Legacy System Integration Using XML and WebSphere

SESSION

NUMBER PE418811

SPEAKER Raju Mukherjee WebSphere Architect
George Kosmides Vice President
Dennis Pagano President

TITLE

AFFILIATION Noospherics Technologies Inc

information@noospherics.com

Agenda

A. ***Issues In Integrating Legacy Applications with New eBusiness initiatives***

B. Integrated View of Enterprise

   - Common Data Format – XML

C. WebSphere: Center of Enterprise XML Architecture

   Design Guidelines for XML-Based Architecture

D. Web Services: Emerging Integration Technology

   - Legacy App Wrapped As Web Service

E. Summary

Terminology – What is a Legacy System?

- Answer: Any system that is running in production within your company

- As soon as it's delivered, it's a "legacy" system
- One person's "legacy system" is another person's "corporate IT asset"

Issues In Integrating Legacy Applications with New eBusiness initiatives

- New e-business systems need to be flexible and robust
- Most enterprises have data and business rules embedded in existing legacy systems
- E-business systems must work smoothly with a variety of legacy systems

The New Systems

- New e-business systems (when properly designed) can provide business solutions that are flexible
  - Support and embrace change
  - Accommodate business rules changes
  - Accommodate changing data formats
  - Easily adapt to new presentation technologies
  - Need to access different platforms

The New Challenges

- Making new systems cooperate with one or more legacy environments
- Need to access different data models and formats
- Integrate new systems with legacy platforms in a way that is seamless
- Need to facilitate easy access to legacy data

Enterprise Integration – What a Difference 10 Years Makes

- Enterprise 1990
  - Stovepipes
  - Not much interaction between applications
- Enterprise 2001
  - Provide one integrated view of enterprise
  - Interaction with multiple legacy systems and platforms
  - B2B – needs access to multiple back-end data
  - The barriers are disappearing!!

Typical Enterprise

- Numerous legacy applications:
  - Different hardware platforms
  - Different execution environments
  - Different development tools
  - Different data formats
- No Integration or only Point-To-Point Integration

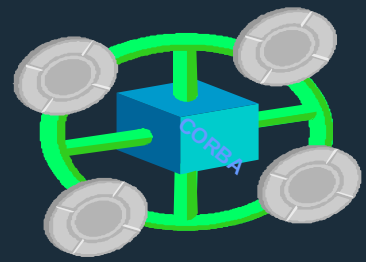
What are the Common Architecture Requirements?

- There are multiple interfaces for calling the service
- One Business Service (e.g.Mortgage Rating) requires execution of other internal and external services
    - E.g Credit, Title, Financials
- External Process – Support B2B transaction with external vendor
- Internal Process - orchestrate execution of existing IT resources

Enterprise Application Integration - EAI

- Enterprise has a variety of disparate systems
  - ERP, CRM, Mainframe, Financials, Custom Apps
- These systems need to be effectively integrated
- When integrated, they can support newer application platforms:
  - Web application servers
  - B2B exchange systems

EAI Basics



- EAI is typically built on **message-brokering frameworks**
- Connectors or **adapters** plug-in to the brokering framework
- Provides for communication among disparate systems at the **data, method** and **application level**
- Minimizes time and effort required to enact new business processes

EAI – Types of Integration

- Data level integration
- Method level integration
- Application-to-application integration
  - Application-to-application communication
  - Data translation and transformation
  - Business processes spanning the disparate systems

Agenda

A. Issues In Integrating Legacy Applications with New eBusiness initiatives

B. Integrated View of Enterprise

- ***Common Data Format – XML***

C. WebSphere: Center of Enterprise XML Architecture

Design Guidelines for XML-Based Architecture

D. Web Services: Emerging Integration Technology

- Legacy App Wrapped As Web Service

E. Summary

Diverse Systems, Common Data Format

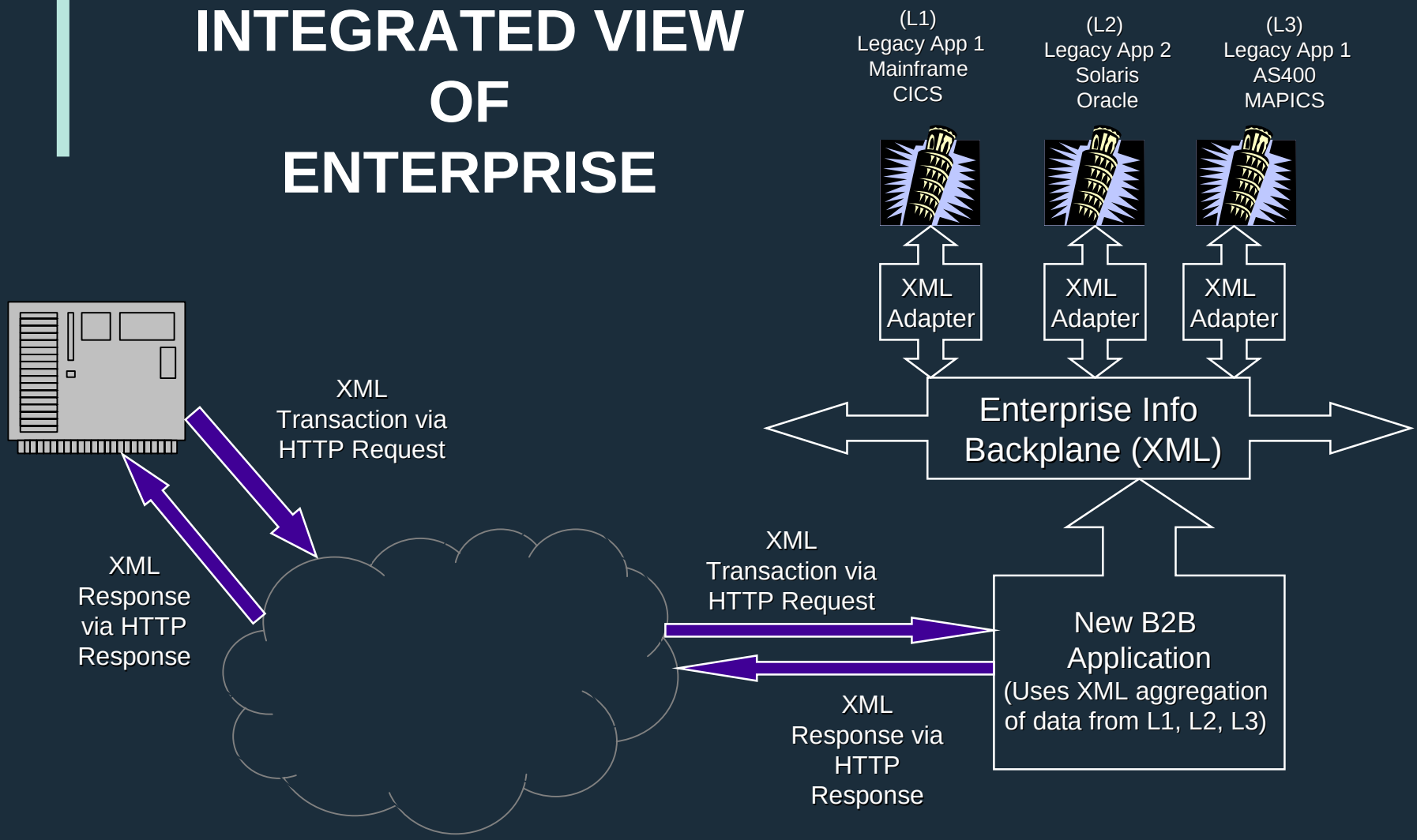
An Enterprise has:

