LogicLibrary™

SDA Reuse Best Practices

Brent Carlson, VP of Technology

5 Aug 04

KNOW WHAT YOU HAVE. MOVE AHEAD.

Agenda

- Software Development Asset (SDA) concepts
- SDAs within a Service-Oriented Architecture
  - Internal
  - External (B2B)
- SDA reuse best practices
- SDA management/governance case study
- Logidex introduction

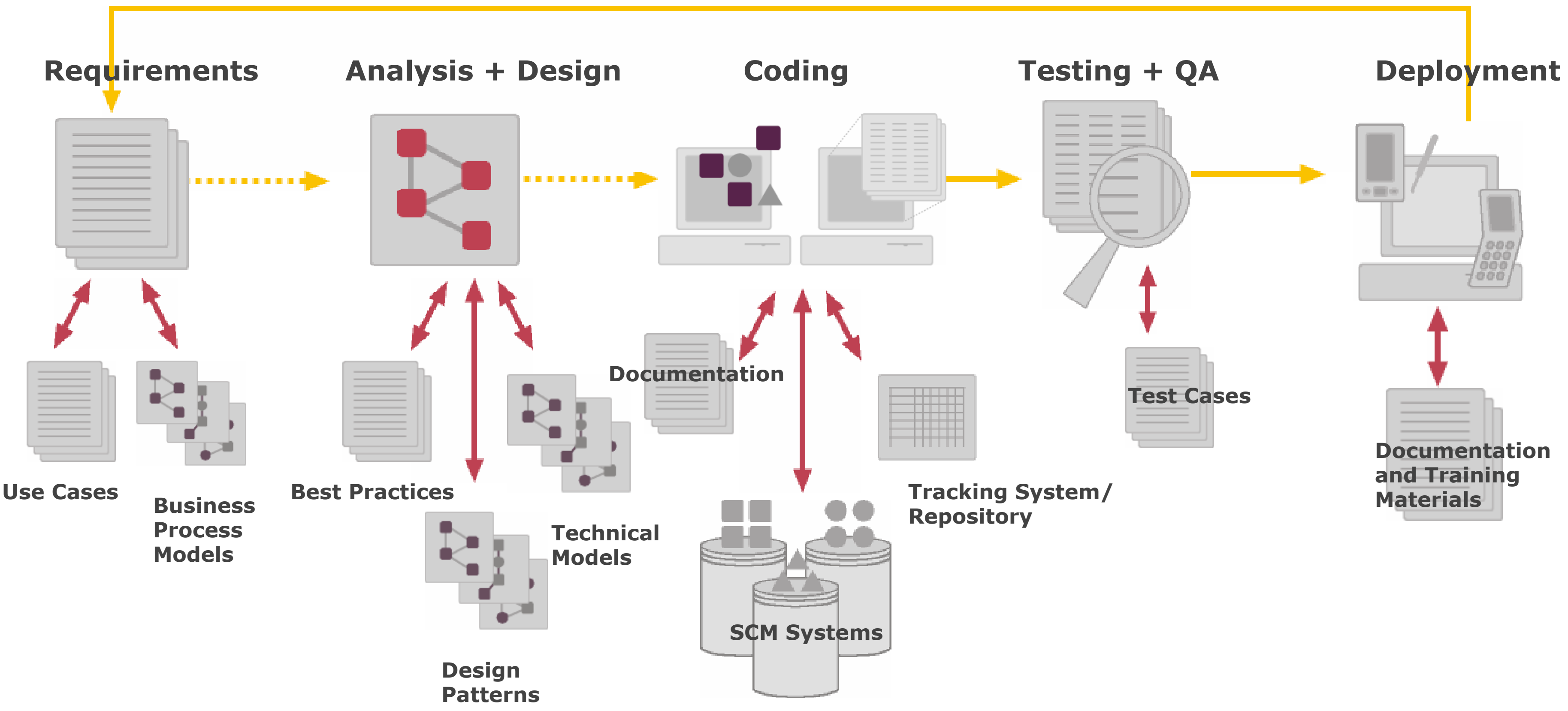
LogicLibrary

SDA Concepts

What is an SDA?

- An SDA is “something of value to an IT organization,” including:
  - **Knowledge** assets
    - Architectures
    - Best practices and processes
    - Design patterns
    - …
  - **Executable** assets
    - Web services
    - Data views
    - Components
    - Applications
    - …
- SDAs are composed of ***metadata***, such as:
  - **Artifacts:** work products, such as code, schemas, models, executable modules, …
  - **Classifiers:** searchable and reportable values such as keywords, development effort, owner, language, …
  - **Relationships to other assets:** dependencies, prerequisites, other asset versions, …

SDA Development Lifecycle



Requirements

Analysis + Design

Coding

Testing + QA

Deployment

Use Cases

Business Process Models

Best Practices

Technical Models

Design Patterns

Documentation

SCM Systems

Tracking System/ Repository

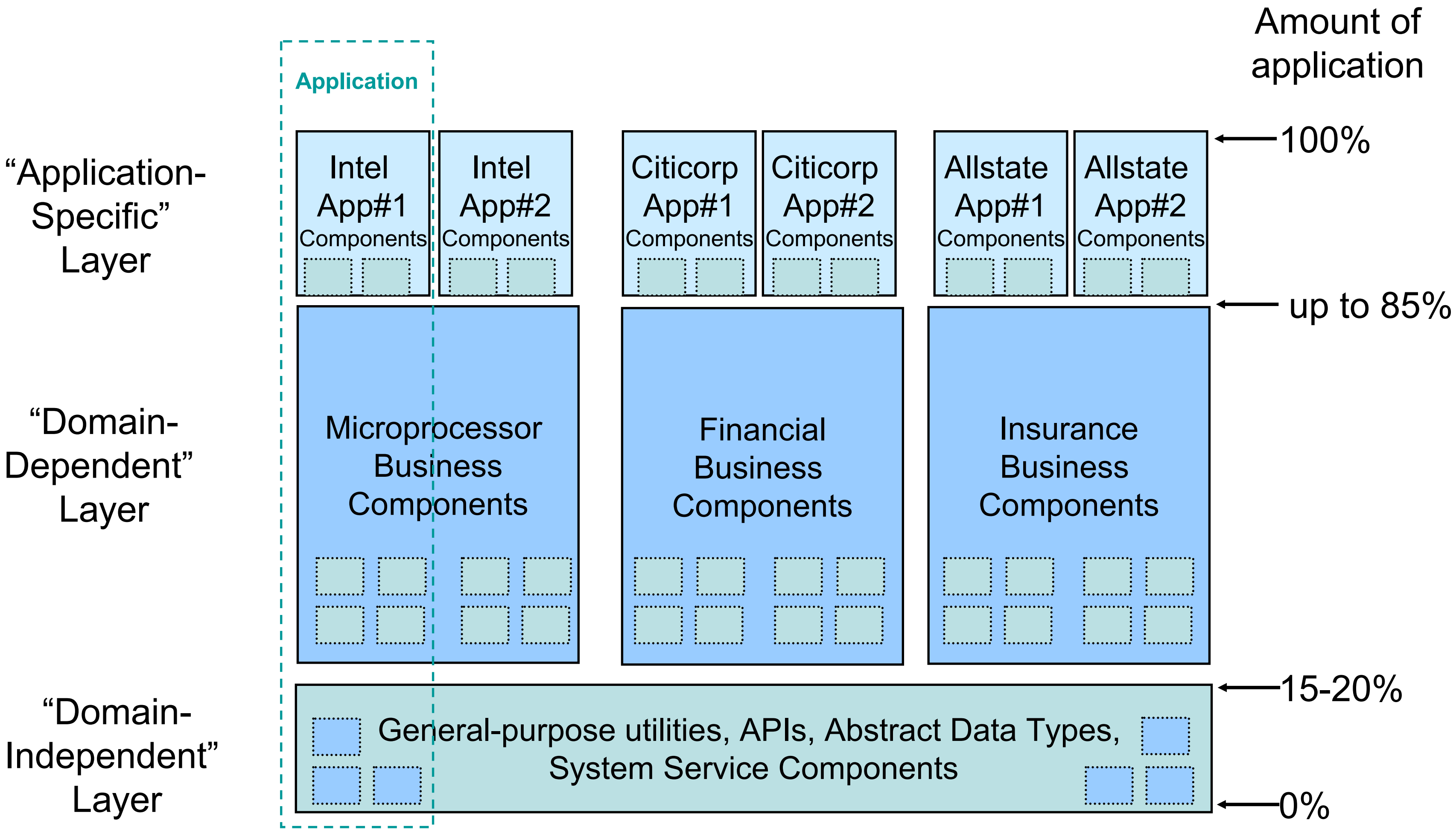
Test Cases

Documentation and Training Materials

Managing SDAs in Context

- SDAs live in multiple contexts:
  - Application Context
  - Technical Context
  - Business Context
  - Deployment Context
- The first three contexts are directly relevant to the software development lifecycle
- Linking deployment context to development contexts is becoming increasingly important
  - Web services management
  - Maintenance/deployment lifecycles

Three Classes of SDAs


Amount of application
Application
"Application-Specific" Layer
"Domain-Dependent" Layer
"Domain-Independent" Layer
Intel App#1 Components
Intel App#2 Components
Citicorp App#1 Components
Citicorp App#2 Components
Allstate App#1 Components
Allstate App#2 Components
Microprocessor Business Components
Financial Business Components
Insurance Business Components
General-purpose utilities, APIs, Abstract Data Types, System Service Components
100%
up to 85%
15-20%
0%

LogicLibrary

SDAs Within a Service-Oriented Architecture

SOA Concepts

- What is service-oriented architecture?
  - Definition and delivery of core application functions through a series of coarse-grained services meant to be assembled through a message-oriented infrastructure
  - **Strongly conducive to SDA reuse**
- Not all Web services fit into an SOA
  - e.g., fine-grained RPC-type services
- Not all SOAs use Web services technology
  - e.g., guaranteed messaging using message-oriented middleware (MOM) technology
- Services may be implemented using a variety of techniques
  - COM/.NET/J2EE components, integration server adapters, …

Making Sense of Services within an SOA

To be effective, services within an SOA must be:

