Next-Generation Java™ Web Technologies in Practice: The Document Services Platform

Charlton Barreto

Senior Computer Scientist/Architect
Adobe Systems, Inc.
http://www.adobe.com/products/livecycle/

TS-8844

Goal

Learn how to architect and build enterprise systems using Java™ Web Services and SOA technologies.

Agenda

Motivation

Technologies

Relating the Tiers

Document Service Bus

Applications

Composite Application Scenario

- Geographically distributed manufacturing operation
- Global partner for multiple sectors
- JIT delivery of product by stage
  1. Base components
  2. Assembled components
  3. Finished goods
  4. Distribution
- Guaranteed uninterrupted operations required

What Web Services + SOA Achieve

- Ease of integration
- Extensibility
- Interoperability
- Machine-to-machine communications
- Scalability

…Translating Into an Architecture

- XML-based messages
  - For platform-independent machine processing
- Framework focused on extensibility
  - Targeting interoperability
- Machine-processable descriptions
  - Facilitating integration
- Loose coupling
  - Maximising scalability

Agenda

Motivation for Web Services and SOA

Technologies

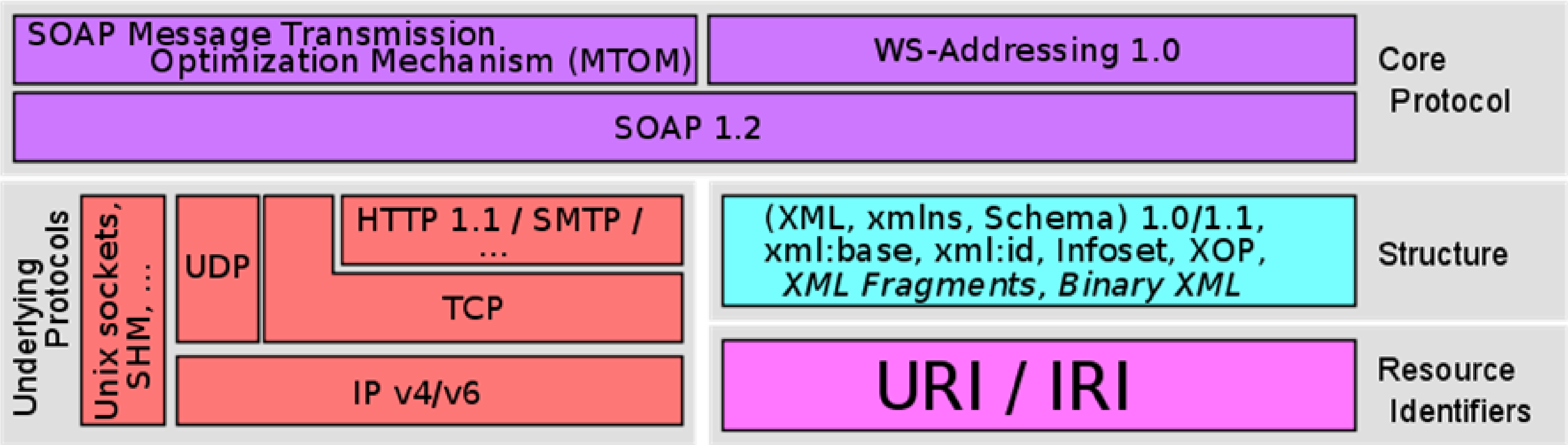
Relating the Tiers

Document Service Bus

Applications

How Do We Handle Messaging

- How do we handle messaging between activities at each centre
- Web services messaging framework



SOAP Message Transmission Optimization Mechanism (MTOM)

WS-Addressing 1.0

SOAP 1.2

Core Protocol

Underlying Protocols

Unix sockets, SHM, ...

UDP

HTTP 1.1 / SMTP / ...

TCP

IP v4/v6

(XML, xmlns, Schema) 1.0/1.1, xml:base, xml:id, Infoset, XOP, *XML Fragments, Binary XML*

Structure

URI / IRI

Resource Identifiers

Messaging Framework

- SOAP version 1.2
  - XML-based
  - Extensible
  - HTTP binding
    - 2 MEPs
    - SOAP Web Method features
    - Allows for REST-style interactions
- Java API for XML Web Services (JAX-WS)
  - Support for SOAP 1.2
  - WSDL generated on request

Addressing Information

- WS-Addressing 1.0
  - No addressing information provided in SOAP core capabilities
  - Typical interaction: HTTP request-response
  - Messages travel between endpoints
- JAX-WS API
  - Support for WS-Addressing
  - WSDL, EJB™ architecture, VM clients
  - SOAPAction header

Handling Non-XML Data

- XOP and MTOM
  - XML-binary Optimized Packaging
    - An alternate serialization of the Infoset
    - Optimizes serialization of binary content
  - SOAP Message Transmission Optimization Mechanism
    - How to use XOP in SOAP
- JAX-WS API
  - Support for XOP and MTOM
  - Maps **`xmime:expectedContentType`** to Java platform
  - Attach vs. in-line
  - FastWebServices

Doing Something Useful With This Framework

- JAX-WS API, using SOAP + WS-Addressing + MTOM, allows complex exchange of rich information
- Message exchange can be enhanced with added processing

But:

- How does each centre know what messages are accepted by an endpoint?
- How does each centre perform useful tasks with Web services?