- a variety of data formats and data semantics BUT
- one common metamodel of the meaning of the business data
- This can be expressed in XML
- This data format can be used AMONG systems in the enterprise
- This data format can be used BETWEEN partners in a B2B world

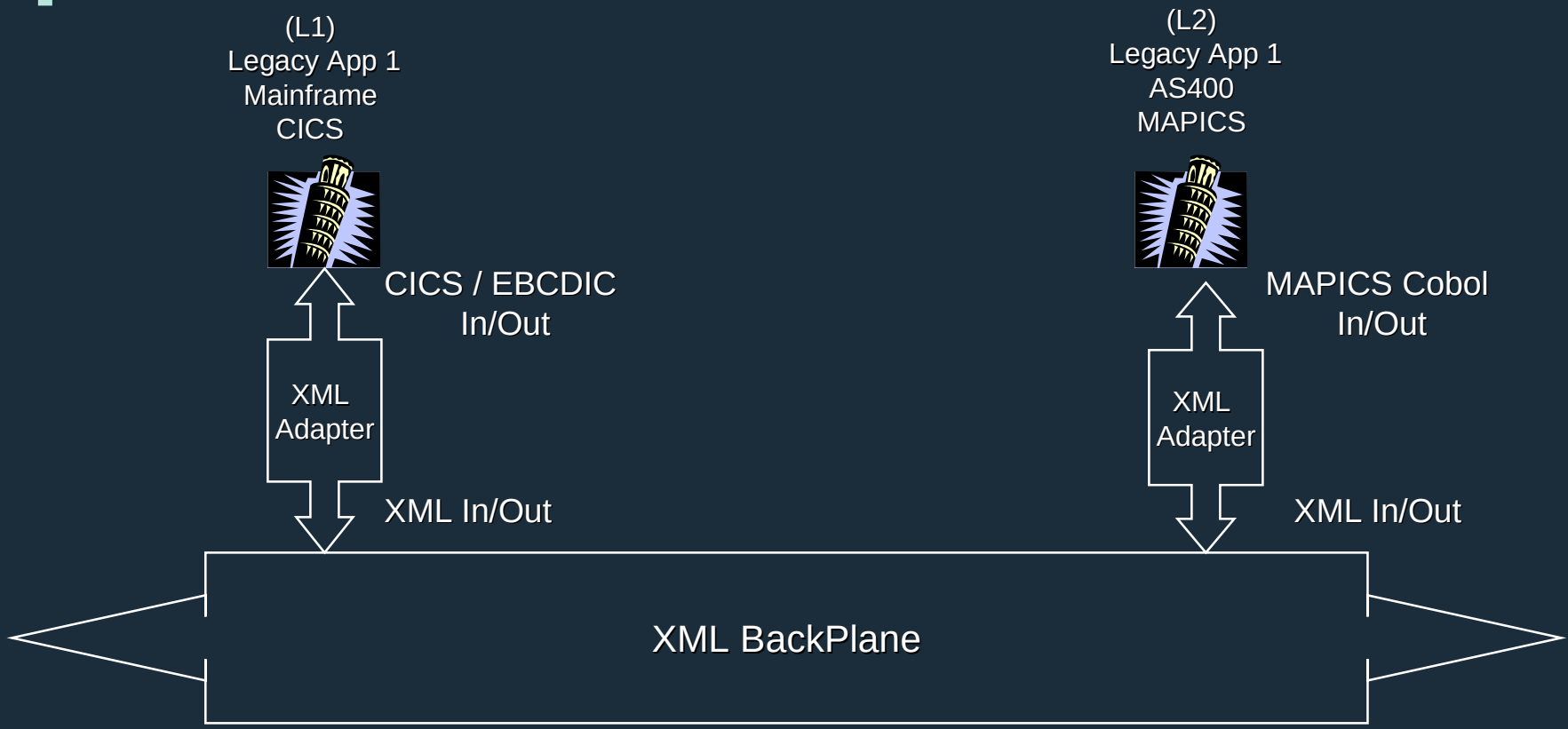
The Solution: XML Backplane for Integration

- Varied data from separate Legacy Systems feed into common enterprise XML data model
- Common model shared by multiple new enterprise and B2B applications
- Use Publish/Subscribe Mechanism to share data among applications
- IBM MQSeries Integrator can serve as this XML BackPlane