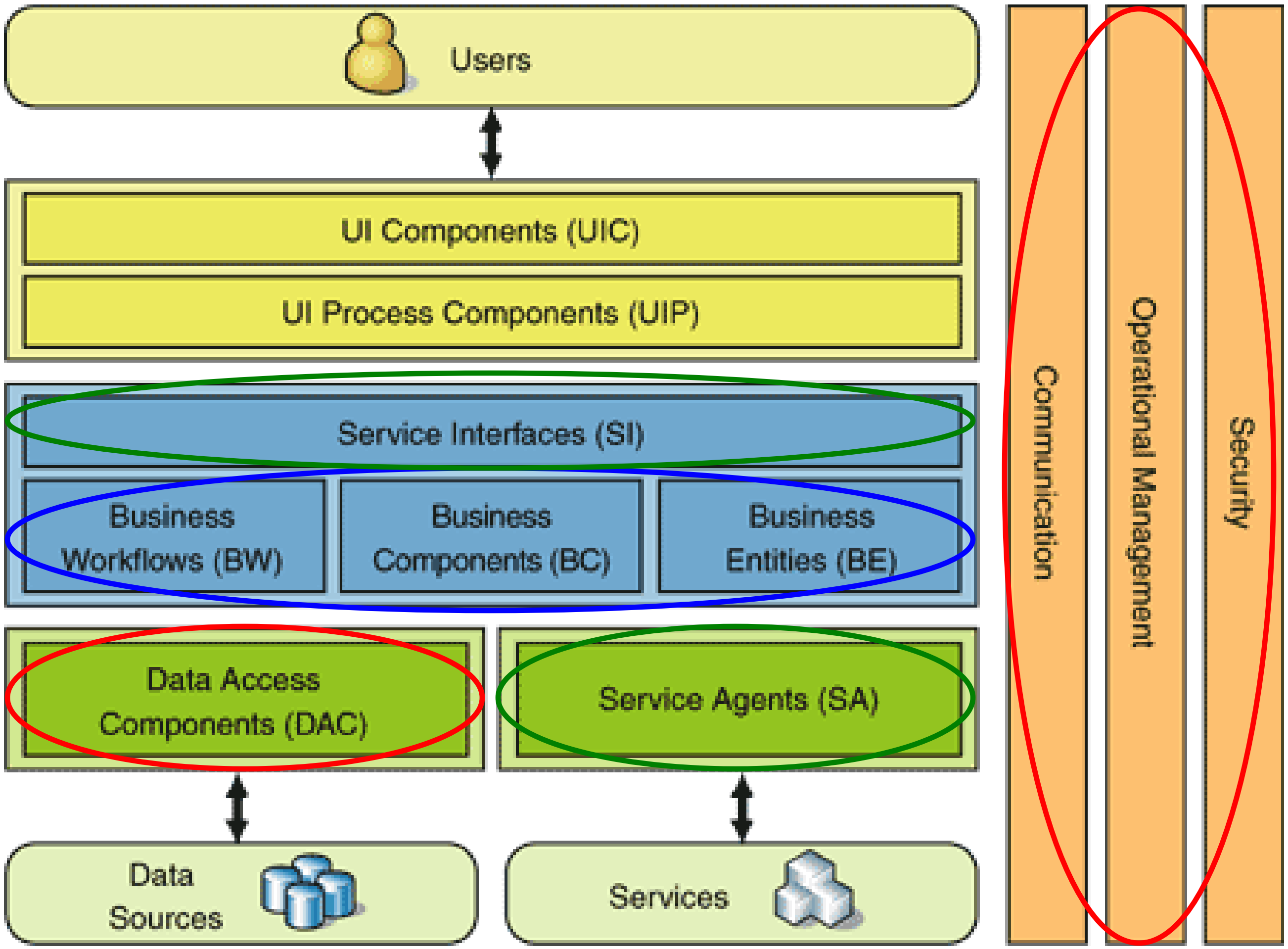
- Derived from ***business requirements***
  - Typically expressed as business processes
- Sufficiently ***coarse-grained*** to:
  - Enable workflow-oriented assembly
  - Maximize work-to-overhead ratio
- Be placed into ***business*** and ***technical context***
  - Managed definition to avoid redundancy
  - Managed implementation to ensure best practices are being followed

SDAs within an SOA

SDAs are key building blocks within an SOA

- Services themselves are SDAs
  - Meant to be published and consumed
  - Need to be managed over their typically multi-versioned lifecycle
- … and services consume other SDAs
  - Other services
  - Components (both business and technical)
  - Legacy APIs
  - Knowledge assets (e.g., Enterprise Solution Patterns, such as Service Interface and Service Gateway)
  - …

Sample Technical Context: .NET



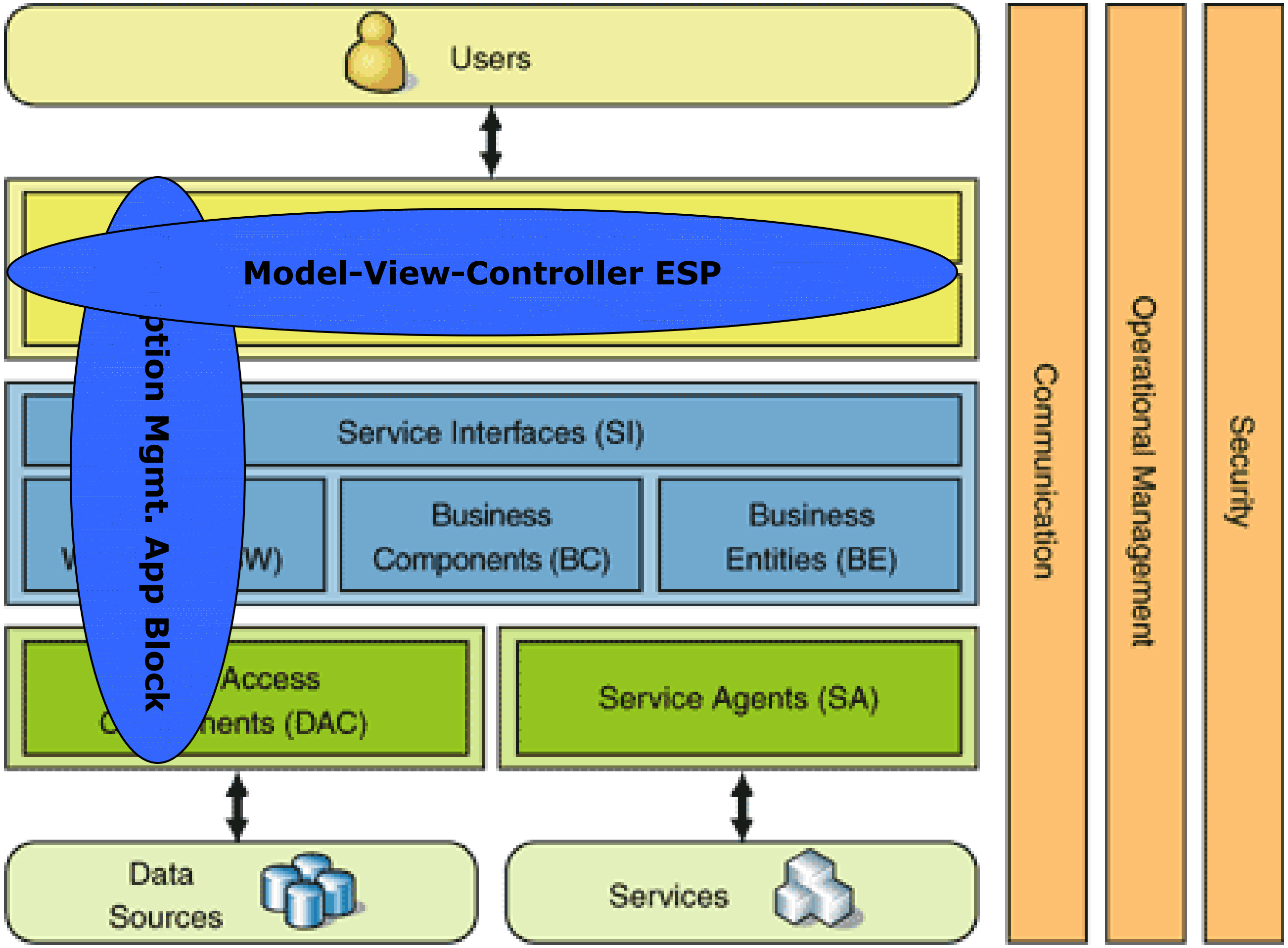
Business Services

Business Components

Technical Components

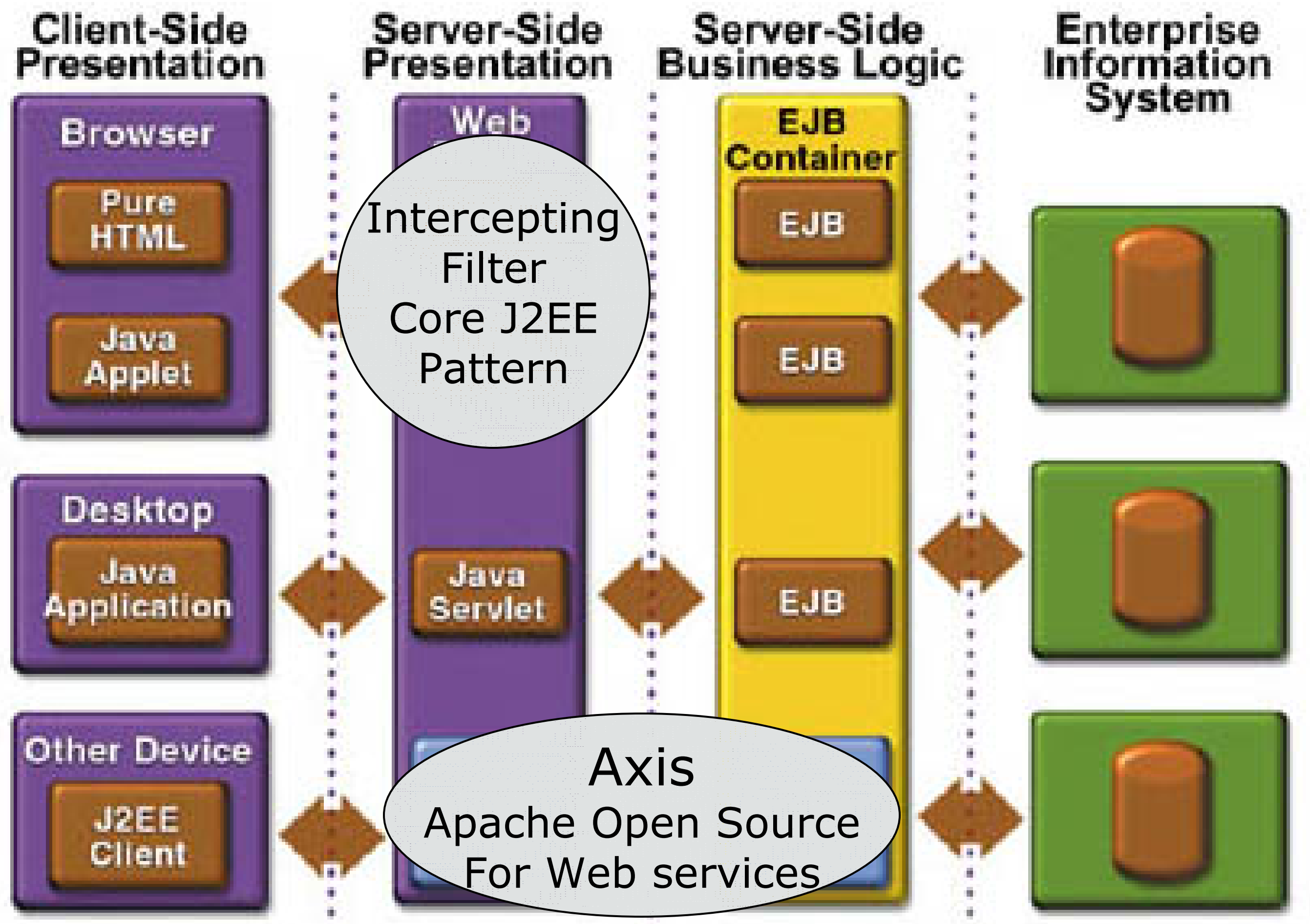
* http://msdn.microsoft.com/architecture/patterns/