Describing Java Web Services and SOA

- Loose coupling: simple tasks are exposed as stand-alone services
- In order for the centre to use those services and do useful things:
  - Need to describe how to interact with a service: **WSDL 2.0**
  - Need to describe how to orchestrate interactions between services: **WS-BPEL 2.0**
  - Need to describe interactions between sets of services and orchestrations: **WS-CDL 1.0**

Service Description

- Needed: A machine-processable language for describing Web services in an SOA
- WSDL 2.0
  - New features
    - Reusability: inheritance
    - Extensibility: MEPs, WS-Policy
    - Web-friendly: safe operations, complete HTTP/1.1 support
    - SOAP 1.2 binding
  - Apache Woden -> Axis 2
    - Parsing
    - SOAP components
    - HTTP binding, operation safety, RPC style
- **Java Business Integration** (JBI) uses WSDL 2.0

Business Processes

- Needed: A machine-processable language for describing in an SOA how business processes are performed
- Web Services Business Process Execution Language (WS-BPEL) 2.0
  - Specify business processes composed of Web services
  - Expose business processes as Web services
  - Abstract and executable processes
- Java technology implementations abound
  - ActiveEndpoints, IBM, Intalio, SEEBURGER, Sun
- **JBI** uses BPEL

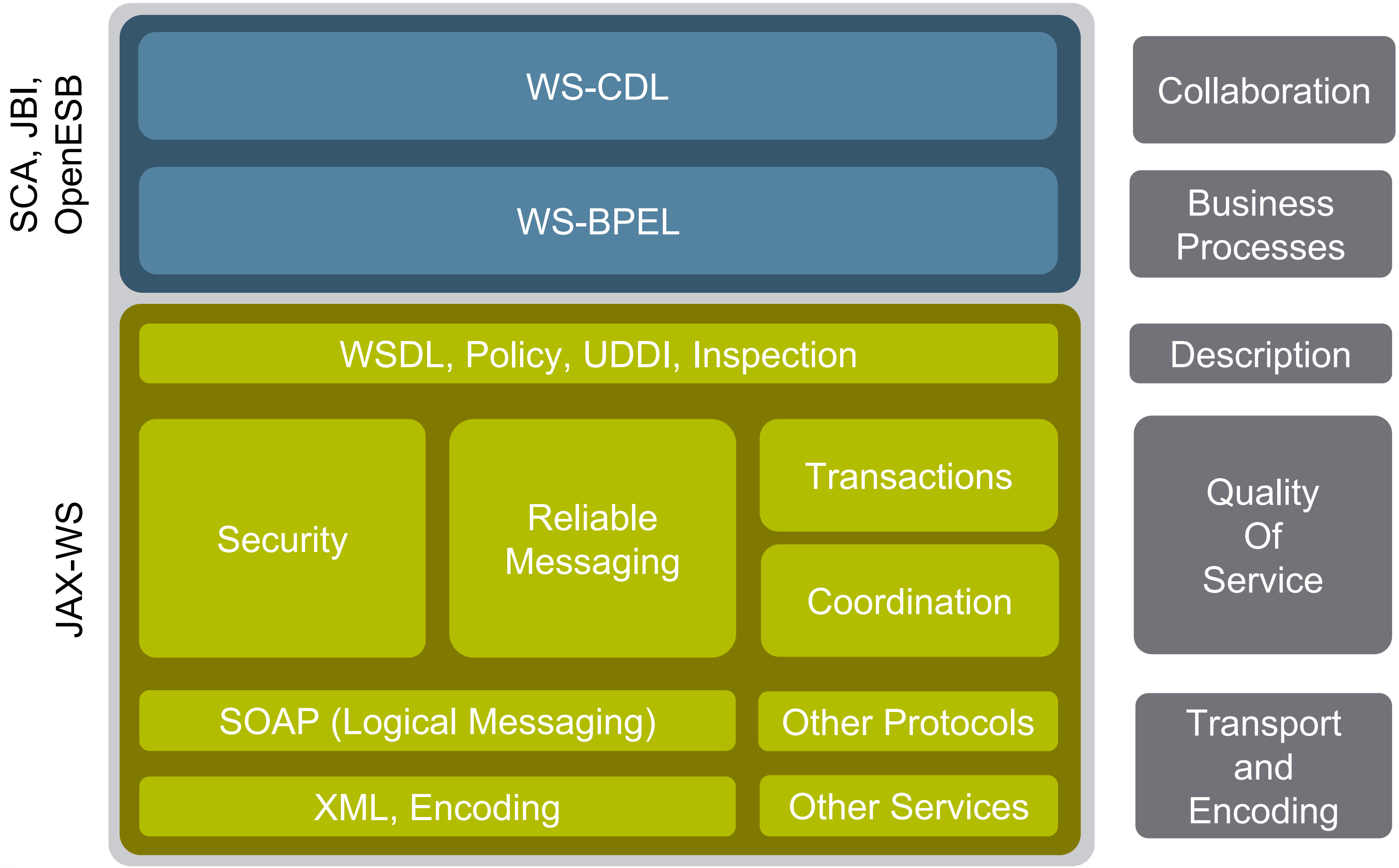
Collaboration

- Needed: A machine-processable language for describing how business processes and other application collaborate
- WS-CDL: Choreography Description Language
  - Describes collaboration protocols of cooperating service participants in which:
    - Services act as peers
    - Interactions may be long-lived and stateful
  - Build distributed Service Oriented Architecture (SOA) based systems through the principle concept of interaction
- Java technology implementations
  - Pi4tech, Imperial, Adobe, etc.
- CDL can use **JBI**
- **OpenESB** can use CDL

Service Constraints/Semantics

- Needed: A machine-processable language for describing service constraints and semantics
- Service component architecture
    - Assembly of individual services into larger applications
    - Defining format for configuring a module for deployment
    - Components that are composed into modules can be
    - Custom programs
    - BPEL processes
    - WS-A endpoints
    - Anything else conforming to the interface specification
- **JBI** and SCA are applications