INTEGRATED VIEW OF ENTERPRISE



Common Data Format: XML



Agenda

A. Issues In Integrating Legacy Applications with New eBusiness initiatives

B. Integrated View of Enterprise

- Common Data Format – XML

C. WebSphere: Center of Enterprise XML Architecture

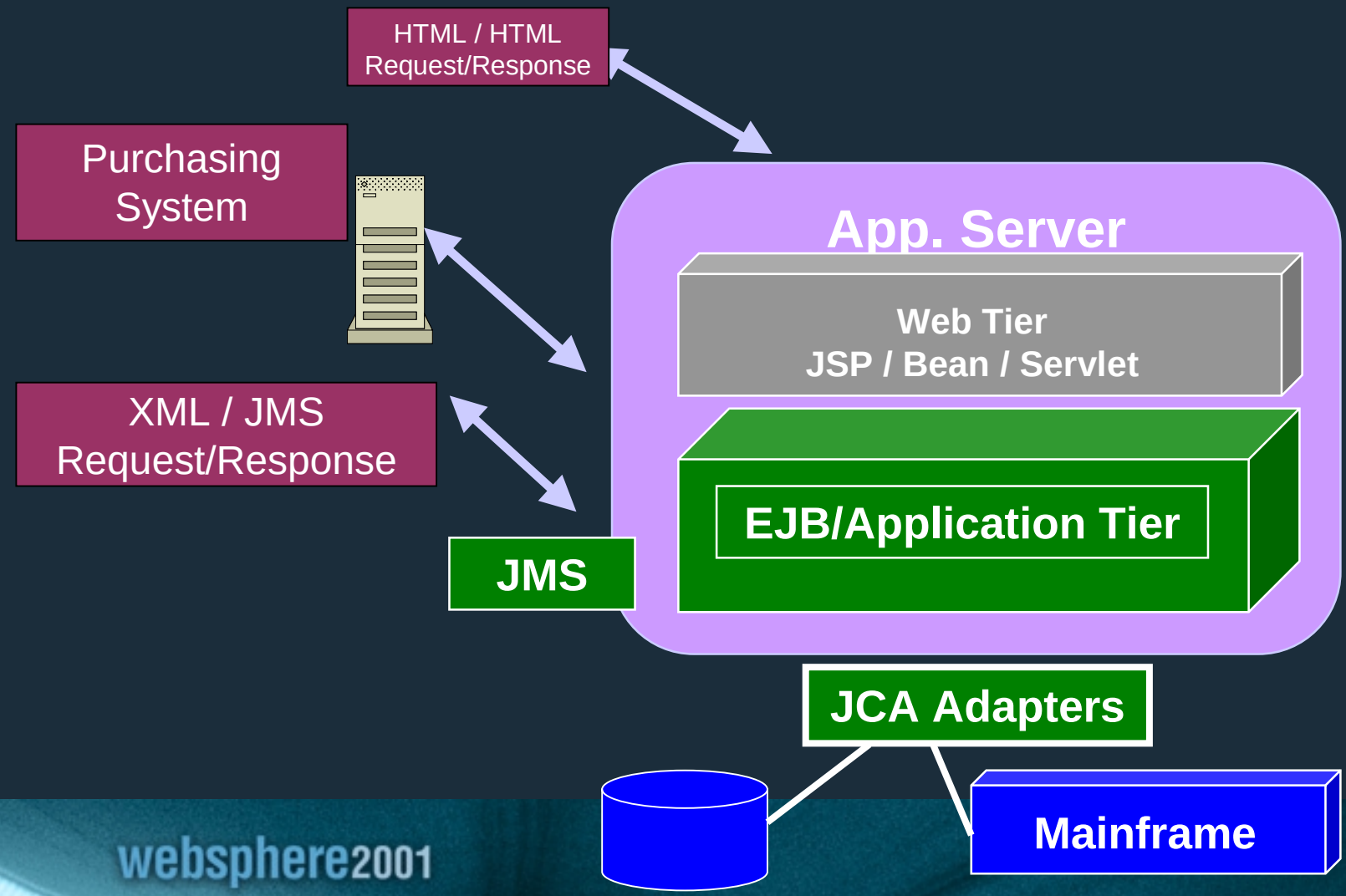
Design Guidelines for XML-Based Architecture