.NET SDA Examples



* http://msdn.microsoft.com/architecture/patterns/

Sample Technical Context: J2EE

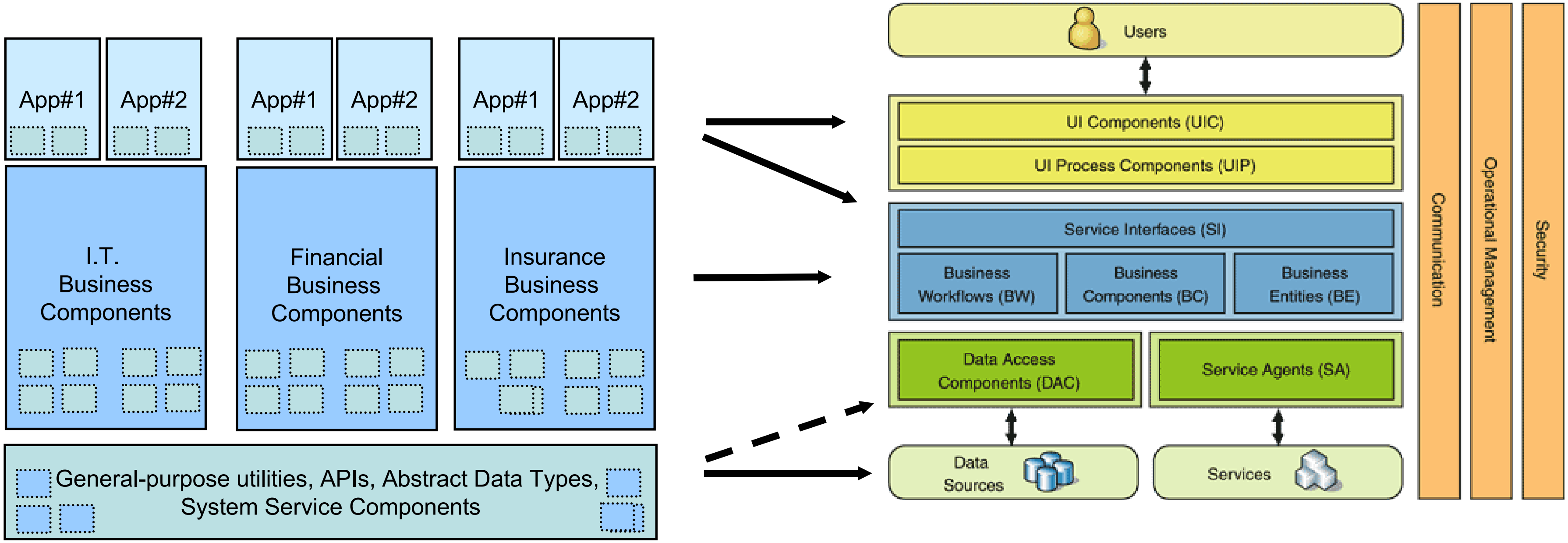


Intercepting Filter
Core J2EE
Pattern

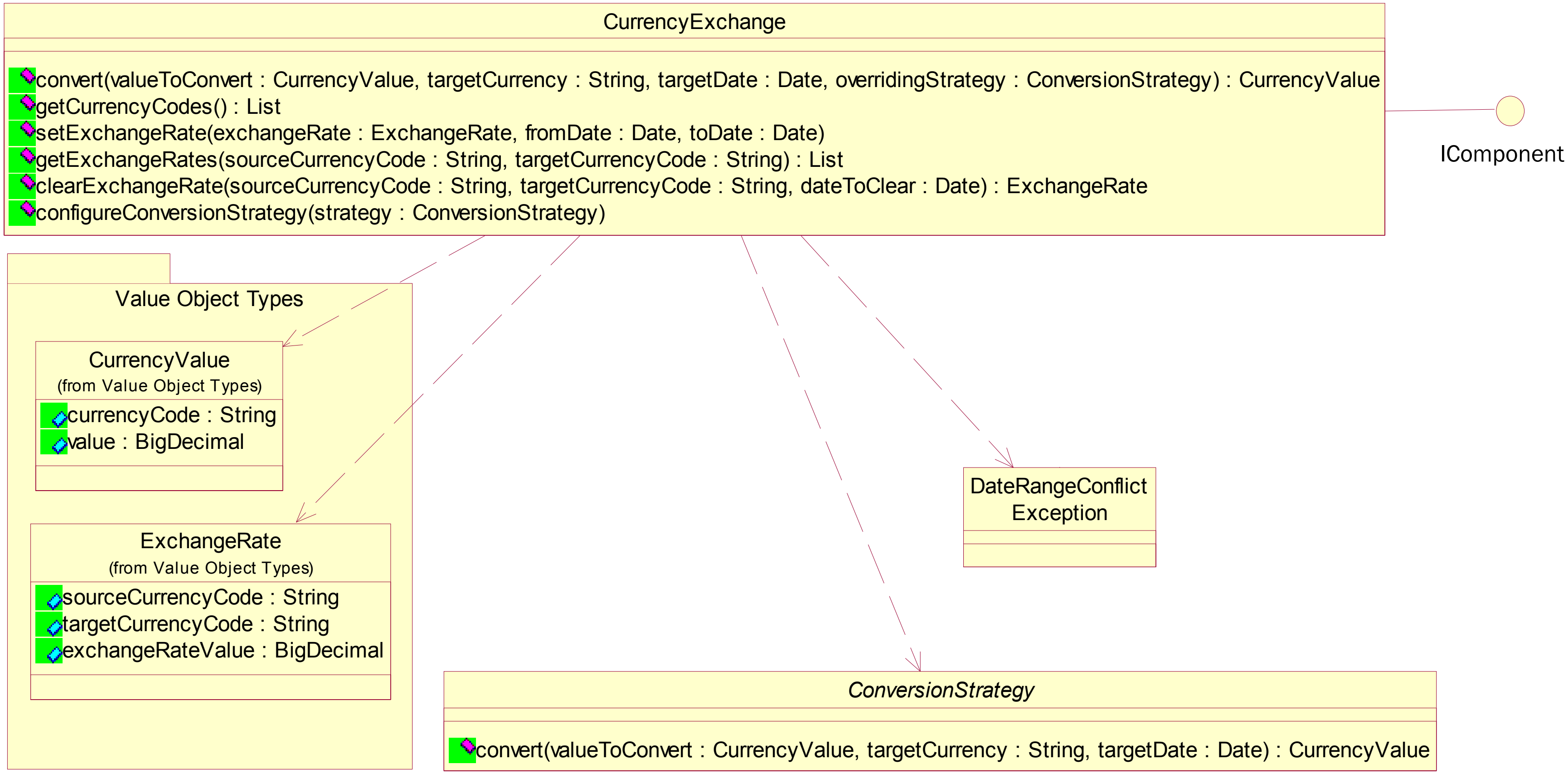
Axis
Apache Open Source
For Web services

* http://java.sun.com

Mapping SDA Classes to the .NET Technical Context



An Example SDA: CurrencyExchange Component



CurrencyExchange

convert(valueToConvert : CurrencyValue, targetCurrency : String, targetDate : Date, overridingStrategy : ConversionStrategy) : CurrencyValue
getCurrencyCodes() : List
setExchangeRate(exchangeRate : ExchangeRate, fromDate : Date, toDate : Date)
getExchangeRates(sourceCurrencyCode : String, targetCurrencyCode : String) : List
clearExchangeRate(sourceCurrencyCode : String, targetCurrencyCode : String, dateToClear : Date) : ExchangeRate
configureConversionStrategy(strategy : ConversionStrategy)