Java Web Technologies



Agenda

Motivation for Web Services and SOA

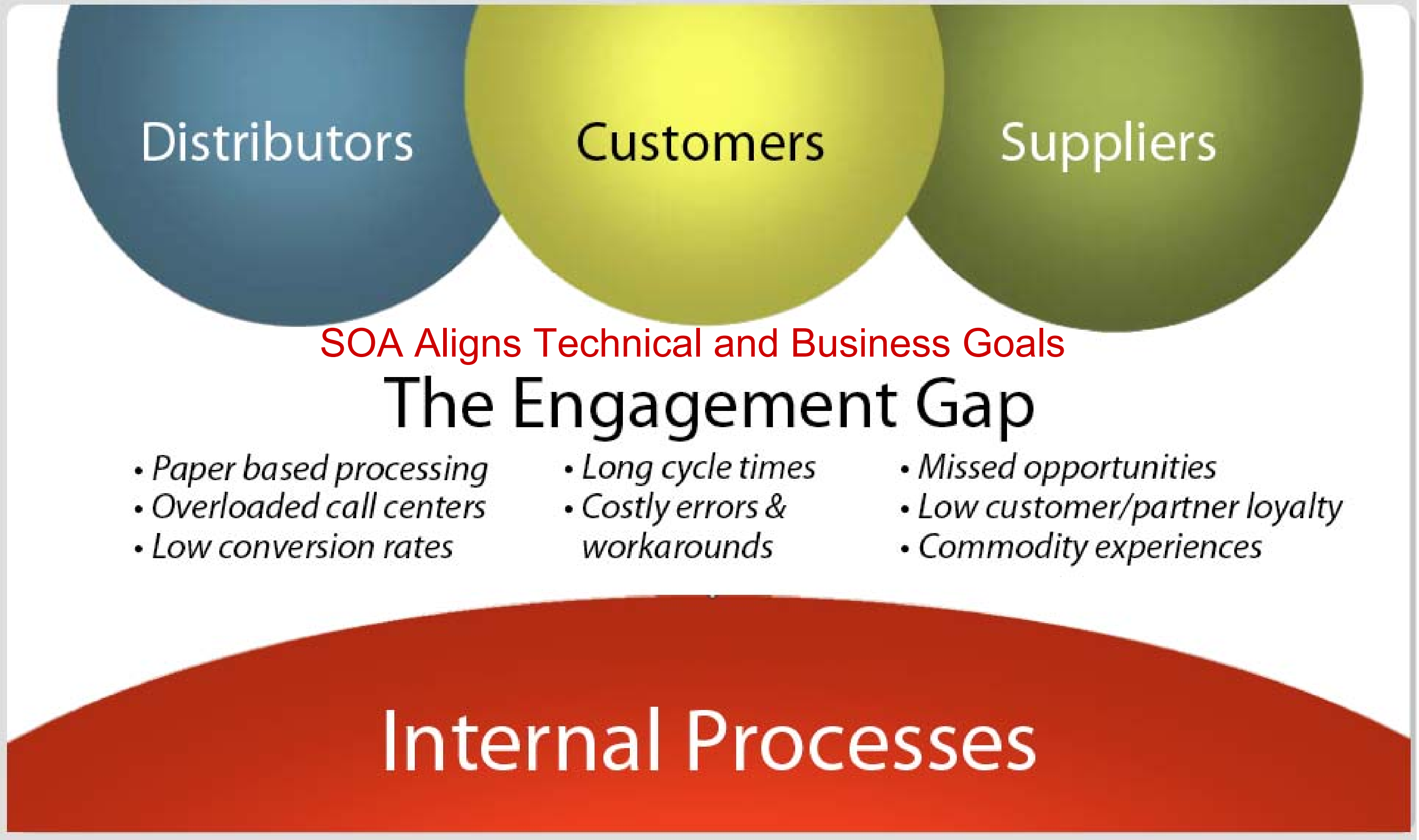
Technologies

Relating the Tiers

Document Services Platform

Applications

Where Do These Technologies Fit In?



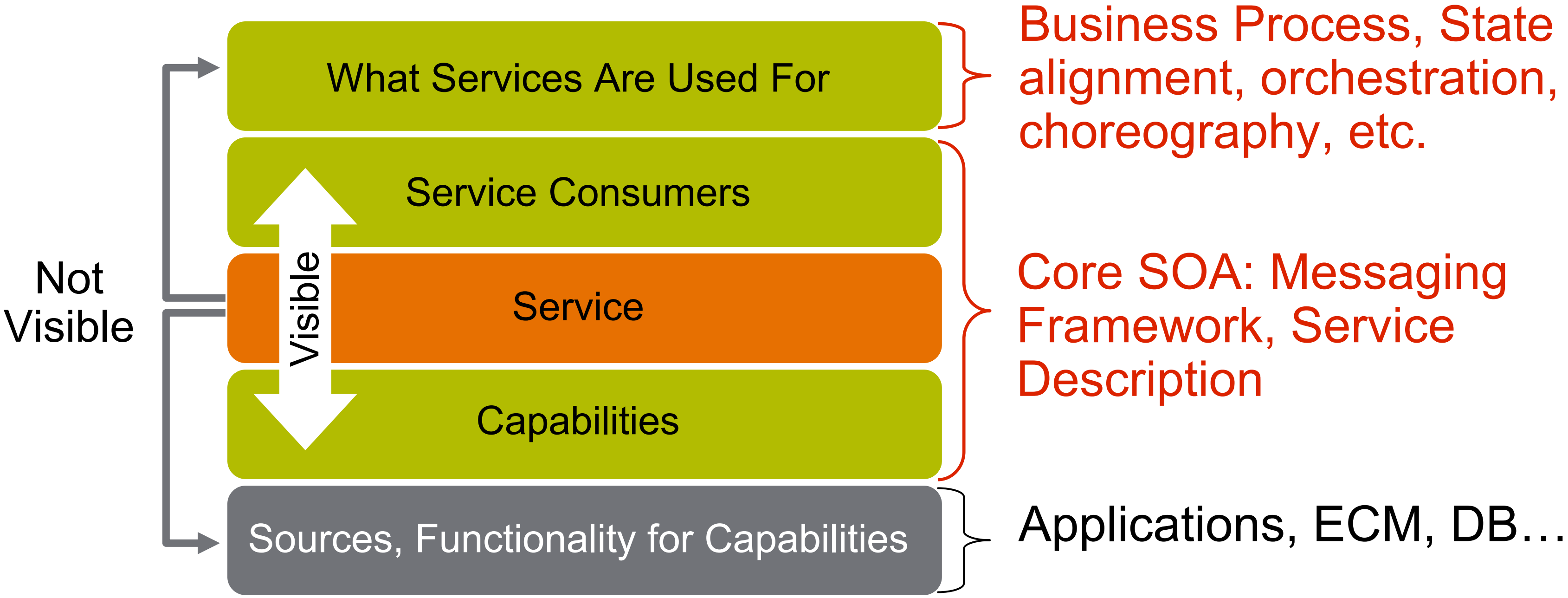
SOA Aligns Technical and Business Goals

- Paper based processing
- Overloaded call centers
- Low conversion rates

- Long cycle times
- Costly errors & workarounds

- Missed opportunities
- Low customer/partner loyalty
- Commodity experiences

How Do These All Fit Together?


