



PRISMTECH

The Integration Server Company

Web Services Messaging

What constitutes Web Services?

- ▶ HTTP transport?
- ▶ WSDL?
- ▶ UDDI?
- ▶ SOAP as an RPC mechanism?
- ▶ Java?
- ▶ J2EE?
- ▶ .NET?
- ▶ Tools to expose my existing components through SOAP interface?
- ▶ Tools to create language binding to SOAP interface?

Another View

- ▶ Web services are loosely coupled software components, delivered over a network via standards-based technologies
 - ▶ Public or private
 - ▶ Uses SOAP as a major building block
 - ▶ As a message format
 - ▶ As an RPC mechanism
- ▶ Web services is a way for applications to integrate with each other

SOAP Communication

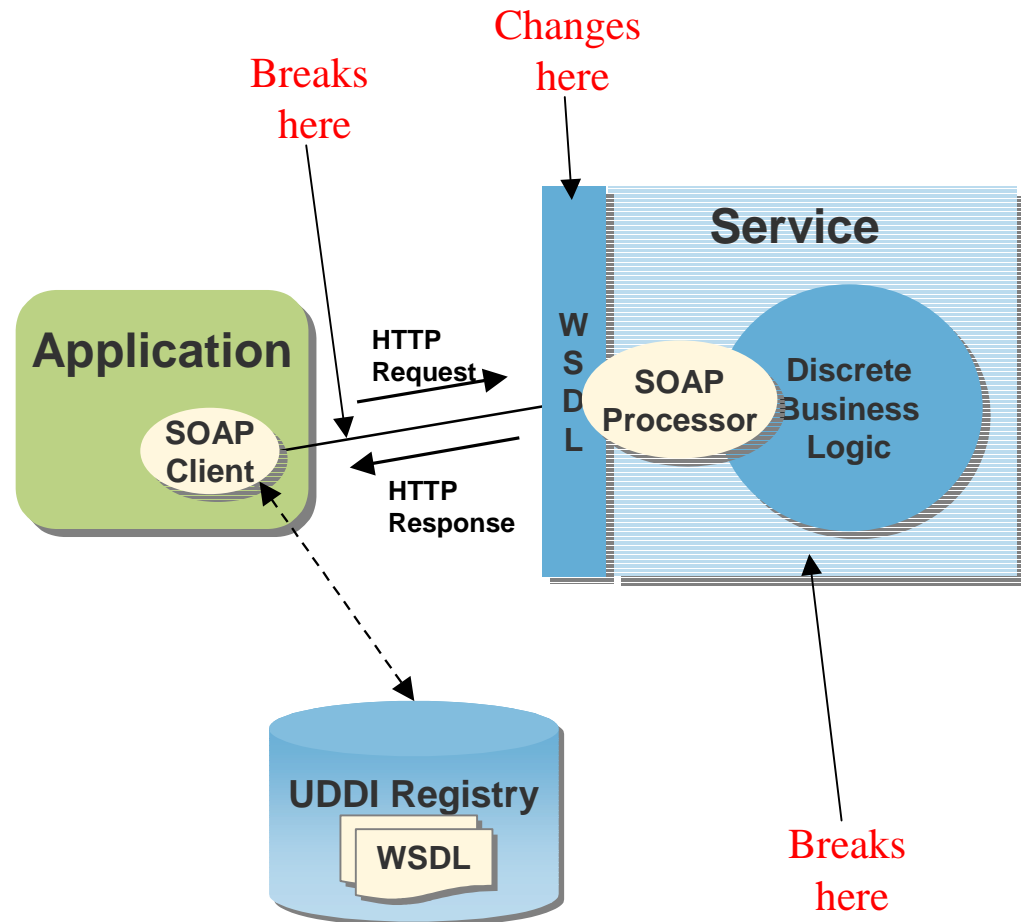
- ▶ SOAP Message Exchange Model
 - ▶ Describes the processing common to all SOAP nodes upon receipt of a message
 - ▶ Identify mandatory header blocks
 - ▶ Process header blocks aimed at the node
- ▶ SOAP Binding Framework
 - ▶ Augments the core rules for a specific protocol
 - ▶ Default binding is HTTP
- ▶ These rules allow for additional models
 - ▶ Request/Response
 - ▶ Reliable Messaging

The Need for Messaging

- ▶ Web Services must exchange critical business information to succeed
- ▶ Main features for business critical systems include the following
 - ▶ Security
 - ▶ Performance
 - ▶ Reliability
 - ▶ Integration

Why Messaging Rather than RPC?