D. Web Services: Emerging Integration Technology

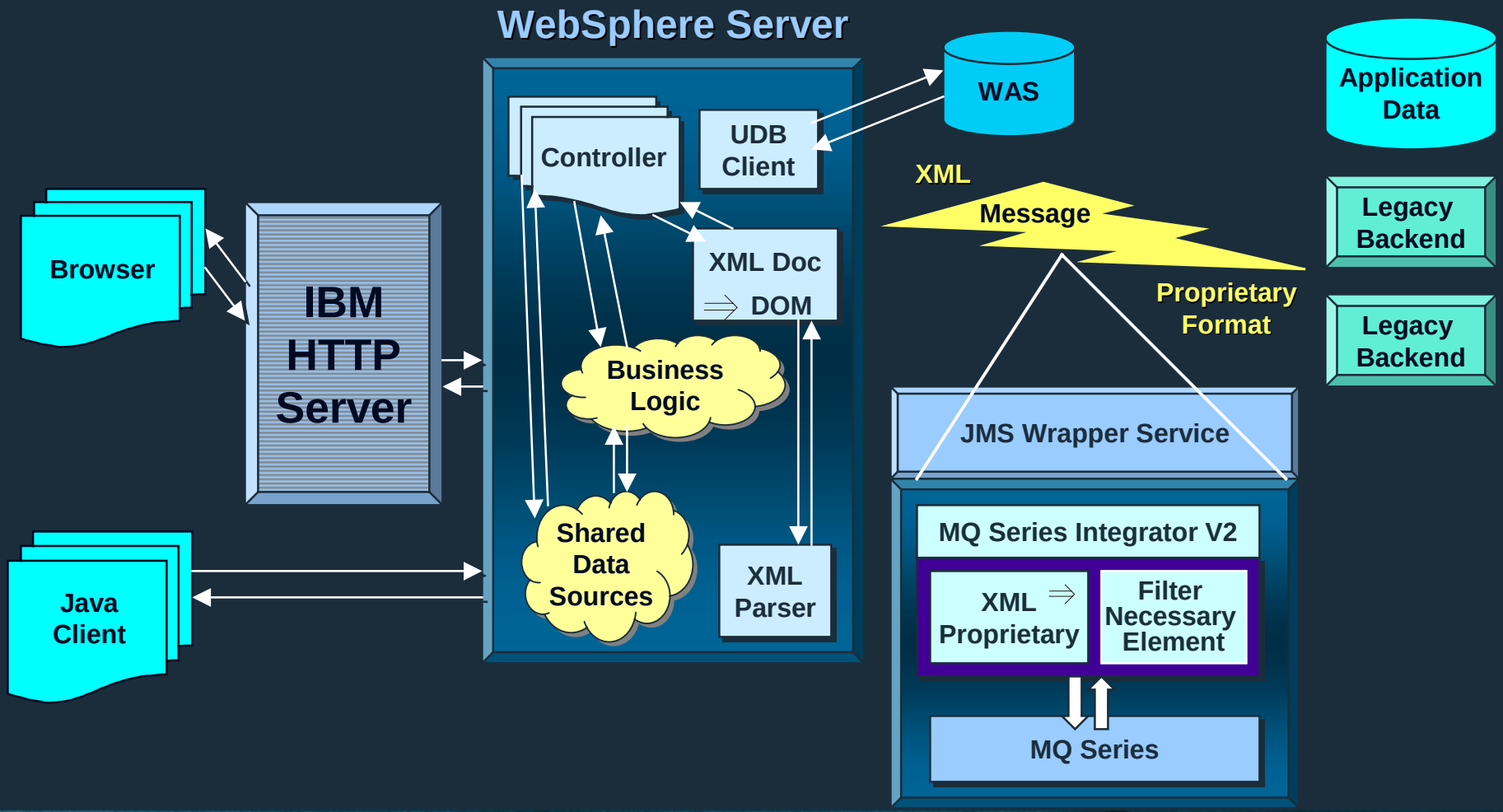
- Legacy App Wrapped As Web Service

E. Summary

WebSphere Application Server and J2EE Integration



WebSphere: Center of Enterprise XML Architecture



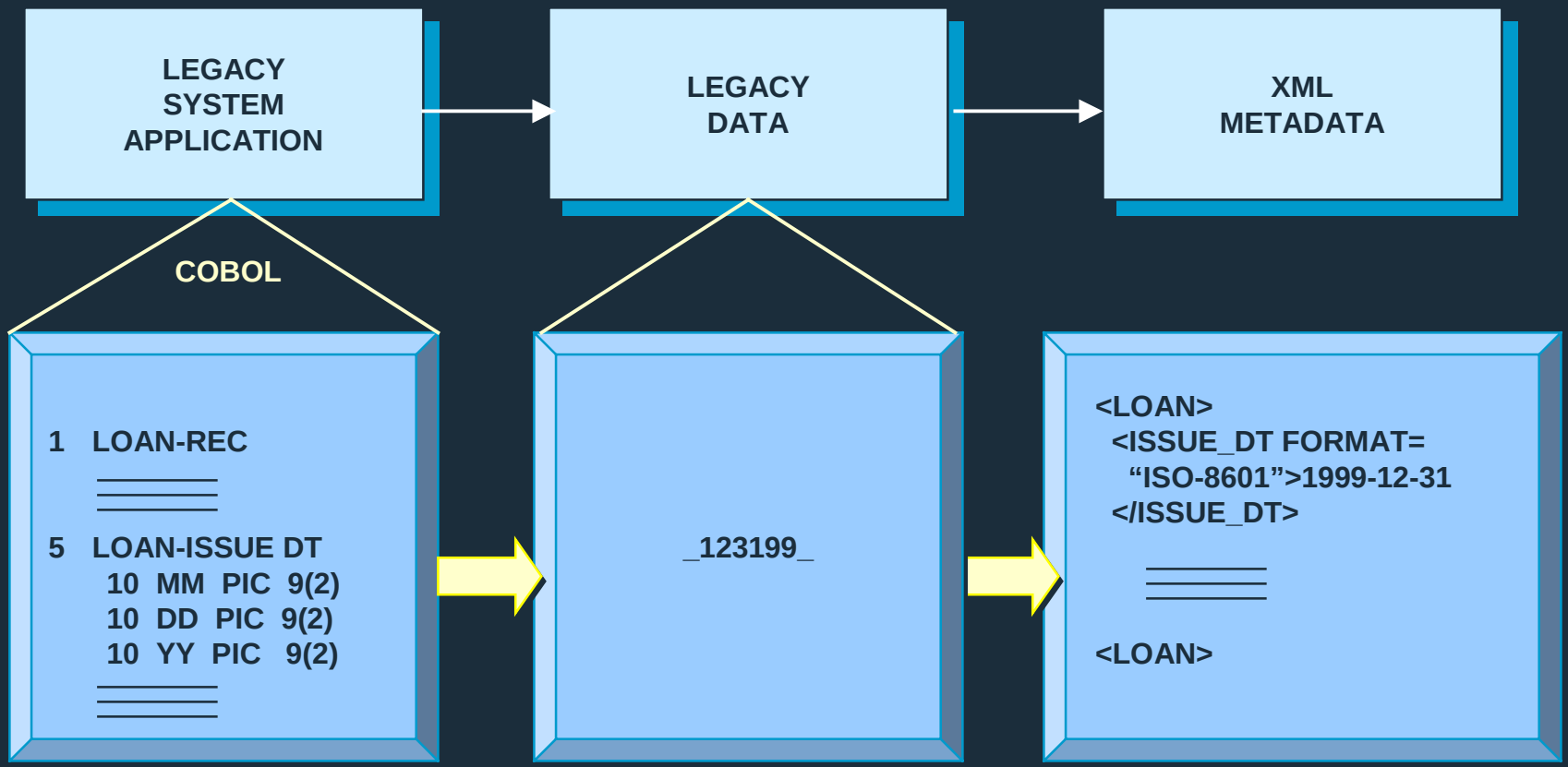
Design Guidelines For XML-Based Architecture

