Enterprise Integration Update

Adam Rybicki arybicki@unicon.net
Steve Wheat swheat@openii.com

December 6, 2005

UNICON

Agenda


I. Introduction
II. History
III. Current Trends and Standards
IV. The Effects of New Trends on Integration
V. Java Standards and Tools
VI. Conclusions and Recommendations

History

- Proprietary EAI solutions popular in mid-90's
- Their value proposition: provide pre-built connectors to most common enterprise systems
- Custom connectors expensive to build
- Locked to specific programming language (often C or C++)
- Security, if any, seemed like an after-thought

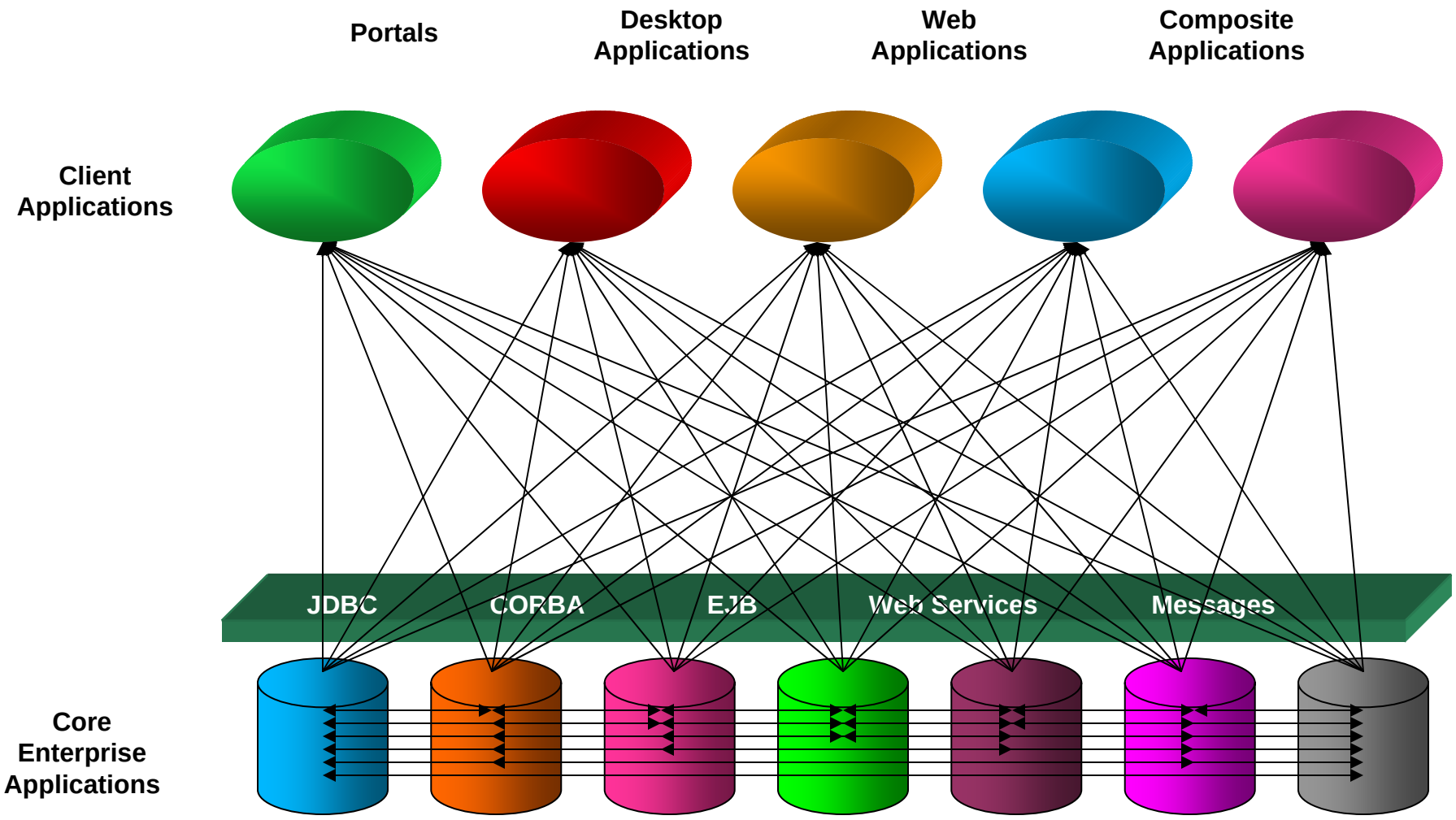
Enterprise Integration Challenges

- Communication
- Connectivity
- Transformation
- Service-Oriented Architecture
- Portability
- Security