- ▶ Web services live in a *usually-connected* world
 - ▶ Internet bandwidth varies due to traffic
 - ▶ Maybe completely down
 - ▶ No guarantees that business partners are connected
 - ▶ Server or network down
 - ▶ Scheduled downtimes
- ▶ Loosely-coupled distributed architecture
 - ▶ All parts don't have to always be there
- ▶ Loosely-coupled from design
 - ▶ Document-centric communication leaves you isolated from API changes



Other Messaging Benefits

- ▶ Even if you are principally interested in more tightly – coupled request/reply style interaction, then SOAP for RPC, UDDI and XML are only part of the solution
 - ▶ HTTP post is not a provider suitable for mission critical systems
 - ▶ Qualities of service
 - ▶ Guaranteed delivery
 - ▶ Dead letter drops
 - ▶
 - ▶ Configuration
 - ▶ Management and instrumentation
 - ▶ Single point of integration

Web Services Integration Strategy

- ▶ Common Messaging Backbone using CosNotification
 - ▶ Provides a strong foundation for the architecture
 - ▶ Mature Qualities of Service (Persistence)
 - ▶ Performance (federation)
 - ▶ Management (admin objects)
 - ▶ Various mappings allow interoperability of JMS, CosNotification, and SOAP nodes
 - ▶ JMS/CosNotification Interworking
 - ▶ Use JMS as the provider for JAXM

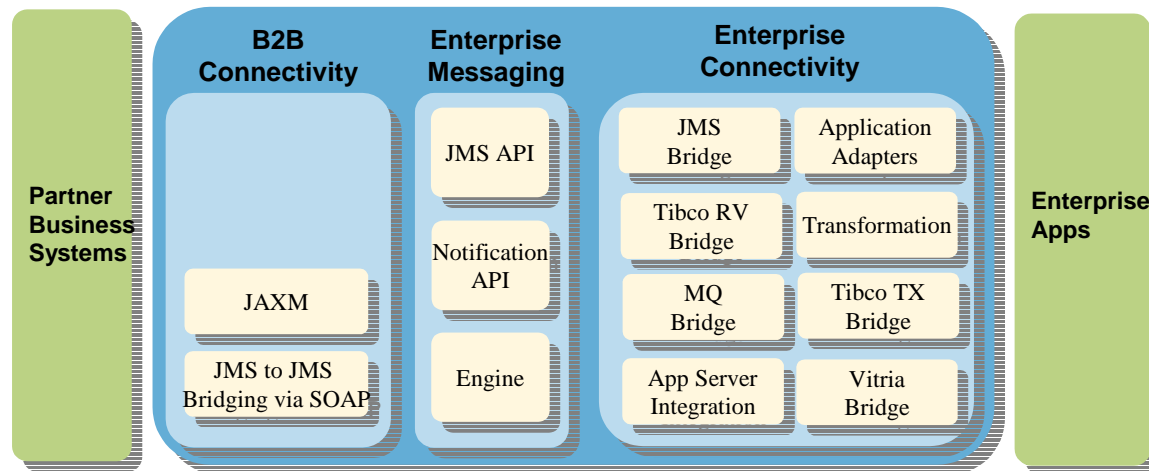
Web Services Integration Strategy (2)