What Services Are Used For
Service Consumers
Service
Capabilities
Sources, Functionality for Capabilities
Visible
Not Visible
Business Process, State alignment, orchestration, choreography, etc.
Core SOA: Messaging Framework, Service Description
Applications, ECM, DB…

In a layer diagram, layer “n” is only visible to layers (n +1) and (n – 1)

Agenda

Motivation for Web Services and SOA

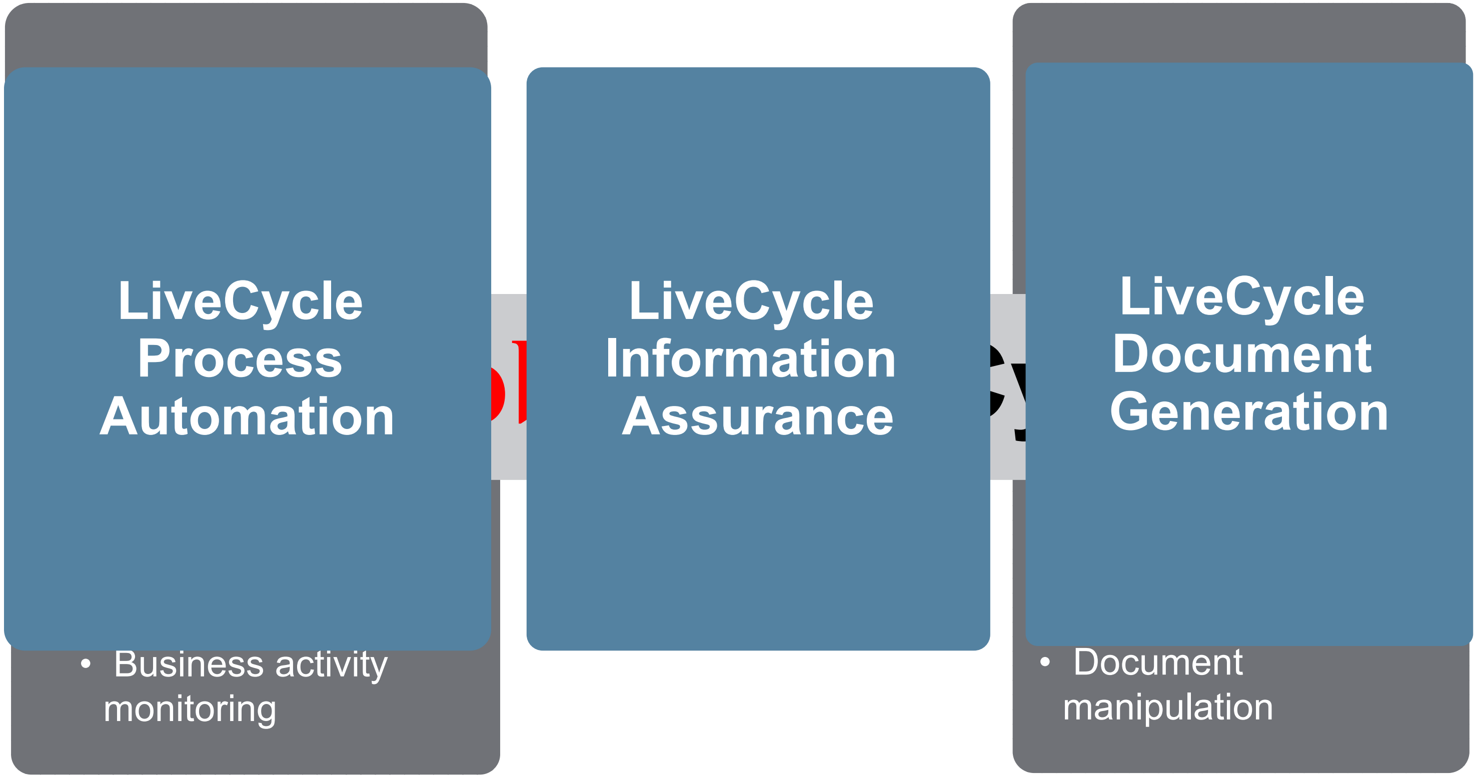
Technologies

Relating the Tiers

Document Services Platform

Applications

How Do These All Fit Together



LiveCycle Process Automation

- Business activity monitoring

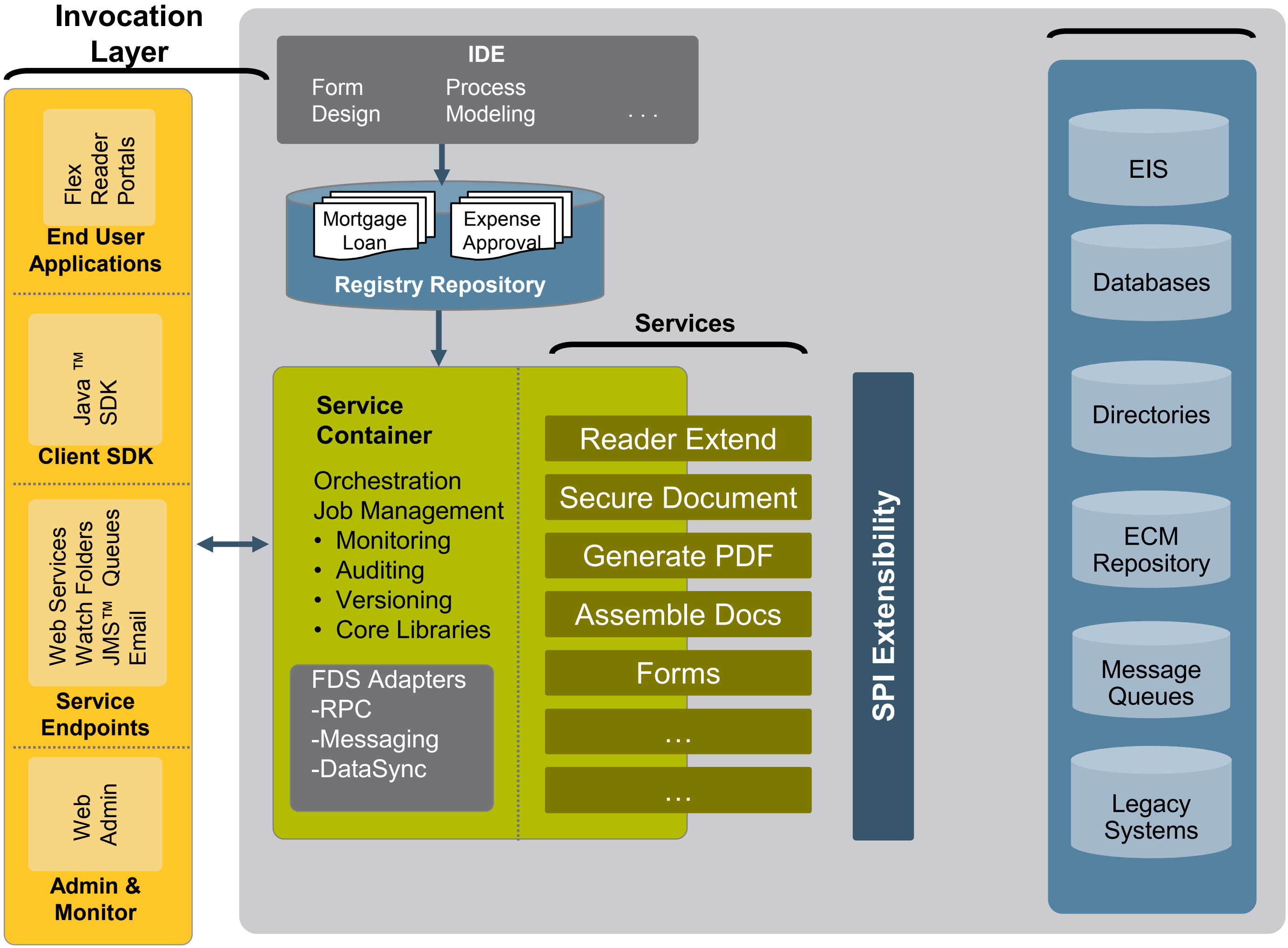
LiveCycle Information Assurance

LiveCycle Document Generation

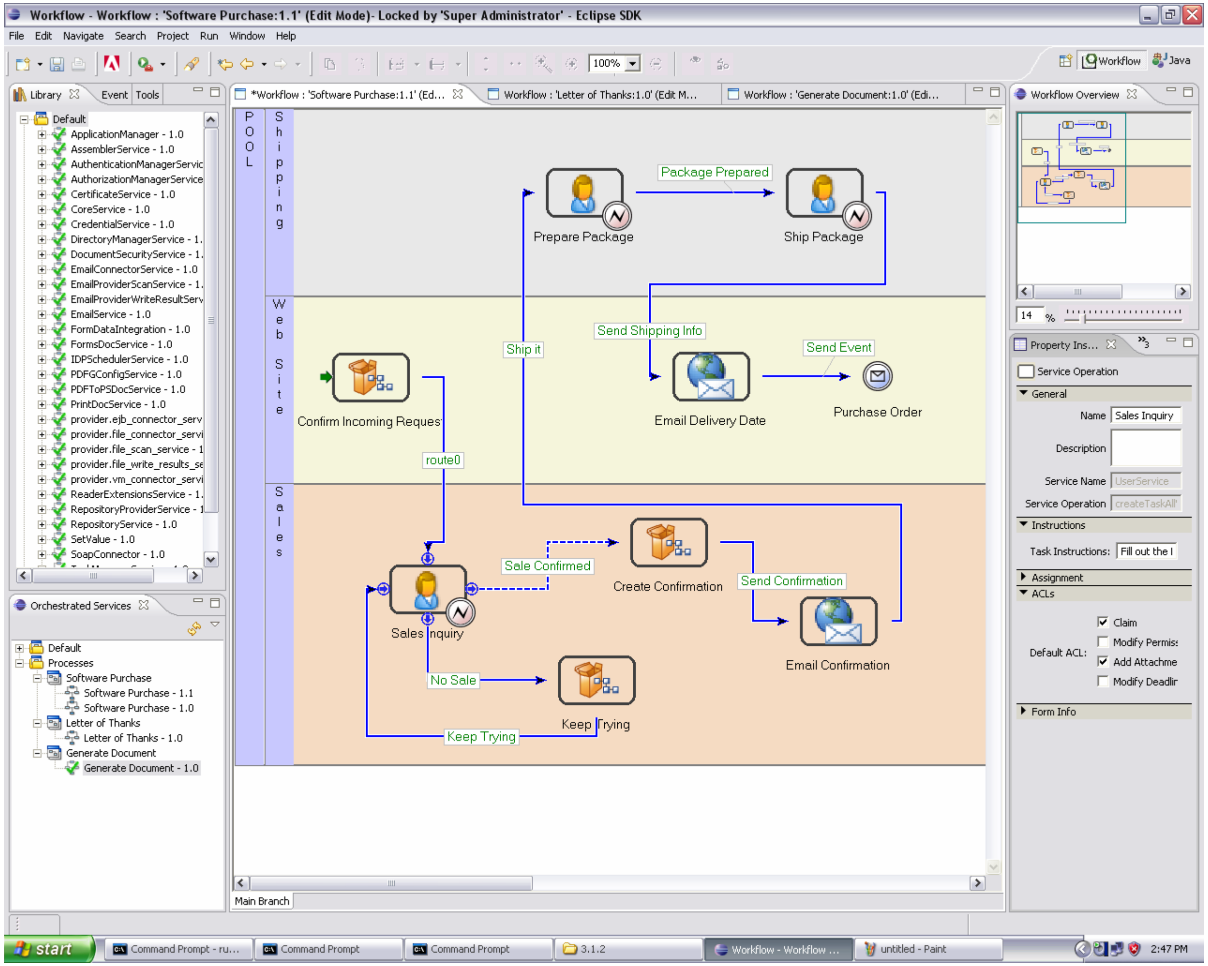
- Document manipulation

Enterprise Platform

Document Services Platform



Process and Collaboration Design



Workflow - Workflow : 'Software Purchase:1.1' (Edit Mode)- Locked by 'Super Administrator' - Eclipse SDK

Property Ins...

□ Service Operation

General

Name: Sales Inquiry

Description:

Service Name: UserService

Service Operation: createTaskAll'