IComponent

Value Object Types

CurrencyValue
(from Value Object Types)
currencyCode : String
value : BigDecimal

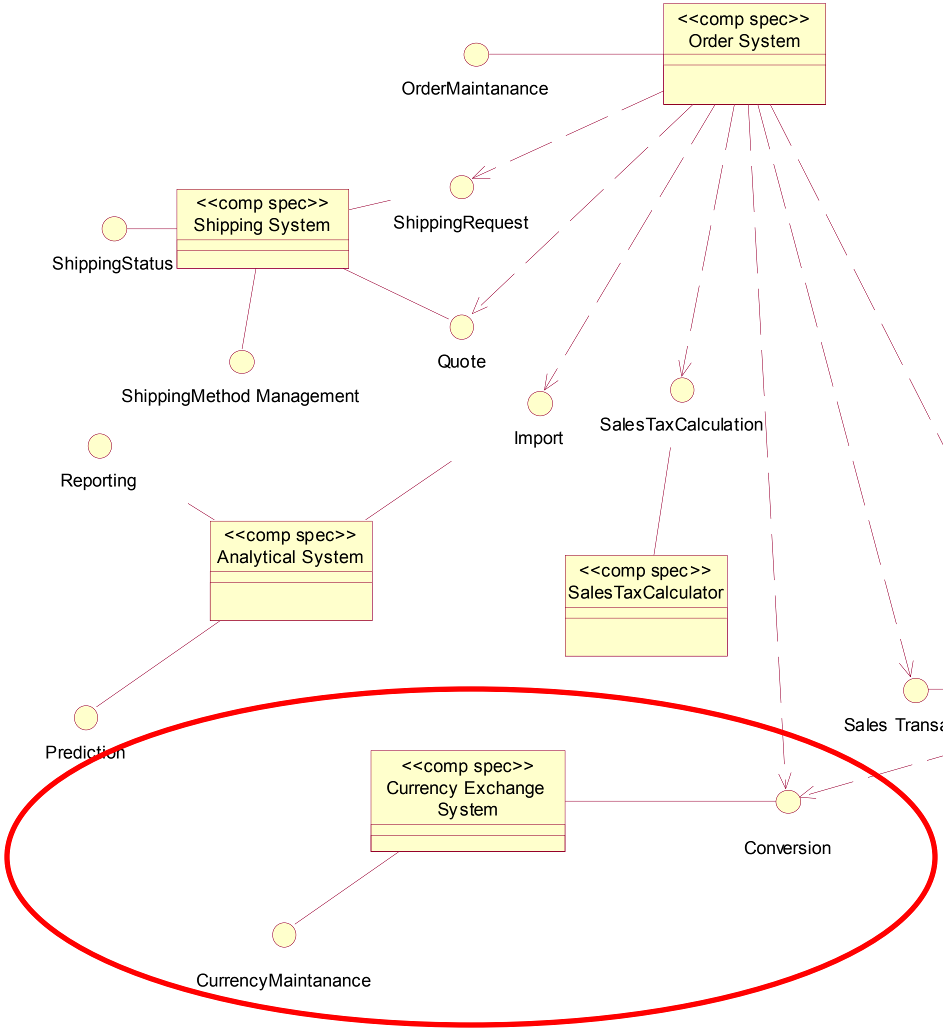
ExchangeRate
(from Value Object Types)
sourceCurrencyCode : String
targetCurrencyCode : String
exchangeRateValue : BigDecimal

DateRangeConflict
Exception

ConversionStrategy

convert(valueToConvert : CurrencyValue, targetCurrency : String, targetDate : Date) : CurrencyValue

Our SDA's Business Context: e-Commerce

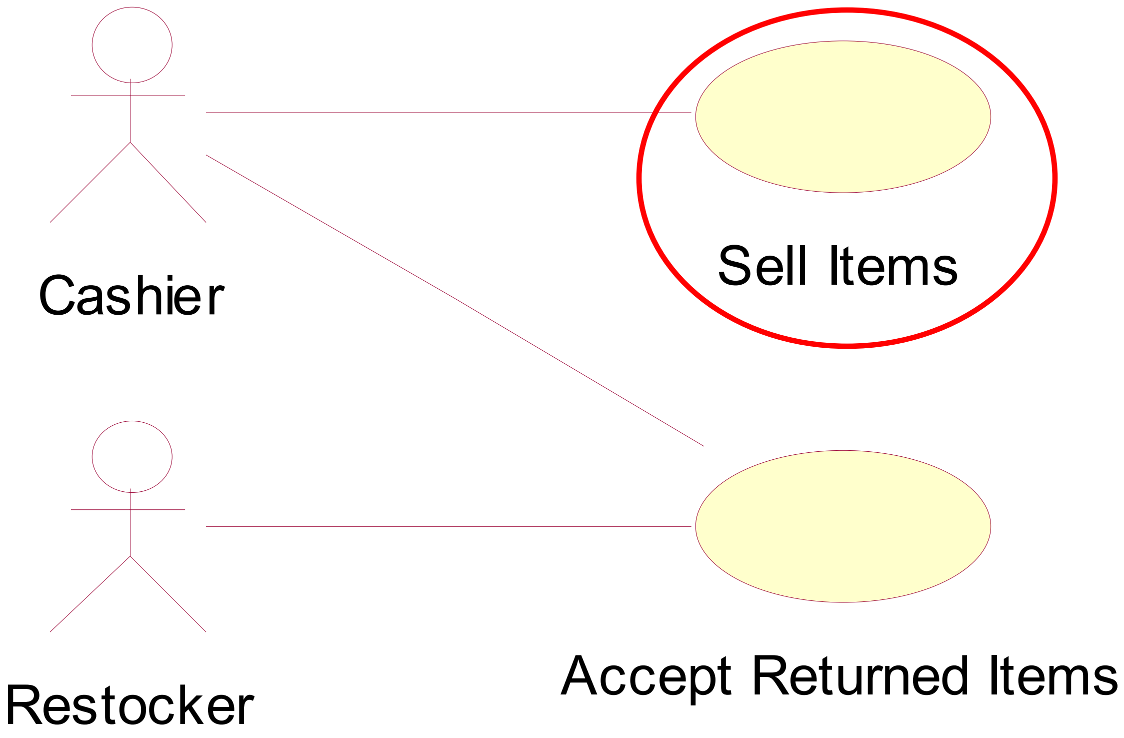


Business Context Elements

- Primary UML construct is class diagram laying out coarse-grained reference components/services
- Other UML constructs to consider:
  - Use Cases
    - Establish initial requirements for business function
      - Actors
      - Preconditions
      - Functional scenario / use case steps
  - Activity Diagrams
    - Describe detailed process or subprocess flow underlying a use case
    - Specific activities can be mapped to functional capabilities to be implemented as Web services

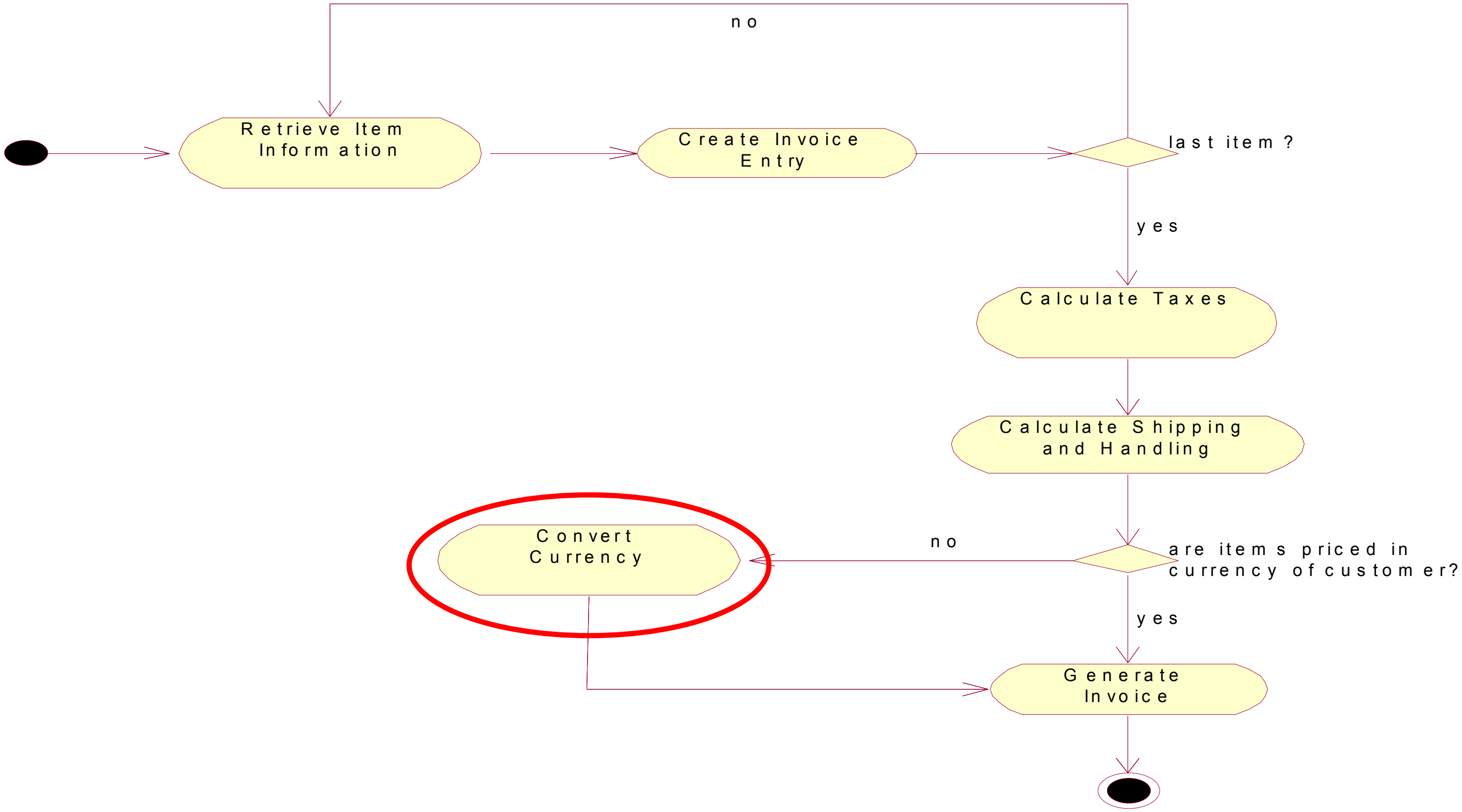
Expanding Our Business Context

- Sample Use Case Diagram



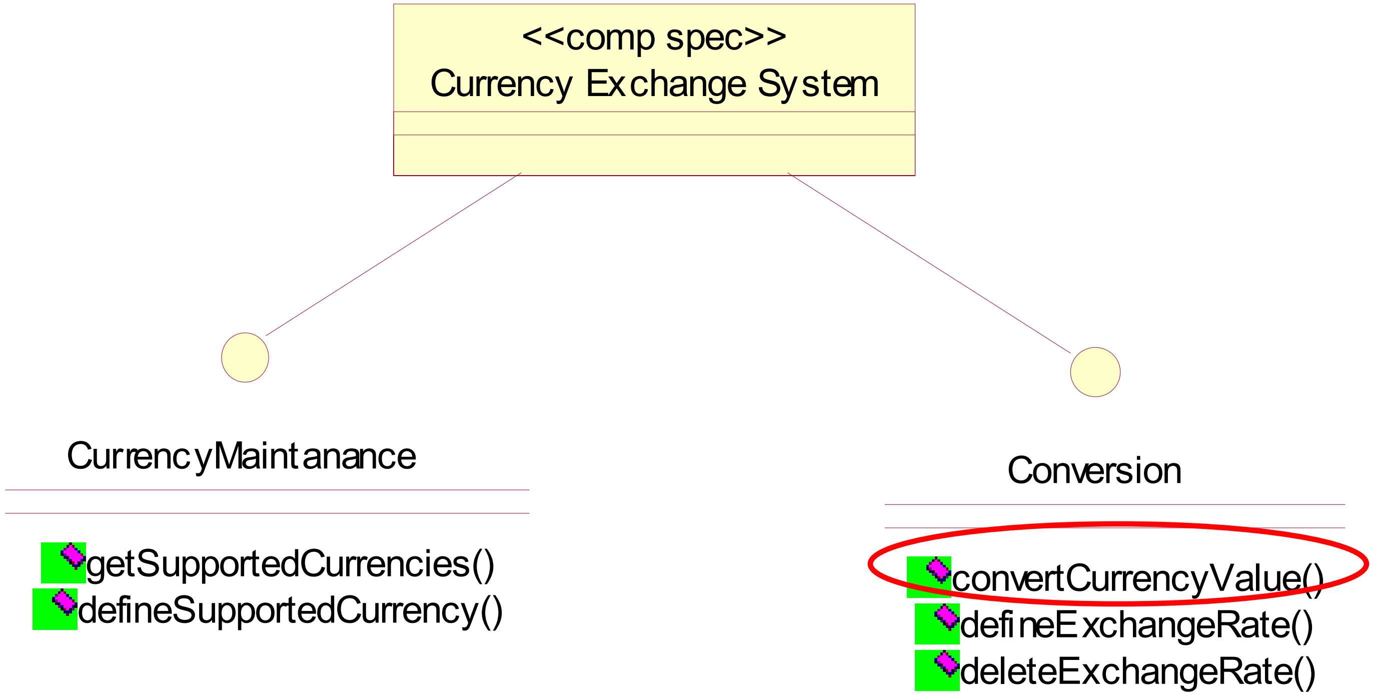
Expanding Our Business Context…

- Drilling into Sell Items Activity Diagram…


no
Retrieve Item Information
Create Invoice Entry
last item ?
yes
Calculate Taxes
Calculate Shipping and Handling
Convert Currency
no
are items priced in currency of customer?
yes
Generate Invoice