▶ Approach

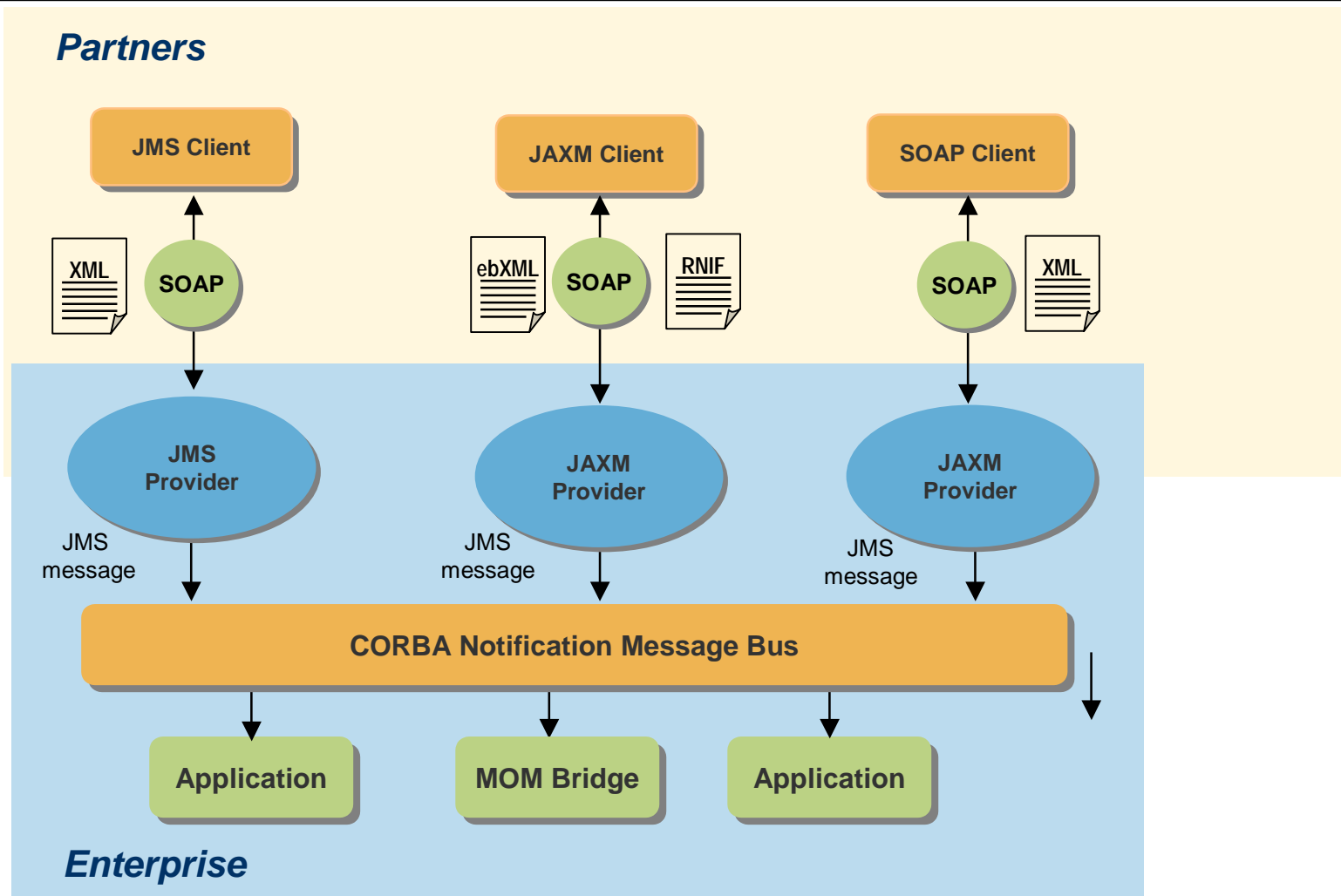
- ▶ JAXM to provide interoperable, SOAP-with-attachments web services integration
- ▶ Wider connectivity to messaging platform