Instructions

Task Instructions: Fill out the I

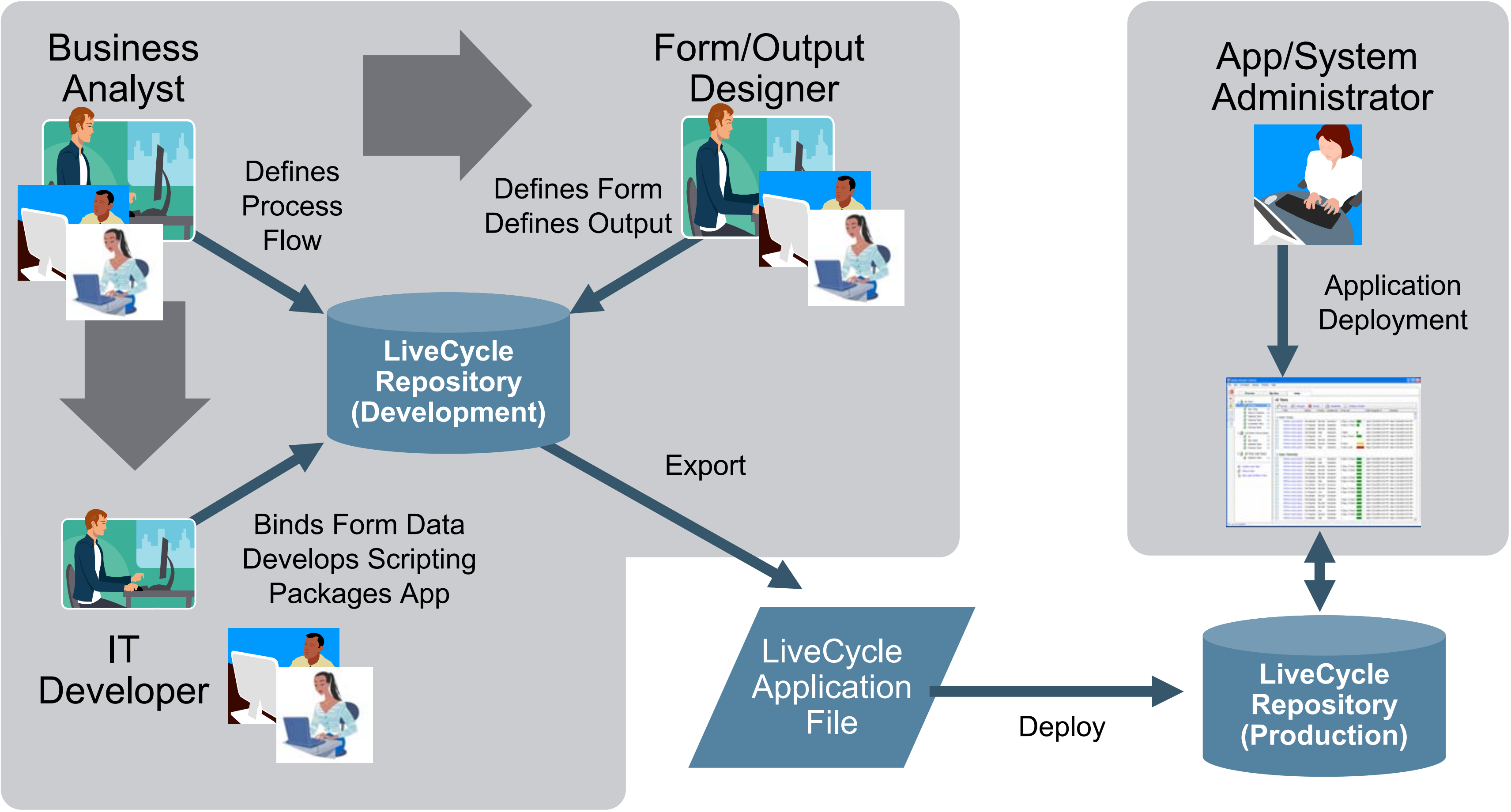
Assignment

ACLs

Default ACL:
- ☑ Claim
- □ Modify Permis:
- ☑ Add Attachme
- □ Modify Deadlir

Form Info

Application Development Flow



Business Analyst

Defines Process Flow

Form/Output Designer

Defines Form Defines Output

LiveCycle Repository (Development)

IT Developer

Binds Form Data Develops Scripting Packages App

Export

LiveCycle Application File

Deploy

App/System Administrator

Application Deployment

LiveCycle Repository (Production)

Data Capture

eForms and Data Capture

Static Form

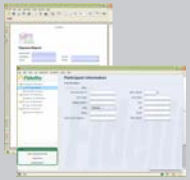
Paper Form Experience

LC Designer

Dynamic Form

eForm Experience

LC Designer



Activity Guide

Wizard Experience

XFA form -> Flex App

Minimal re-authoring using pre-defined styles

LC Designer, Flex Builder

RIA

App Experience

Richer components for data capture

Flex SDK/Builder

Agenda

Motivation for Web Services and SOA

Technologies

Relating the Tiers

Document Services Platform

Applications

Back to Our Scenario

- Geographically distributed manufacturing operation
- Global partner for multiple sectors
- JIT delivery of product by stage
  - Base components
  - Assembled components
  - Finished goods
  - Distribution
- Guaranteed uninterrupted operations required

Composite Application Scenario

- Each operations centre exposes activities as services
  - Web services messaging framework provides the transport, addressing, and payload management
  - WSDL provides the advertisable description
- Each operations centre composes activities into business processes
  - WS-BPEL provides the process definition
  - WSDL provides the advertisable description
