

Web Services Technology Deployment Issues

Gerald W. Edgar & Pranab K. Baruah

IT Architecture & e-Business

Commercial Airplanes Group

The Boeing Company



Introduction

- Web Services technology offers a platform neutral approach for integrating applications to support business integration in two domains:
 - Enterprise Application Integration (EAI), and
 - Business-to-Business integration (B2Bi).
- EAI is the process of creating an integrated infrastructure for linking disparate systems, applications, and data sources within the corporate enterprise.
- B2Bi is the process of secured coordination of information among businesses and their information systems, enabling cross-enterprise business applications such as:
 - collaborative e-commerce,
 - supply chain management (SCM),
 - and customer relationship management (CRM).



Web Services Value Proposition

- The business value that Web Services technology brings is that services and their interfaces can be easily
 - described (WSDL, XML)
 - published (UDDI registry),
 - found (searching a registry),
 - bound statically or dynamically, and then
 - invoked with an internet protocol (HTTP, FTP, SOAP).
- But the hard part is in developing and implementing
 - New business services with J2EE, .Net, or CORBA and exposed as web services.
 - Leveraging existing IT investments (IMS, CICS, ...)

Risks for Deploying Web Services

Large enterprises perceive the following as risks:

- **Standards**

- Web Services standards are in flux or will undergo significant changes
- Increasing fragmentation in the standards efforts, tools, API and technology is highly likely resulting in Web Services interoperability only at the lowest level.
- Most vertical industry standards need to be adapted to Web Services.
- Conflicts between standards will need to be solved through standardized mappings or through custom mappings.

- **Security**

- **Operational Management**

- The performance of Web Services under various business scenarios is not known.
- The effect of Web Services deployment on network bandwidth is uncertain.

- **Development Guidelines**

- Guidance for the applicability of Web Services technology is non-existent.
- Integrated development tools are not yet available.



Key Issues for Business Integration

- How do I leverage my existing processes, infrastructures, applications and data?
- How do I meet my integration requirements with the least effort (time and money)
- What about security and privacy?
- How do I keep the employees trained on ever-changing requirements for new technologies and tools?
- How does this affect my customers and suppliers?
- How do I make my extended enterprise (customers, partners and suppliers) integrate their processes and systems with mine effortlessly?
- How do I manage access to my Web Services?
- How do I manage interactions of Web Services in production environment?



Business Integration Needs

- Integrating the front-end applications and data
 - web portals interacting with business applications to present a common view with little or no interactions between the applications.
 - need to integrate with common collaboration tools – email, instant messaging, discussion forums etc.
 - need to support the extended enterprise - employees, partners and suppliers.

- Integrating the back-end applications and data
 - usually with an Integration Broker (J2EE or .Net)
 - provides an end-to-end integration platform for EAI and B2Bi
 - supports the requirements for
 - Supply Chain Management
 - Collaborative Product Commerce
 - Customer Relationship Management
 - New revenue generating services for customers