▶ Enables

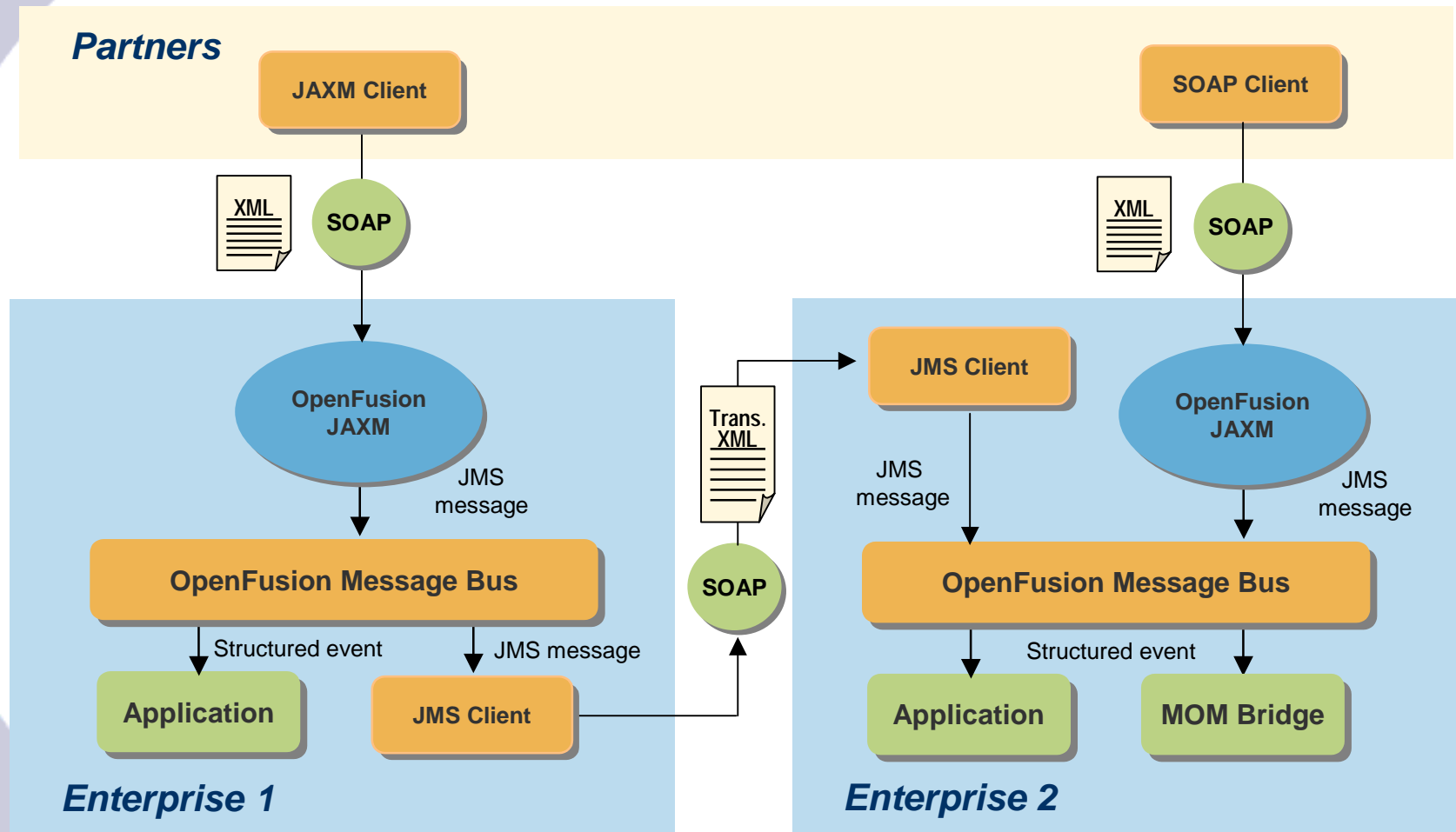
- ▶ More flexible integration with business partners over internet
 - ▶ Does not require the same technology at each end
 - ▶ Only relies on SOAP
- ▶ Example could be the integration of partners in telecoms service provisioning
 - ▶ DLEC, CLEC, Contractors, integrated though SOAP messaging



Heterogeneous Web Services Messaging



Heterogeneous Enterprise Integration using Web Services Messaging



Conclusion

- ▶ Messaging is an important model for Web Services communication in enterprises
- ▶ The CosNotification infrastructure provides a good backbone for Web Services integration
 - ▶ Provides the common glue that extends Web Services to CORBA, J2EE Apps, and third party MOM (MQ, Tibco)