- Each operations centre monitors and manages its processes using Web sevices interfaces

Composite Application Scenario

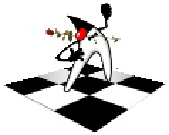
- Global operations composes each centre’s operations into choreographies
    - WS-CDL provides the choreography
    - WSDL provides the service description for composing the choreography
- Global operations monitors and manages its choreographies using Web service interfaces
- Adobe leveraged JAX-WS API technologies in a number of implementations to provide these features

DEMO

Adobe LiveCycle ES

References

- W3C XML Protocol
  http://www.w3.org/2000/xp/Group
- W3C Web Services Description 2.0
  http://www.w3.org/2002/ws/desc/
- OASIS WS-BPEL
  http://docs.oasis-open.org/wsbpel/2.0/
- W3C Web Services Choreography 1.0
  http://www.w3.org/2002/ws/chor/
- Adobe LiveCycle
  http://www.adobe.com/products/livecycle
- JAX-WS API
  http://java.sun.com/webservices/jaxws
- WSIT (Project Tango)
  http://wsit.dev.java.net



- Java Business Integration
  http://www.jcp.org/en/jsr/detail?id=208

For More Information

- Sessions:
    - **TS-8459**
      Service Virtualization:
      Separating Business Logic From Policy Enforcement
    - **TS-8897**
      Designing Service Collaborations:
      The Design of “Wire”-Centric Integration
- BOFs
    - **BOF-8238**
      Building Composite Services Applications
    - **BOF-8872**
      Java Business Integration 2.0
- See Services and Integration Track for more

Q&A

Next-Generation Java Web Technologies in Practice: The Document Services Platform

Charlton Barreto

Senior Computer Scientist/Architect
Adobe Systems, Inc.
http://www.adobe.com/products/livecycle/

TS-8844