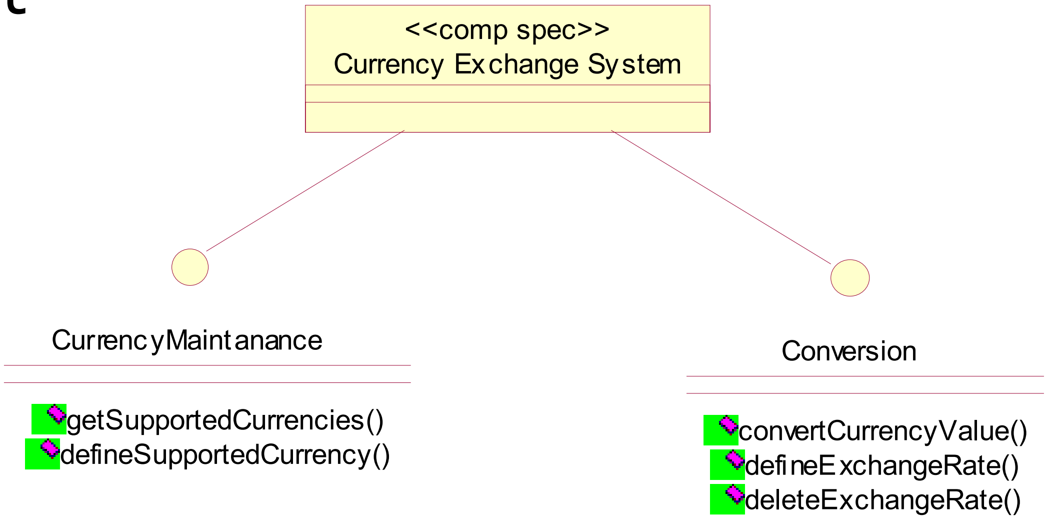
Expanding Our Business Context…

- Associating Convert Currency Activity with Component…



Putting our SDA into Business Context

- Returning to our component model…
  - Map our reference component to our example asset



<<comp spec>>
Currency Exchange System

CurrencyMaintanance

getSupportedCurrencies()
defineSupportedCurrency()

Conversion

convertCurrencyValue()
defineExchangeRate()
deleteExchangeRate()

Reference Component operation	Component method
getSupportedCurrencies	getCurrencyCodes
defineSupportedCurrency	no direct equivalent
convertCurrencyValue	convert
defineExchangeRate	setExchangeRate
deleteExchangeRate	clearExchangeRate

A Business Process Example: RosettaNet

- What is RosettaNet?
  - “RosettaNet is a consortium of major Information Technology, Electronic Components, Semiconductor Manufacturing and Telecommunications companies working to create and implement industry-wide, open e-business process standards. These standards form a common e-business language, aligning processes between supply chain partners on a global basis.”

From RosettaNet’s home page at www.rosettanet.org

Why Is RosettaNet Interesting?

- Prasad Rampalli, Intel’s IT Architect:
  - “We have championed the whole B2B commerce space with a lingua franca based on the RosettaNet standard. It is based on an XML messaging format and has a sufficient level of encryption to enable secure B2B conversations with the multitudes of trading partners that we've got.”

    From Dec 15, 2002 ZDNet interview

    http://techupdate.zdnet.com/techupdate/stories/main/0,14179,2901716,00.html

Emerging Business Process Standards

- Significant industry movement away from EDI towards standardized, XML-based business process and messaging definitions
  - Web services is accelerating this trend
- Other examples of business process standards:
  - ACORD: insurance
  - IFX: financial services
  - ebXML: electronic business/trading

Emerging Business Process Standards

- Advantages:
  - Can learn from industry best practices and experience
    - XML-based message sets
    - Standard data dictionaries
    - Reference business processes
  - Packaged applications are adopting standards as they move towards Web service-oriented APIs

Besides, your business partners are probably already doing something here and will take you with them – whether you like it or not!

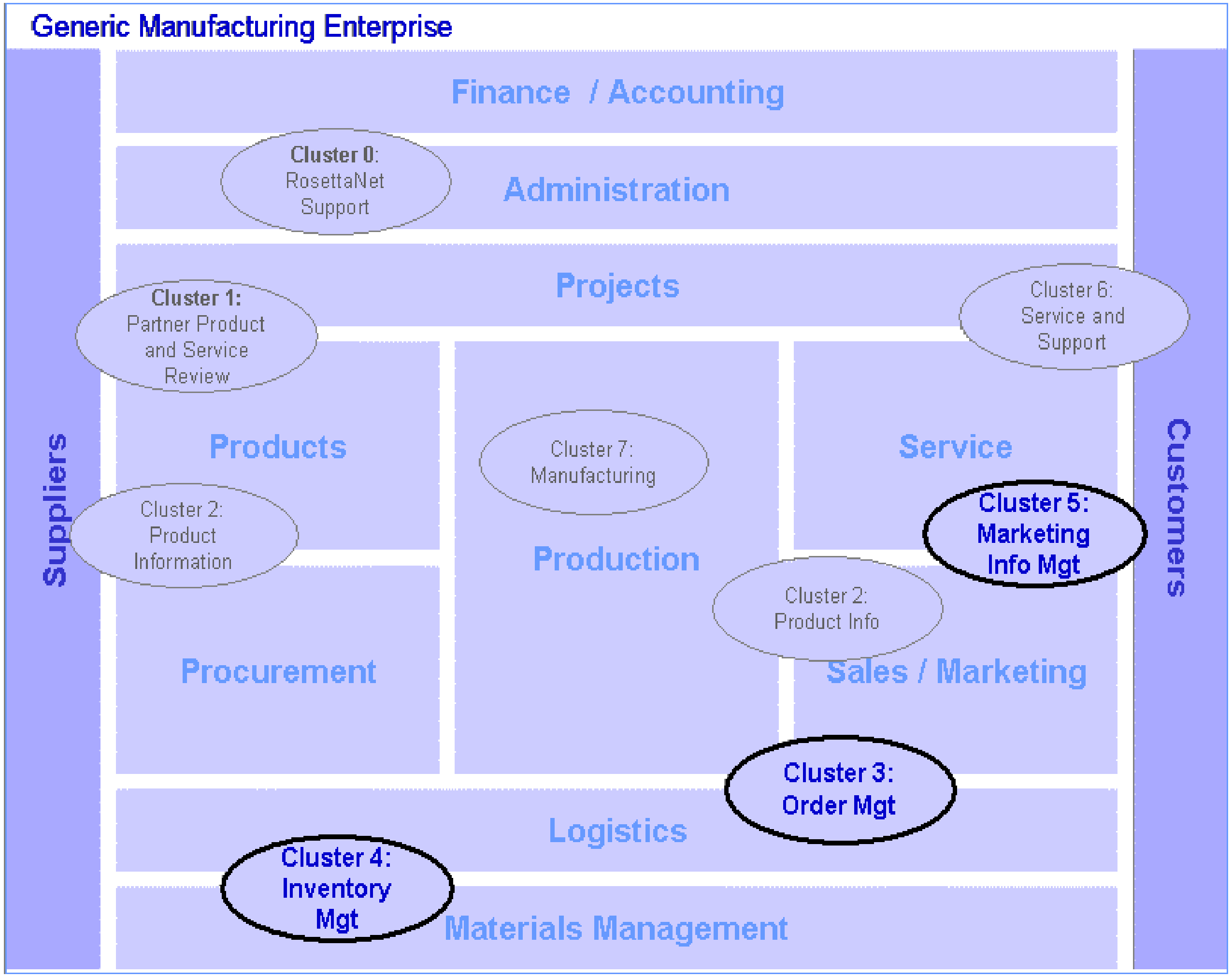
  - Intel: "…13 percent of all our machine-to-machine, B2B transactions today are happening through the (RosettaNet) standard – roughly 30,000 transactions a month."

