending the response message
  - This pattern allows for access to the application but limited reuse and flexibility

© IBM Corporation 2006

Wrapping Example



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2002/06/soap-envelope"
  SOAP-ENV:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
<SOAP-ENV:Body>
<ns1:getPARSService
xmlns:ns1="http://www.tpfsystem.com/WebServices/PARSService">
                  <ScreenScrapeReq>17JULJFKLAX</ScreenScrapeReq>
</ns1:getPARSService>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
MI0MI
MI0BID          DC   C'MI'
MI0CHK          DC   X'00'
MI0CTL          DC   X'00'
MI0PGM          .
MI0FCH          .
MI0BCH          .
MI0CCT          .
MI0ADR          DC   X'FE01BC'
MI0ACC          DC   C'517JULJFKLAX'
```

Gateway Component

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://www.w3.org/2002/06/soap-envelope"
  SOAP-ENV:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
<SOAP-ENV:Body>
<r:getPARSServiceResponse xmlns:r="http://www.tpfsystem.com/WebServices/PARSService">
  <ScreenScrapeRes>
    AB          1023   1100    17JUL JFKLAX           Y1     245.00&#0010;
    BC          99     1230    17JUL JFKORD           Y1     215.00&#0010;
    BC          109    1415    17JUL ORDLAX           S4&#0010;
    QQ          745    1500    17JUL JFKLAX           Y2     275.00+
  </ScreenScrapeRes>
</r:getPARSServiceResponse>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```
UI0OM
UI0HDR          DC   C'OM'
UI0BID          DC   X'00'
UI0CHK          DC   X'00'
UI0CTL          .
UI0PGM          .
UI0FCH          .
UI0LCH          .
UI0CCT          .
UI0ADR          DC   X'FE01BC'
UI0CMW          DC   AL1(#NOP)
UI0LNA          DC   AL1(#NOP)
UI0TEY          DC   C' AB 1023 1100 17JULJFKL …'
UI0END
```

Sample TPF Application

© IBM Corporation 2006

SOA Enablement Patterns (cont’d)

- Refacing
  - An existing application is modified to provide a new SOA interface; business logic is separated from the message logic
  - A service wrapper is written to transform datatypes between XML and native format needed by the application
  - This pattern allows for access to the application and increased reuse and flexibility

© IBM Corporation 2006

Refacing Example

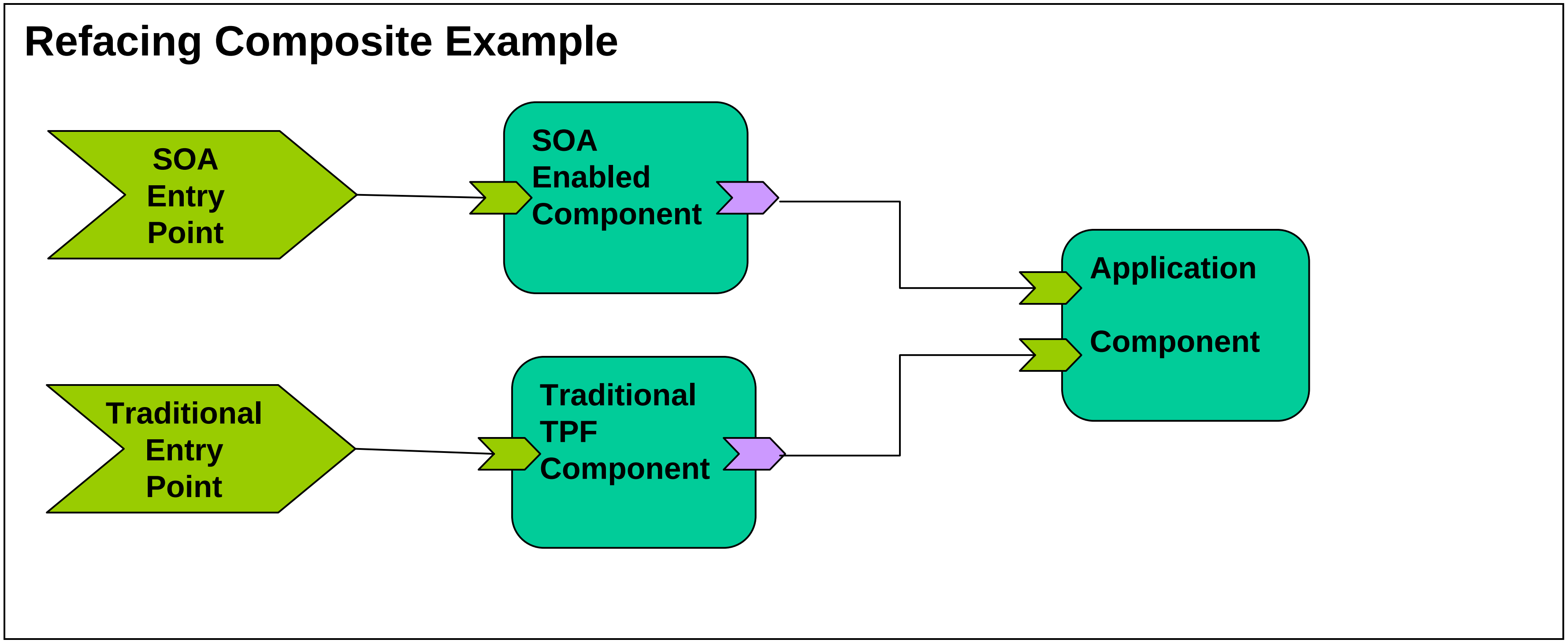
- **Sample Request Message**
- <?xml version="1.0" encoding="ISO-8859-1"?>
- <SOAP-ENV:Envelope
-   xmlns:SOAP-ENV="http://www.w3.org/2002/06/soap-envelope"
-   SOAP-ENV:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
- <SOAP-ENV:Body>
- <ns1:getAvailability xmlns:ns1="http://www.tpfsystem.com/WebServices/AvailabilityService">
-   <Date>
-     <Day>17</Day>
-     <Month>JUL</Month>
-   </Date>
-   <CityPair>
-         <DepartureCity>JFK</DepartureCity>
-         <DestinationCity>LAX</DestinationCity>
-   </CityPair>
- </ns1:getAvailability>
- </SOAP-ENV:Body>
- </SOAP-ENV:Envelope>