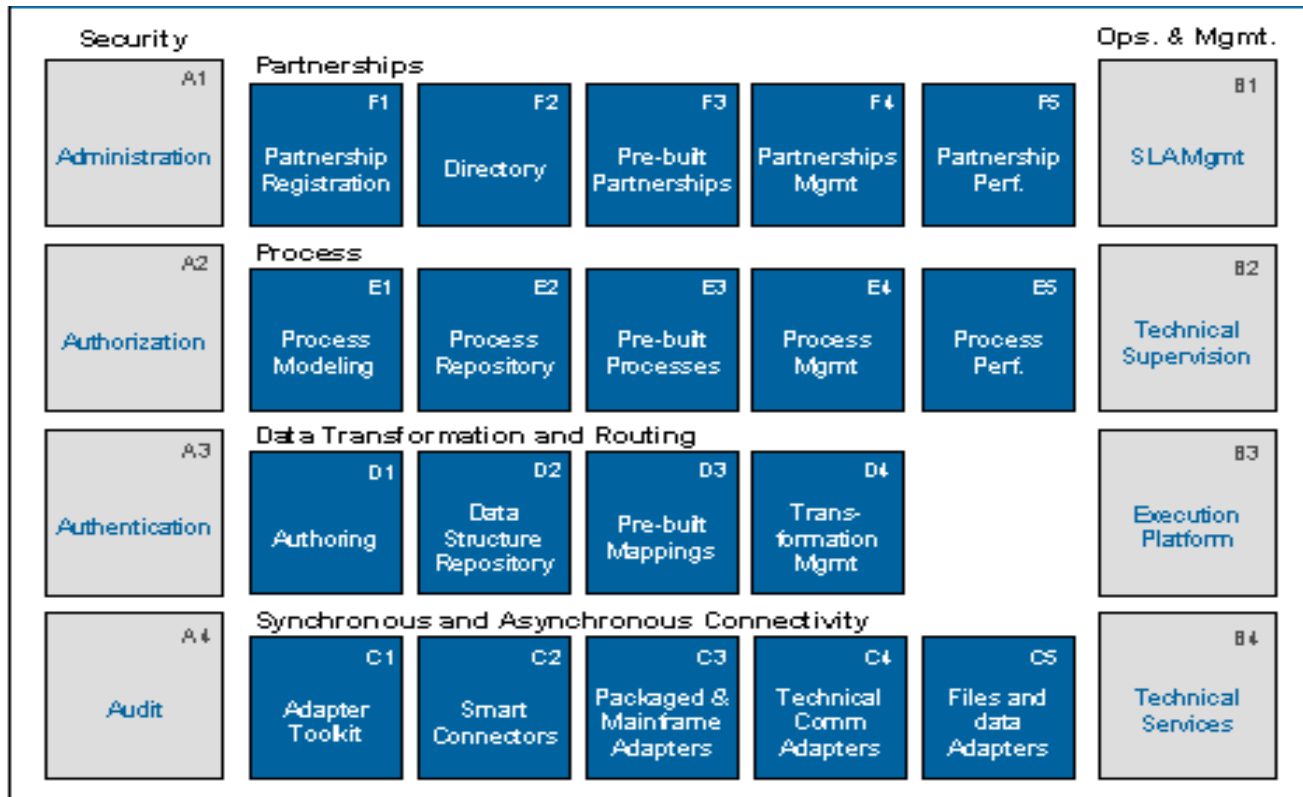


Integration Broker

- Integration Broker technologies
 - built on top of existing middleware technology, most often on messaging middleware, J2EE, CORBA, COM+ platforms.
 - provides both EAI and B2Bi capabilities - provides an end-to-end integration platform addressing the critical business components required to automate business processes across the extended enterprise, which includes the trading partners.
 - provides wide-ranging, pre-built application adapters, and bi-directional connectivity to multiple applications, including packaged and mainframe applications.
 - Examples: TIBCO, webMethods, IBM WebSphere Business Integrator, IONA E2A, BEA WebLogic, Microsoft BizTalk Server
- Beginning to be widely accepted in industry in solving complex integration problems
 - **It works!**
 - at low levels.
 - **Access to business services can be managed**
 - through common services being put in place.
 - **Interactions of services in production environment can be managed.**
 - by using the Broker