Back to RosettaNet…

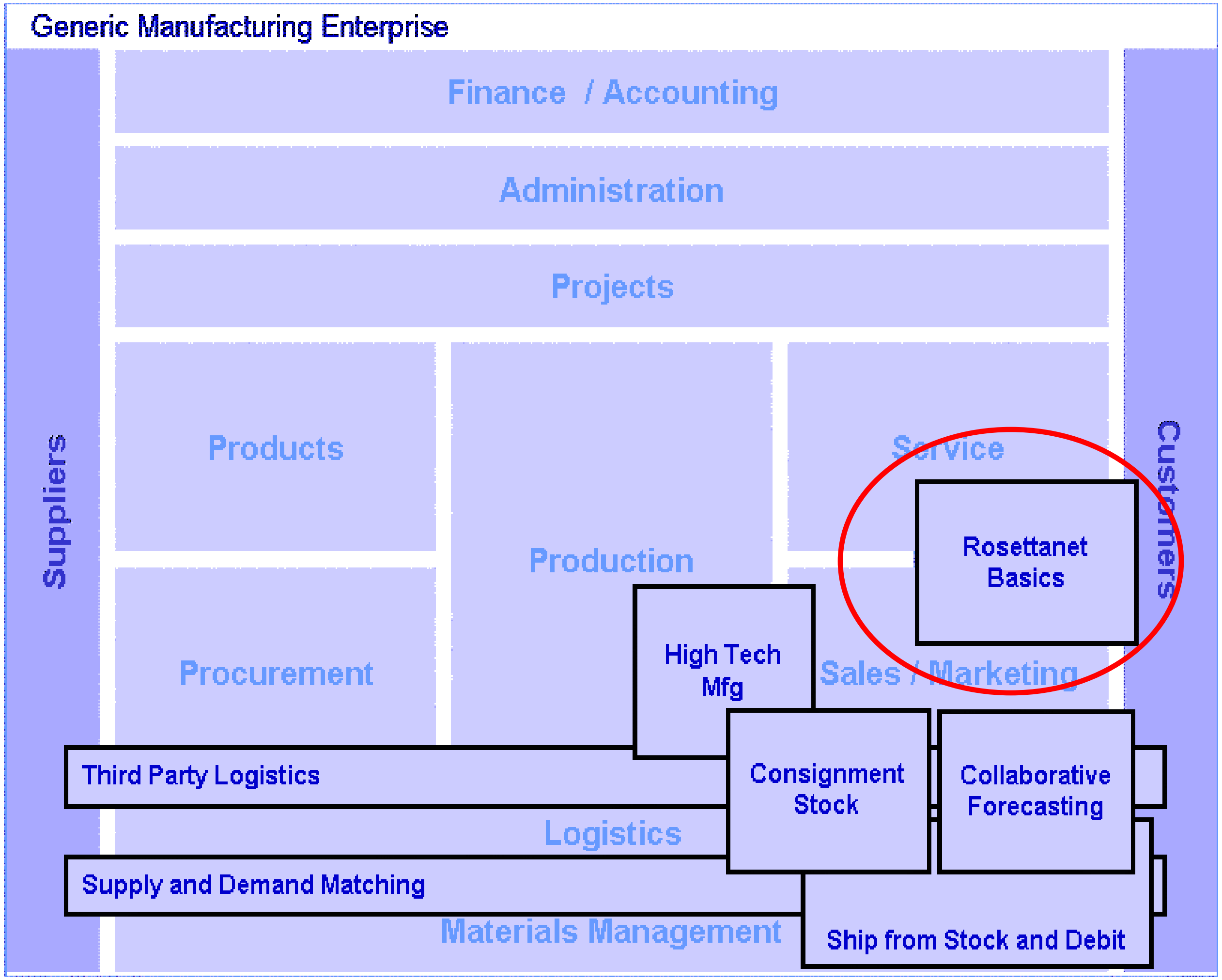
- Composed of Clusters, Segments, and Partner Interface Processes (PIPs)

- Example:
  - Cluster 3: Order Management
    - Segment 3C: Returns and Finance
      - PIP 3C3: Notify of Invoice

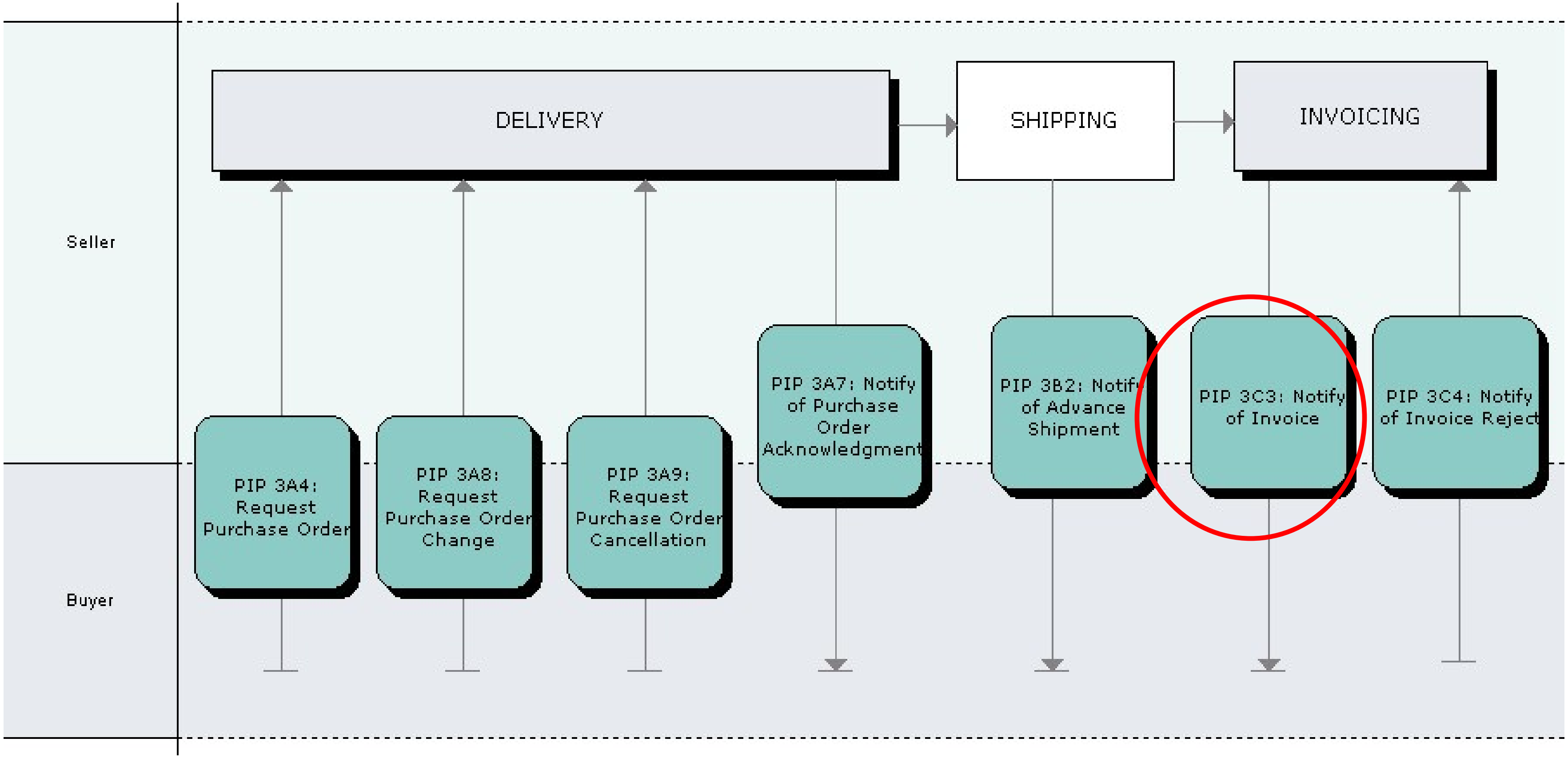
Drilling Into Our Example...



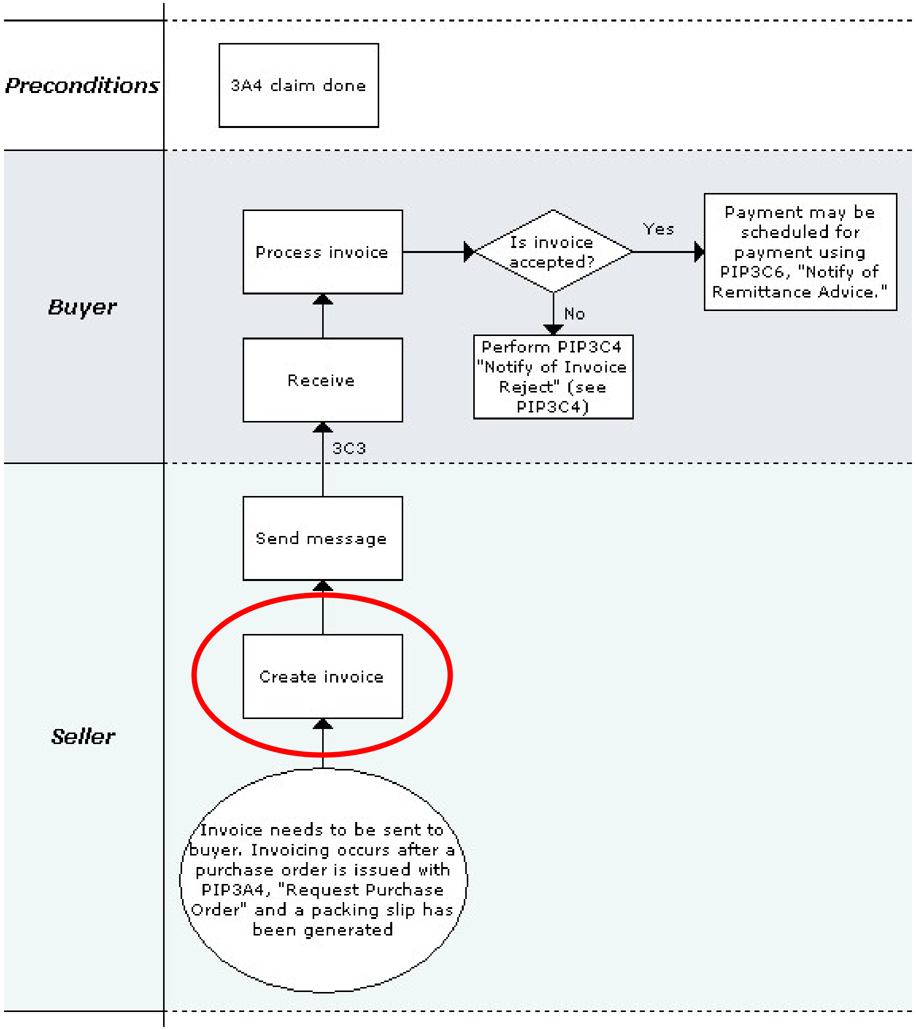
Drilling Into Our Example…



Drilling Into Our Example...