Examples of Applicable Standards

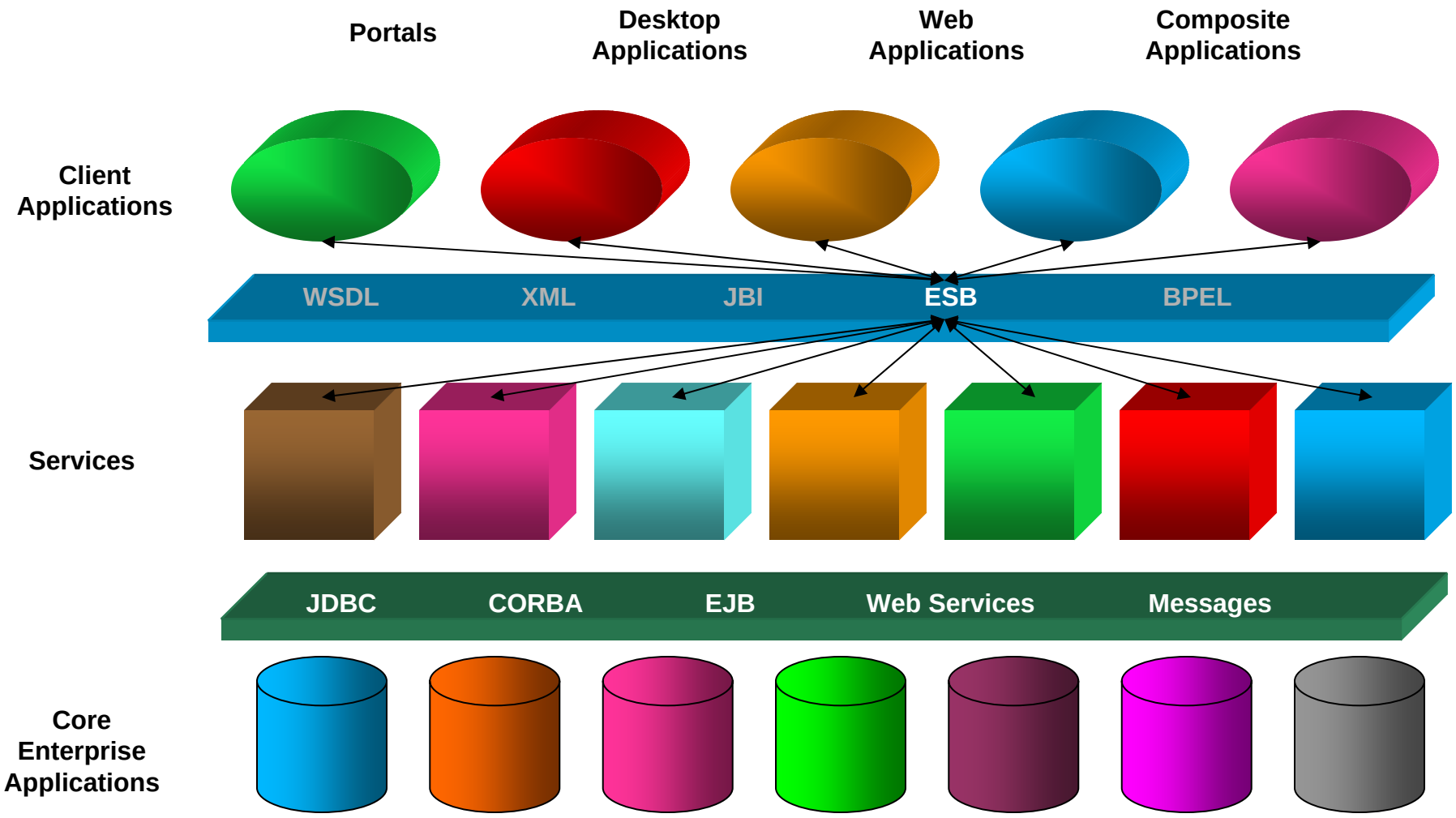
- Communication: Messaging / JMS
- Connectivity: Web Services: XML, SOAP, WSDL, UDDI
- Transformation: XSLT, Xquery
- Service-Oriented Architecture (SOA): based of the above standards; loosely-coupled
- Portability: Java, HTTP, XML, SOAP on Windows, Unix, mainframes
- Security: SSL, certificates, signatures, SAML plus all of the WS-* security standards

Point-to-Point Integrations



UNICON

Service Oriented Architecture



December 6, 2005

UNICON

Enterprise Service Bus

- Not necessarily a message bus
- Evolved from MOM, Integration Brokers, Application Servers, XML, JMS, E-Commerce, and Web Services
- ESB provides the backbone for building an enterprise SOA-based integration environment
- ESB is not just an abstract pattern. It is a product category with a distinct definition and many vendor offerings.