Application Integration Framework

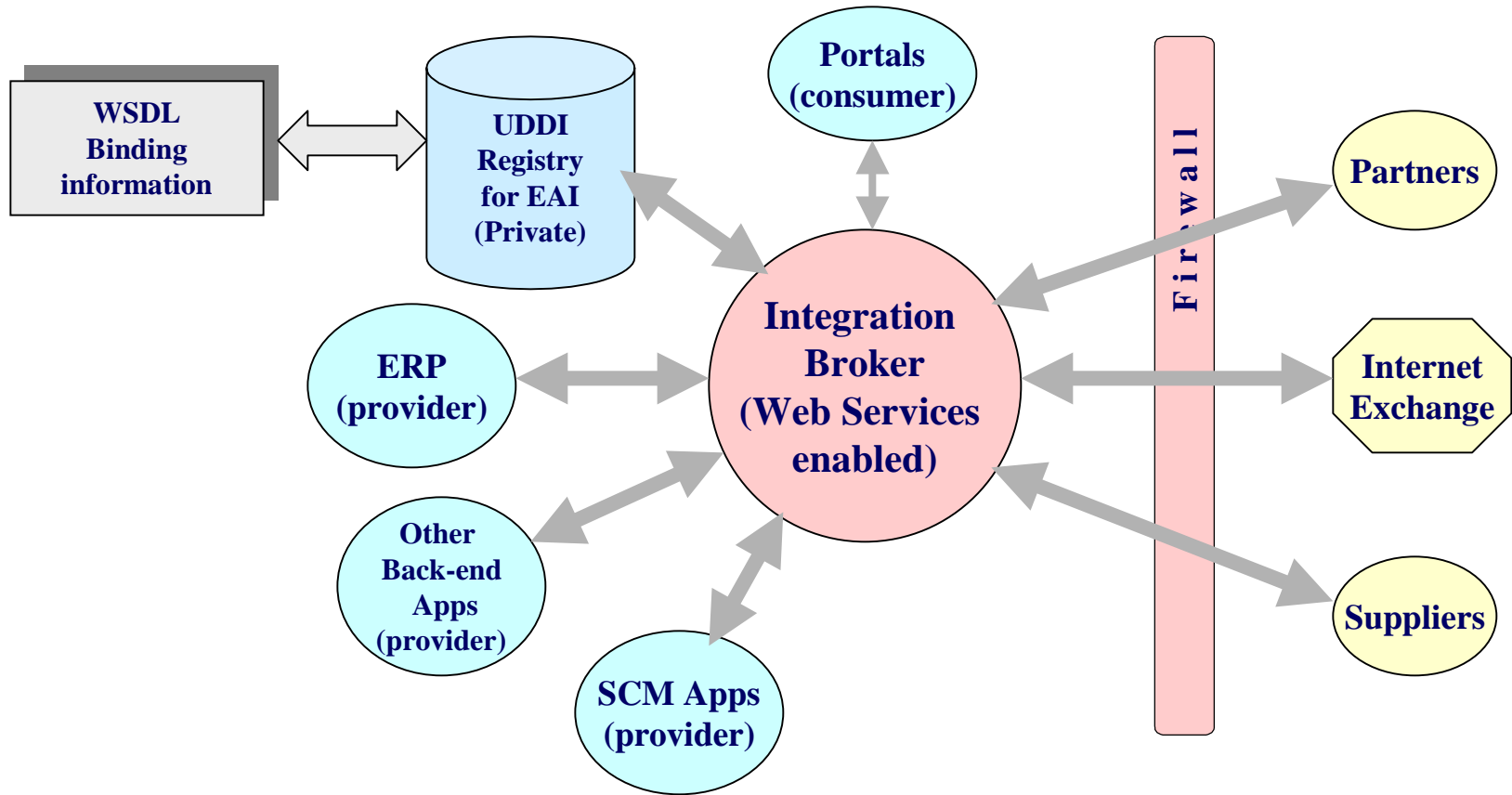


Source: Giga Information Group

- Implemented in most commercially available Integration Brokers products
- So, how and where does Web Services technology fit?
- Can I leverage my investments on Integration Brokers with Web Services?



A Web Services Broker?



Web Services Broker = Integration Broker PLUS

- Web Services support (UDDI, WSDL, SOAP)
- Registry
- Development, Deployment & Publishing Tools



Development Issues

- **What types of Web Services should be built**
 - Web Services to be consumed by other applications, applications that only consume other Web Services, or Web Services that both producers and consumers of Web Services request?
 - The more complex the Web Services are, the harder it will be to manage their interactions in production environment.
- **What level of Granularity**
 - The level of granularity at which it makes sense to build Web Services (i.e., a date routine Web Service, an inventory look-up Web Service or a general ledger Web Service).
 - The finer the level of granularity, the more complex the network of Web Services will become in production
 - Security issues to constrain vulnerabilities.
- **Synchronous vs. Asynchronous Web Services**
 - What criteria developers should use to determine when it makes sense to build synchronous Web Services and when it makes sense to build asynchronous Web Services.
 - Current one-way data transfers may be amenable to asynchronous services.
 - The more interdependent synchronous Web Services a company has in production, the greater the risk will be that a Web Services deadlock could occur in production.

Development Issues (continued)

■ Transaction Integrity

- How to design web services to ensure transactional integrity.
- How to handle both compensating and XA-style transactions.

■ Testing

- How to ensure that all Web Services that are for both producers and consumers can meet any and all service levels as defined for them in the target production environment.
- How to test for scalability of Web Services, especially the externally facing Web Services where usage loads may be unpredictable.

■ Architecture Patterns

- How to best architect, build, test and maintain Web Services.
- Possibly integrated in development and testing tools.