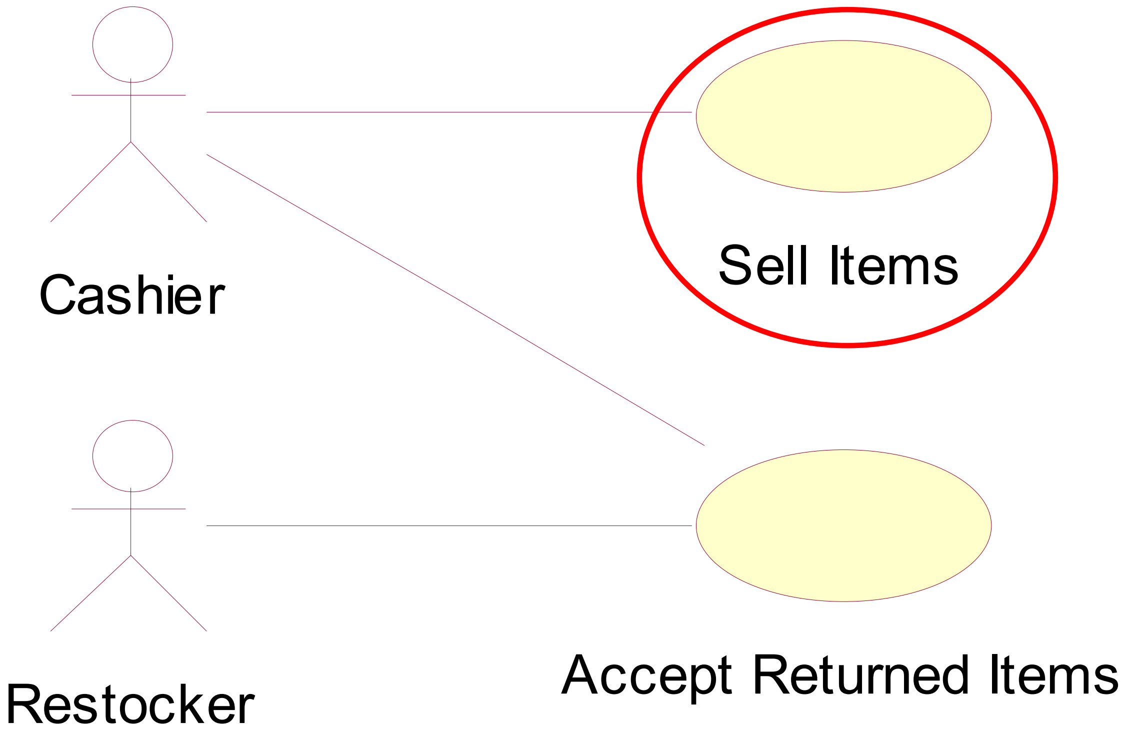

DELIVERY
SHIPPING
INVOICING
Seller
Buyer
PIP 3A4: Request Purchase Order
PIP 3A8: Request Purchase Order Change
PIP 3A9: Request Purchase Order Cancellation
PIP 3A7: Notify of Purchase Order Acknowledgment
PIP 3B2: Notify of Advance Shipment
PIP 3C3: Notify of Invoice
PIP 3C4: Notify of Invoice Reject

Drilling Into Our Example…

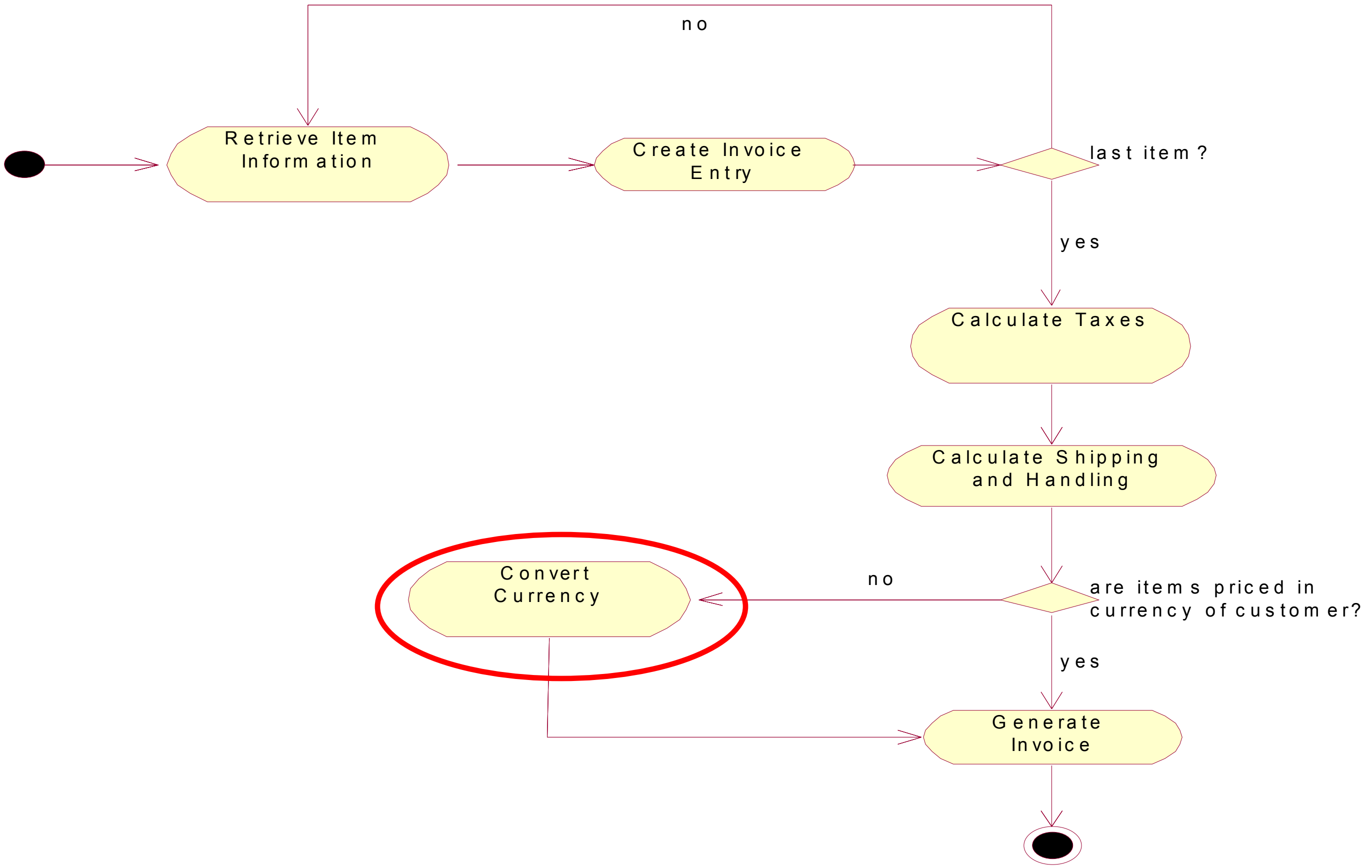


Preconditions
3A4 claim done
Buyer
Process invoice
Is invoice accepted?
Yes
Payment may be scheduled for payment using PIP3C6, "Notify of Remittance Advice."
No
Perform PIP3C4 "Notify of Invoice Reject" (see PIP3C4)
Receive
3C3
Seller
Send message
Create invoice
Invoice needs to be sent to buyer. Invoicing occurs after a purchase order is issued with PIP3A4, "Request Purchase Order" and a packing slip has been generated

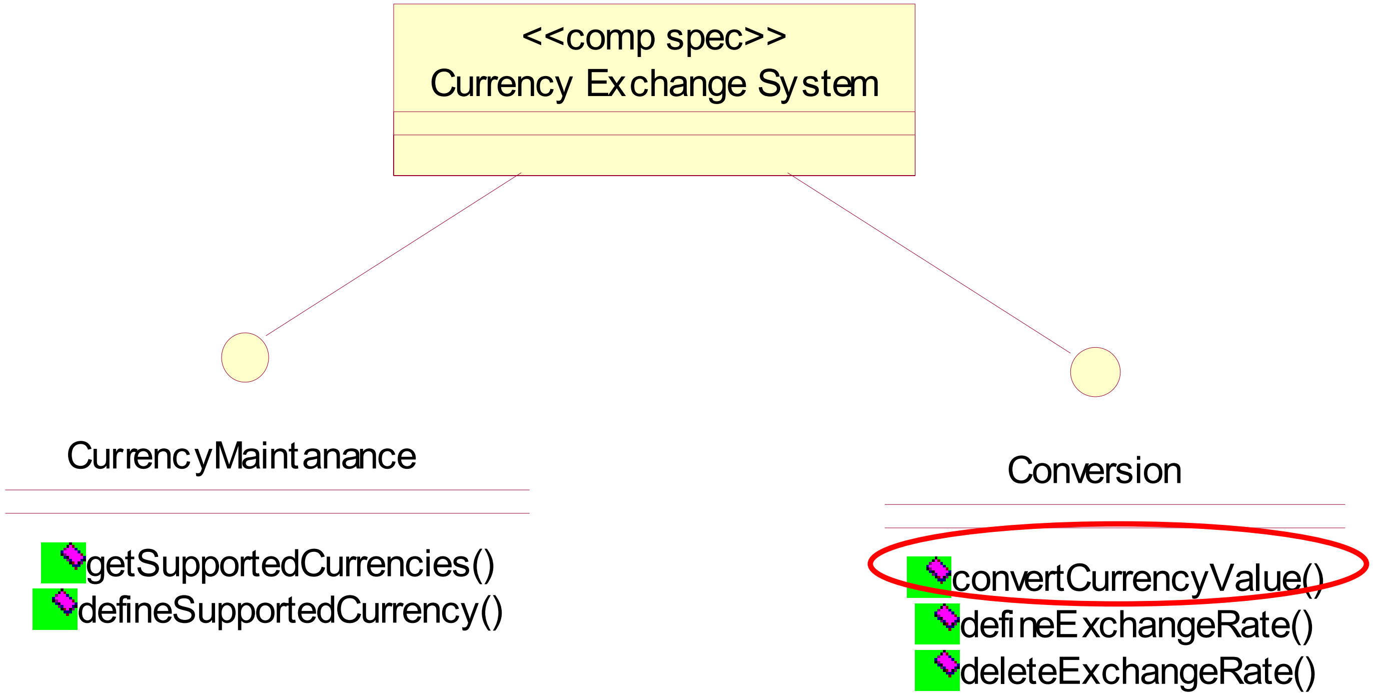
…Bringing Us Back To Our Original Use Case