Refacing Example (cont’d)

- **Sample Response Message**

```
•   <?xml version="1.0" encoding="ISO-8859-1" ?>
•   <SOAP-ENV:Envelope
•     xmlns:SOAP-ENV="http://www.w3.org/2002/06/soap-envelope"
•     SOAP-ENV:encodingStyle="http://www.w3.org/2003/05/soap-encoding">
•   <SOAP-ENV:Body>
•   <r:getAvailabilityResponse xmlns:r="http://www.tpfsystem.com/WebServices/AvailabilityService">
•               <carrier>AB</carrier>
•               <flight>1023</flight>
•               <time>1100</time>
•               <Date>
•               <Day>17</Day>
•               <Month>JUL</Month>
•               </Date>
•               <CityPair>
•                           <DepartureCity>JFK</DepartureCity>
•                           <DestinationCity>LAX</DestinationCity>
•               </CityPair>
•               <class-of-service>Y1</class-of-service>
•               <price>245.00</price>
•   </r:getAvailabilityResponse>
•   </SOAP-ENV:Body>
•   </SOAP-ENV:Envelope>
```

Refacing Example (cont'd)



© IBM Corporation 2006

SOA Enablement Patterns (cont’d)

- Componentizing
  - The existing application is deconstructed into components.
  - This pattern is more applicable to new applications
  - Creates service interfaces to internal functions that were not previously accessible
  - This pattern allows for maximum value for access to the application, for reuse and flexibility