New Java-Specific Standards

- EJB 3.0
- JBI (JSR 208)
- BPELJ
- JDBC 4.0

EJB 3.0

- Simplifies the EJB programming model
- Eliminates the requirements for Home, Remote interfaces and deployment descriptor
- Allow transaction demarkation with annotations
- Remove the requirement to implement boilerplate methods (ejbCreate(), ejbActicate(), ejbRemove(), ...)
- Continue support for 2.1 and older

JBI (JSR 208)

- Java Business Integration
- Standardizes the Enterprise Application Integration APIs
- Prevents vendor lock-in with EAI and B2B applications
- Helps with building SOAs
- Too new to know its potential for success, but it has gained traction

JDBC 4.0

- 4.0 is labeled as an “Ease of Development” release
- Can use Generics and Annotations
- SQL 2003-compliant XML data type
- Connection.isValid()
- Connection and Statement pooling now a part of the standard
- Specific SQL exceptions: transient and non-transient (transient might succeed if retried)
  - Non-transient: SQLSyntaxErrorException, SQLInvalidAuthorizationSpecException, SQLIntegrityConstraintViolationException, SQLDataException, SQLNonTransientConnectionException
  - Transient:SQLTimeoutException, SQLTransactionRollbackException, SQLTransientConnectionException

Java Tools

- Java 5 needed for all standards that require annotations
- EJBs are much simpler to write:

```
import javax.ejb.*;

/**
* A stateless session bean requesting that a remote business
* interface be generated for it.
*/
@Stateless
@Remote
public class HelloWorldBean {
   public String sayHello() {
      return "Hello World!!!";
   }
}
```

For more information, read this article: http://www.javaworld.com/javaworld/jw-08-2004/jw-0809-ejb_p.html

JBI

- Reference Implementation: JBI SDK 1.0
  - **SunSequencingEngine**
    SunSequencngEngine works by accepting an input message and sequentially invoking a series of services. The list of services is described in an XML syntax created by an external tool and deployed to the engine. When the engine receives a service request, it locates the service list corresponding to the service and executes each service in the service list.
  - **SunTransformationEngine**
    SunTransformationEngine allows transformation of an XML document input in one format to another format. The transformation is based on a specified style sheet.
  - **SunSOAPBinding**
    SunSOAPBinding allows external web services to invoke services in the JBI environment and vice versa. An external web service is likely to use a different message format and/or network protocol from an internal one. These service interactions are made possible via SOAP bindings and WSDL files.
  - **SunFileBinding**
    SunFileBinding allows services to communicate with the local file system in which JBI SDK 1.0 is installed.

JBI Implementations

- JBI implementations are ESBs
- Commercial ESB implementations have been slow to implement JBI
- Open source JBI implementations exist today:
  - Open-ESB is available from java.net: http://open-esb.dev.java.net/
  - Iona started Celtix: http://celtix.objectweb.org/
  - LogicBlaze started ServiceMix: http://servicemix.org/Home
  - Mule: http://mule.codehaus.org/

Java Tools

- Apache Axis provides a rudimentary WS toolkit
- Sun has JAX-RPC
- The University of Colorado has designed and is implementing an SOA
  - Based on WSDL standard
  - Using Glue from webMethods (now replaced with the next generation product called Fabric)
  - Glue simplifies WS programming greatly:
    - no SOAP, WSDL, or XML
    - POJO to WSDL and WSDL to POJO conversions
    - Includes a SOAP server

Java Tools

- University of Illinois developed OpenEAI.
- It is an ESB that facilitates implementing SOA.
- Focuses on documenting interfaces and information exchange among enterprise applications.
- Key differentiator: individual enterprise applications only integrate with the ESB, not with each other.
- Currently JMS-based, but adding WS interface for a future release.
- There is a commercial company offering support (not Unicon). ☺
- Messaging protocol and message format in XML. Examples:
  - Person
    - Query
    - Create
    - Generate
    - Delete
    - Update
  - Course
    - Query
    - Create
    - Generate
    - Delete
    - Update

Enterprise Integration Summary

- Enterprise integration is not synonymous with Web application integration
- Service-Oriented Architectures (SOA)
  - Usually Web Services
  - Easily Accessible
  - Interface Definitions
  - Discoverable
- SOA can be implemented ad-hoc (ala University of Colorado) or using an ESB product like OpenEAI (ala University of Illinois)
- Standards should make enterprise integration easier
- There are too many standards to choose from

Conclusions and Recommendations

- The first priority should be to access enterprise data from the authoritative data source rather than from a replica. This does not mean directly reading from that data source, but rather through an ESB.
- Integration platform should be independent of the ERP platform vendor.
- Only platform-independent, standards-based, and vendor-agnostic solutions should be considered.
- While an EJB-based SOA is still an SOA, it is too difficult to access from .NET or other non-Java applications.
- Web services based on SOAP and WSDL are proven while JBI, for example, is not.
- Message-based ESB may be OK, but there should be a WS interface to it, too.