1. Transform persistent “static” legacy data into standard format.
2. Create common enterprise data model in XML
3. Layered, XML-Based Enterprise Architecture

Design Guideline 1: Transform Persistent Legacy Data into Standard Format

- Convert Legacy Data sources (files, DBs, etc) to XML
  - When data is static – just data, no legacy application
  - XML Document defines the data and its structure
  - XML Repository for all data

Converting Legacy Data to Common XML Data Format



Design Guideline 2: Create Common Enterprise Data Model in XML

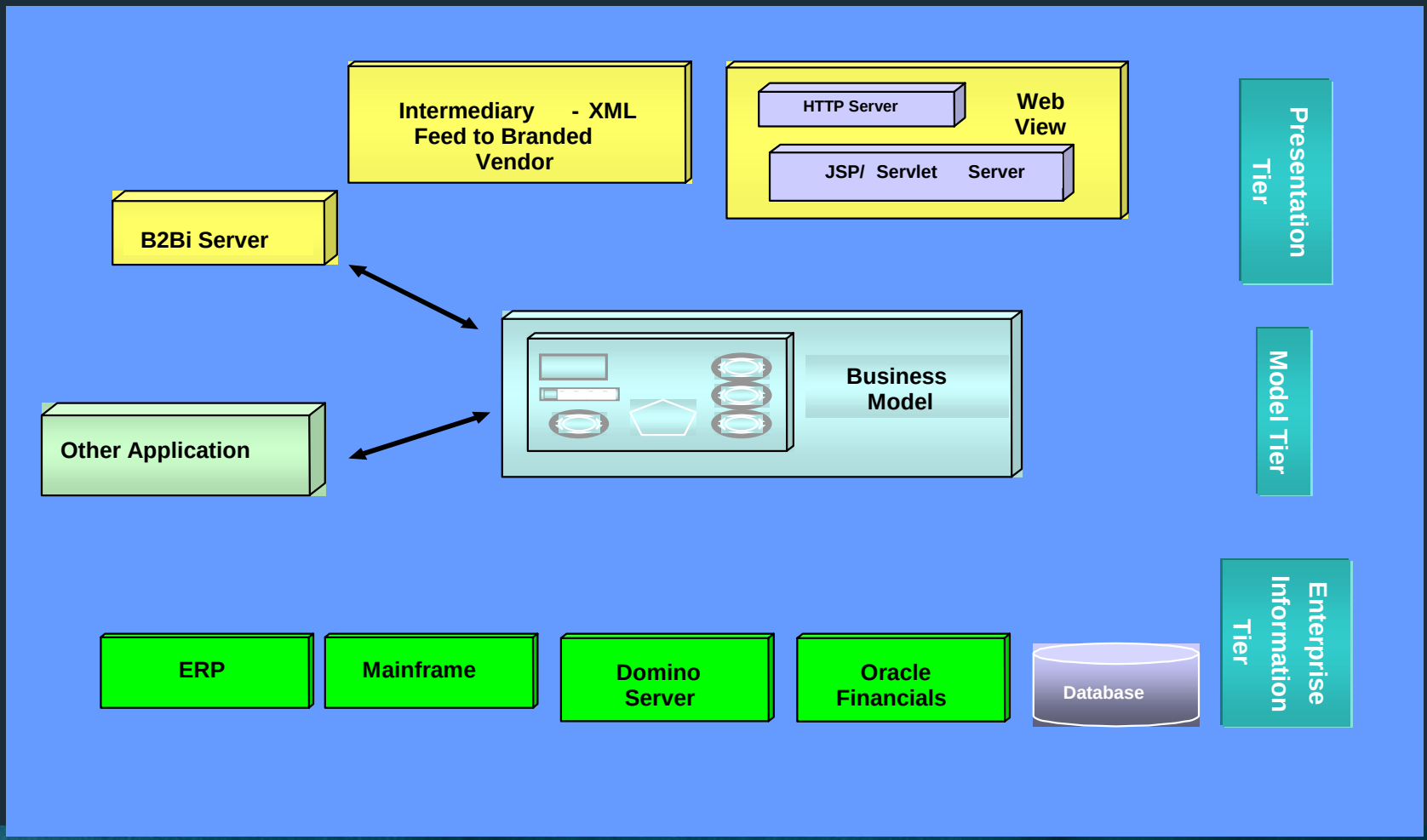
- Legacy Application input and output in the form of XML
- Common Data Model provides integrated view of the Enterprise
- Represented in Lingua Franca of XML so that all IT resources can access the data.
- Supports EAI within an organization
- Supports Business-to-Business Integration (B2Bi) between organizations

Design Guideline 3: Create Layered, XML-Based Architecture

Decoupled layers

- User interface – Presentation Tier
  - Interactive users - Web users, customers, agents
  - Batch users - bulk orders, portal interface, branded intermediary
  - Other systems - collaborative exchange networks
- Model Tier
  - Business model is key strategic asset
  - Technology independent
  - Can support many View layers
- Service Tier (Enterprise Information System Tier)
  - Interfaces to variety of internal systems - mainframes, ERP
  - Interfaces to variety of external systems - credit services, etc

Layered Architectures



Design Patterns For Layered Architectures

- Patterns enable rapid change
  - **Composite** pattern – treat the complexities as one large-grain object
  - **Mediator** pattern – quickly switch between various tier models

Layered Architecture – Strengths

- Layered metamodel assures that one layer can change without affecting other layers
- Architecture is better able to handle and embrace change
  - Metamodel is not based on any one technology, platform or bandwidth
- Architecture is better able to rapidly respond, adjust, respond to changes in
  - User interface technologies
  - Business model
  - Services