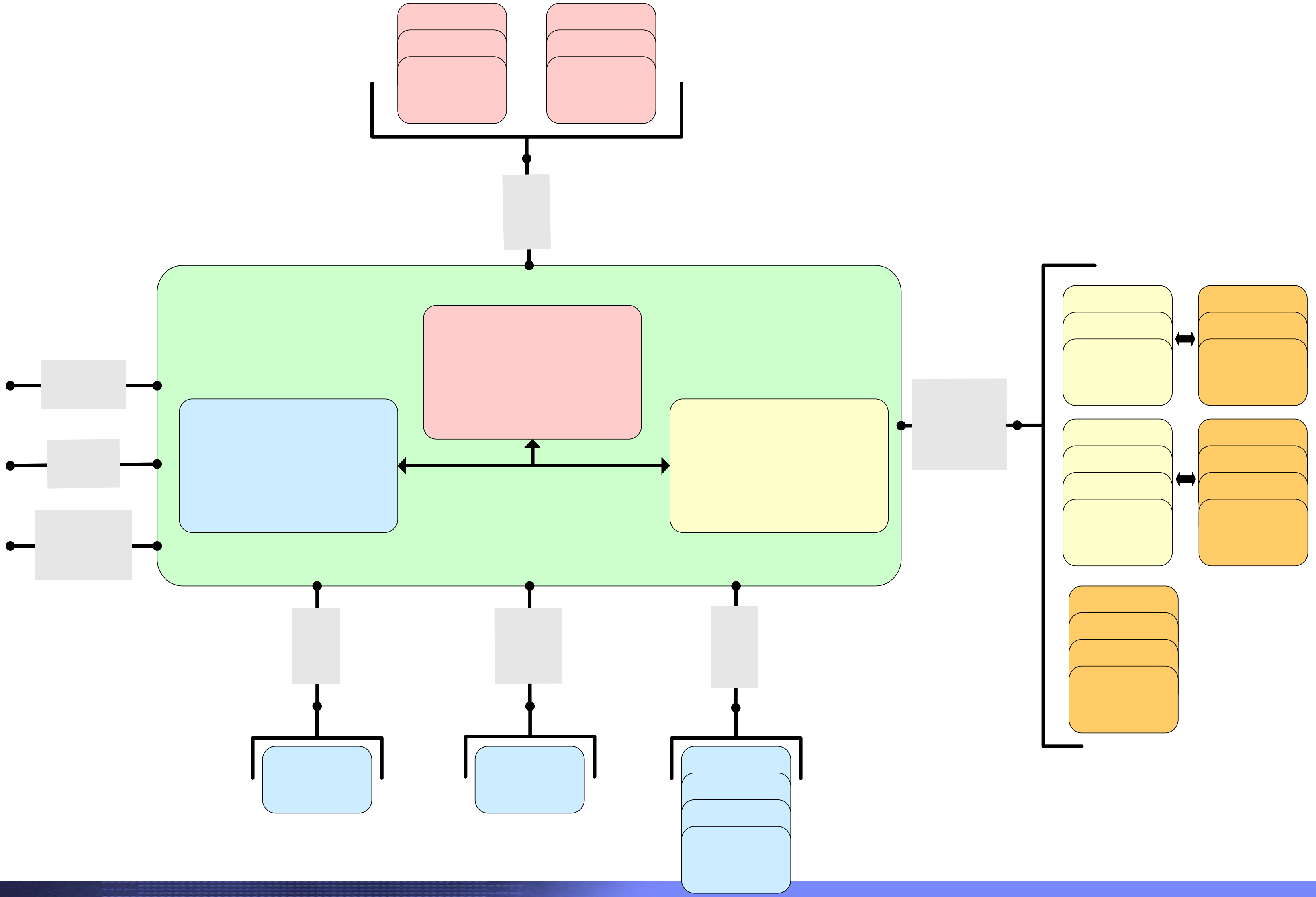
© IBM Corporation 2006

SOA Enablement Considerations

- Create application portfolio
- Incremental changes
- Defensive coding
- Session state (AAA, RCB, etc.)
- Strategy and planning

© IBM Corporation 2006

SOA Direction of Intent for z/TPF



© IBM Corporation 2006

Recap

- SOA can be implemented incrementally
  - Use it to open TPF business logic and data
- Wider adoption of SOA yields greater benefits
  - Enable TPF applications to participate in enterprise-wide business processes
- Web site for z/TPF SOA Whitepaper:
  - http://www.ibm.com/software/htp/tpf/pages/SOA_White_Paper_v1_2.pdf
- For further information, see your IBM customer rep

© IBM Corporation 2006

Trademarks

IBM, xxx, xxx, and are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.
Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.
Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
UNIX is a registered trademark of The Open Group in the United States and other countries

Other company, product, or service names may be trademarks or service marks of others.

Notes

All customer examples cited or described in this presentation are presented as illustrations of the manner in which some customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics will vary depending on individual customer configurations and conditions.

This publication was produced in the United States. IBM may not offer the products, services or features discussed in this document in other countries, and the information may be subject to change without notice. Consult your local IBM business contact for information on the product or services available in your area.

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

Information about non-IBM products is obtained from the manufacturers of those products or their published announcements. IBM has not tested those products and cannot confirm the performance, compatibility, or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Prices subject to change without notice. Contact your IBM representative or Business Partner for the most current pricing in your geography.

This presentation and the claims outlined in it were reviewed for compliance with US law. Adaptations of these claims for use in other geographies must be reviewed by the local country counsel for compliance with local laws.

© IBM Corporation 2006