Agenda

A. Issues In Integrating Legacy Applications with New eBusiness initiatives

B. Integrated View of Enterprise

   - Common Data Format – XML

C. WebSphere: Center of Enterprise XML Architecture

   Design Guidelines for XML-Based Architecture

D. Web Services : Emerging Integration Technology

   - ***Legacy App Wrapped As Web Service***

E. Summary

Emerging Integration Technology : Web Services

The Concept:

- Wrap enterprise applications as services
- Standard service interface
- Standard Directory service for:
  - registering applications as services
  - finding desired service
- Standard mechanism for requesting execution of a service
- Recursive definition of services:
  - One service may be composed of sequenced execution of other services

Web Services Standards

UDDI Registry - Universal Description, Discovery, and Integration

- Common Registry for finding services

WSDL - Web Service Description Language

- Standard mechanism for describing service interface

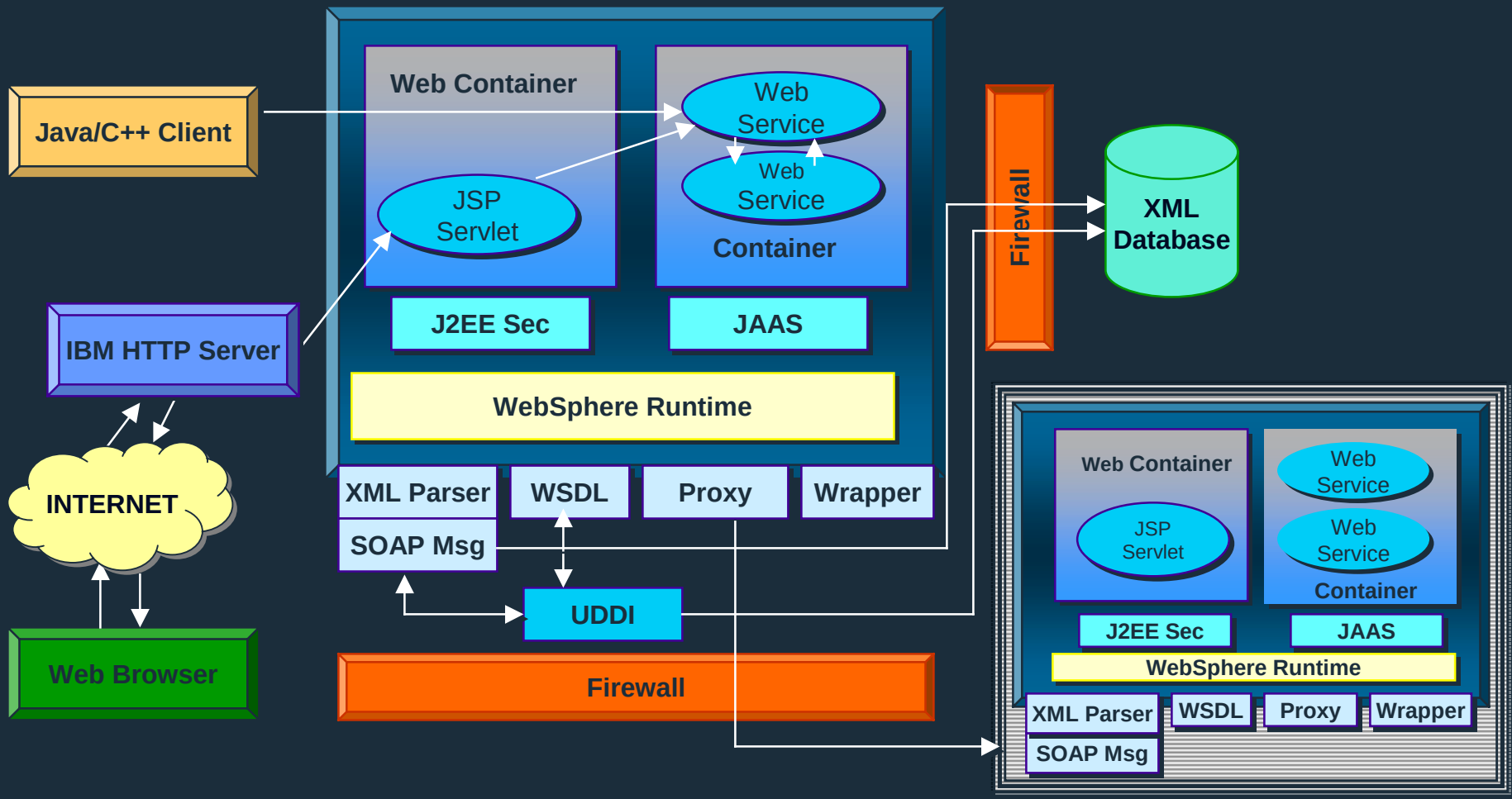
SOAP – Simple Object Access Protocol

- Standard protocol for accessing web service across Internet

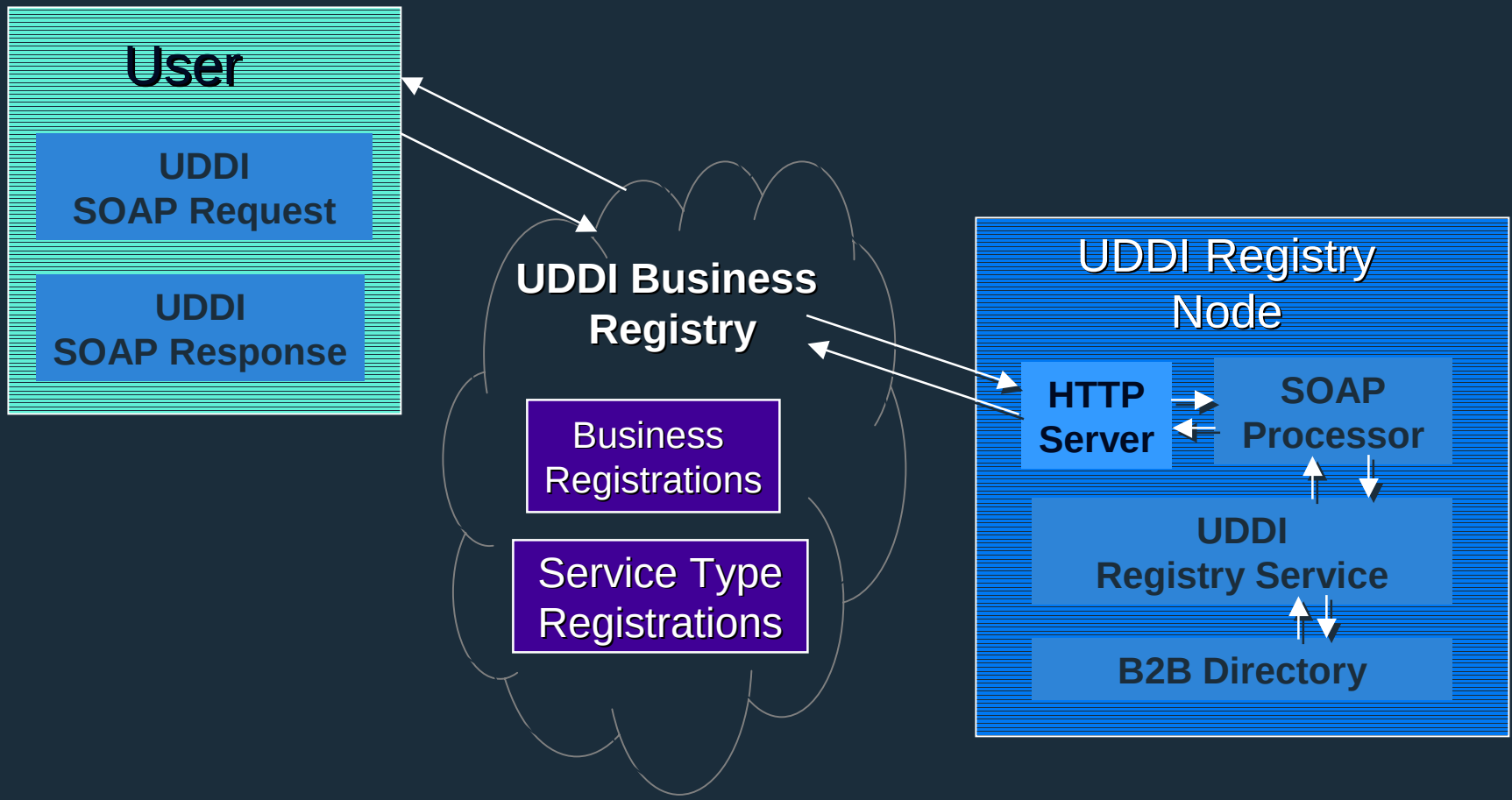
IBM Web Services Tool Kit

- available now on AlphaWorks
- Soon will be released as IBM product

WebSphere As Web Services Platform



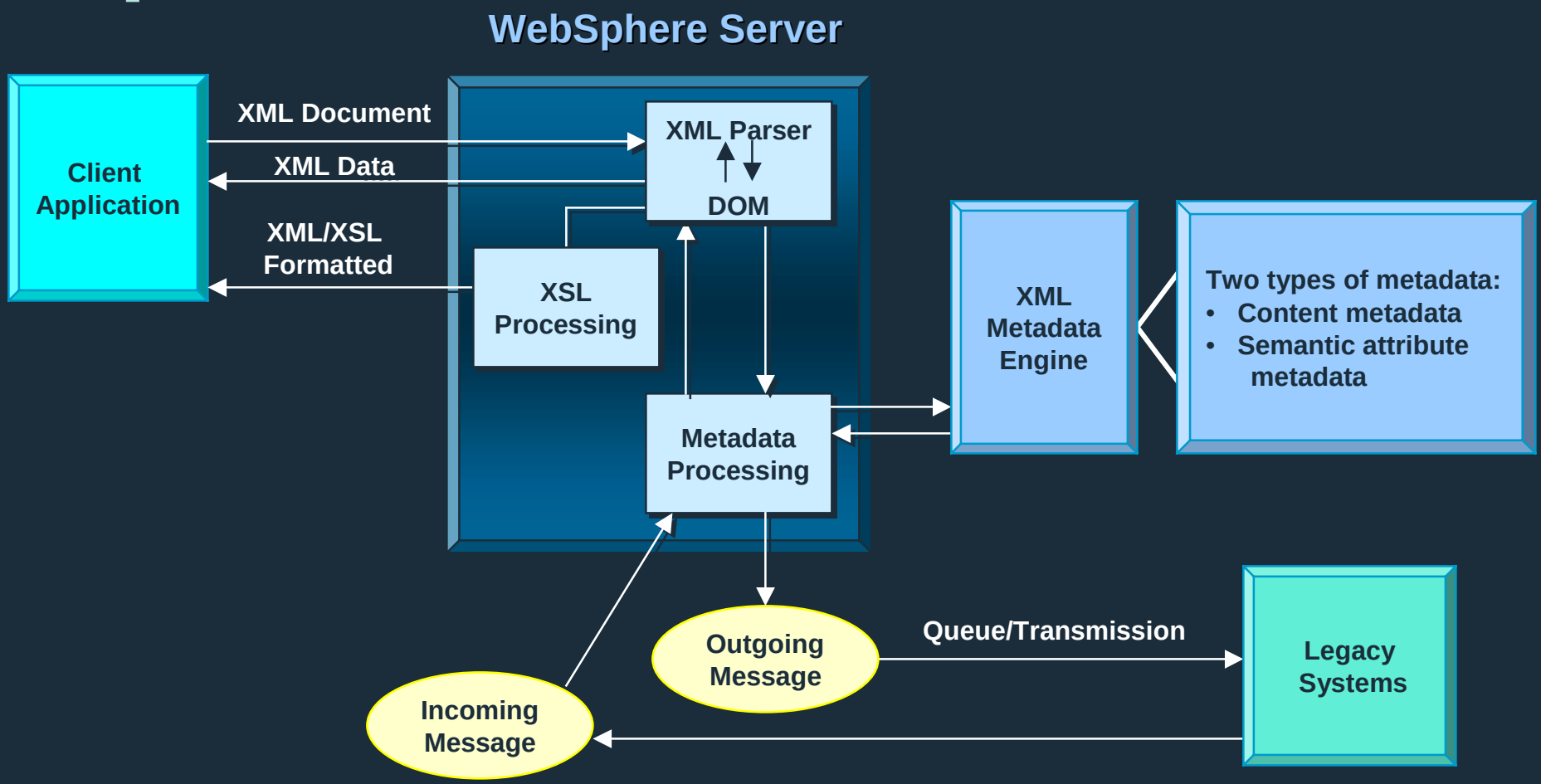
UDDI ARCHITECTURE



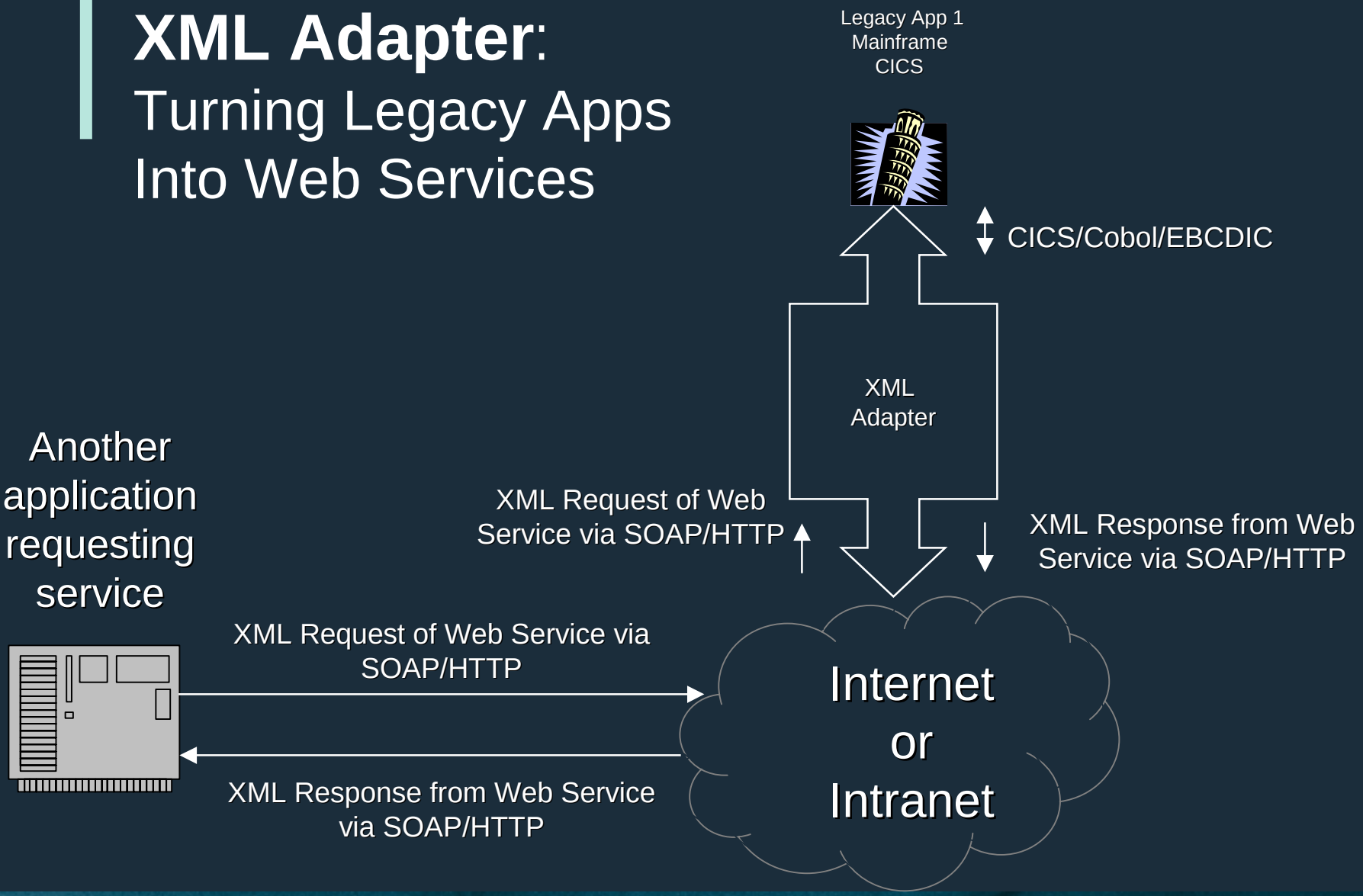
Wrap Legacy Application As a Web Service

- XML Request Document
  - Identifies Legacy App function and parameters
- XML Response Document
  - Contains results of executing the legacy transaction
- Common protocol (HTTP / SOAP) to share data

XML Transaction Framework



XML Adapter: Turning Legacy Apps Into Web Services



XML Backplane Supporting Enterprise Portal

- Use XML for Content Management and Delivery
- XML feeds WebSphere Portal Server
- Provides single point of access to all the Legacy information

Agenda

A. Issues In Integrating Legacy Applications with New eBusiness initiatives

B. Integrated View of Enterprise

- Common Data Format – XML

C. WebSphere: Center of Enterprise XML Architecture

- Design Guidelines for XML-Based Architecture

D. Web Services: Emerging Integration Technology

- Legacy App Wrapped As Web Service

E. Summary

Summary

- XML provides us with a tool to make architectures more flexible
- Not language dependent
- Not platform dependent
- Not technology dependent
- Web Services – Emerging standard for B2B and EAI Integration

More Information

Visit www.noospherics.com

- Updated slides with additional architecture diagrams
- White Paper elaborating on this presentation

Contact Noospherics Technologies for:

- Architecture, Design, and Development Services
- Help building complex Enterprise Systems

Raju Mukherjee - **raju@noospherics.com**

George Kosmides - **gkosmides@noospherics.com**

Dennis Pagano - **dpagano@noospherics.com**

(**410) 517- 3100 www.noospherics.com**