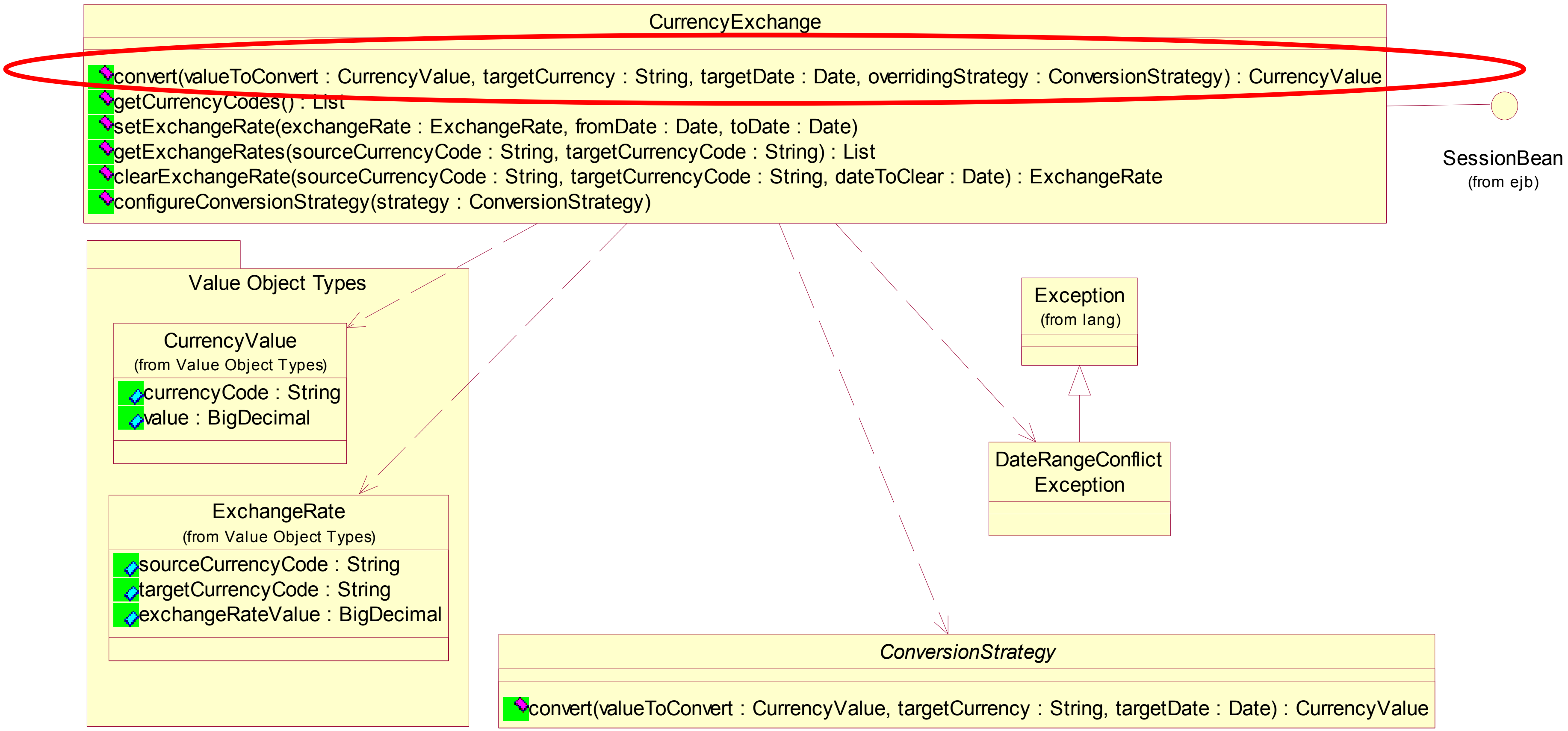
…Bringing Us Back To Our Original Use Case


no
Retrieve Item Information
Create Invoice Entry
last item ?
yes
Calculate Taxes
Calculate Shipping and Handling
Convert Currency
no
are items priced in currency of customer?
yes
Generate Invoice

...And Our Original Component Model



...And Finally To Our Original Component!


CurrencyExchange
convert(valueToConvert : CurrencyValue, targetCurrency : String, targetDate : Date, overridingStrategy : ConversionStrategy) : CurrencyValue
getCurrencyCodes() : List
setExchangeRate(exchangeRate : ExchangeRate, fromDate : Date, toDate : Date)
getExchangeRates(sourceCurrencyCode : String, targetCurrencyCode : String) : List
clearExchangeRate(sourceCurrencyCode : String, targetCurrencyCode : String, dateToClear : Date) : ExchangeRate
configureConversionStrategy(strategy : ConversionStrategy)
SessionBean
(from ejb)
Value Object Types
CurrencyValue
(from Value Object Types)
currencyCode : String
value : BigDecimal
ExchangeRate
(from Value Object Types)
sourceCurrencyCode : String
targetCurrencyCode : String
exchangeRateValue : BigDecimal
Exception
(from lang)
DateRangeConflict
Exception
ConversionStrategy
convert(valueToConvert : CurrencyValue, targetCurrency : String, targetDate : Date) : CurrencyValue

So What?

- Business processes matter to an IT organization!
  - After all, you are “in business” to support the business (and not the other way around…)
- Defining the right business processes is hard work
  - Most organizations do not have a clear view of how they do business
  - Using standards-based processes (in part or in whole) can help to rationalize the business
  - Don’t try to “boil the ocean” – pick and choose the most important/dynamic areas of business to focus on
- Linking business processes to existing and new technology is also hard work
  - Web services (and service-oriented architectures) can help
  - Tools can help

Which leads us to our best practices and case study…

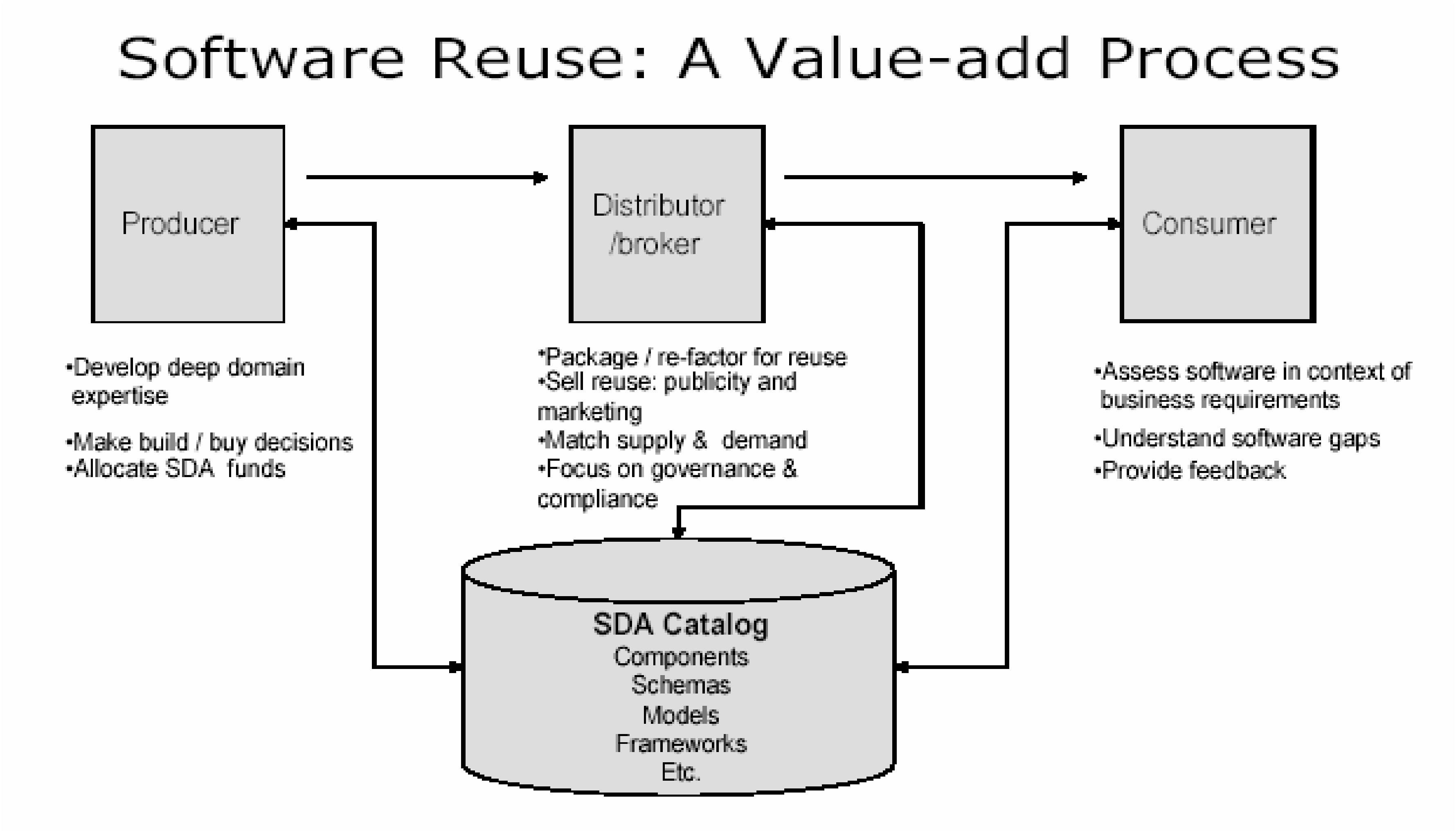
LogicLibrary

SDA Reuse Best Practices

SDA Reuse Best Practices

- SDA reuse is supported by production/distribution/consumption roles
  - **Production:** Identification and preparation of existing and newly-defined candidate reusable SDAs
  - **Distribution:** Publication of those SDAs into SDA library
  - **Consumption:** Use of SDA library to discover appropriate reusable SDAs on a per-project basis
- SDA reuse effectiveness is assessed by management governance and ROI review

SDA Reuse Lifecycle



Software Reuse: A Value-add Process

- Develop deep domain expertise
- Make build / buy decisions
- Allocate SDA funds

- Package / re-factor for reuse
- Sell reuse: publicity and marketing
- Match supply & demand
- Focus on governance & compliance

- Assess software in context of business requirements
- Understand software gaps
- Provide feedback

Tools Enablement of SDA Reuse

- Traditionally reuse has occurred via **word-of-mouth**
  - Effective for very small groups but doesn't scale
- Various **tool-based attempts** have been made in the past, including:
  - Spreadsheet "registries"
  - Full CASE tooling
  - Version control repositories
  - Ad-hoc collaboration databases
- New **SDA metadata repository** space has emerged in past 2-3 years
  - Gartner March 2004 report and Magic Quadrant on Metadata Repositories
  - These **tools**, plus **rationalization** of development space into **.NET** and **J2EE** component and service architectures and dramatically **improved IDE tooling** (e.g., VS, WSAD, Eclipse) enable for the first time effective SDA reuse practices and governance

SDA Production

- Best practice recommendation:
  virtual/matrixed SDA architectural review team
  - **Team members:**
    - ***Team leader*** dedicated to SDA reuse program
    - ***Matrixed team members*** drawn from project teams
      - Lead designer/developer skills required
      - 10%-20% job responsibility
  - **Team objectives/responsibilities:**
    - ***Identify*** candidate reusable SDAs – "active discovery"
    - ***Review*** proposed reusable SDAs – asset hardening
      - Adherence to architecture
      - Necessary functionality implemented and supported
      - Mandatory artifacts provided
    - ***Publish*** approved SDAs into SDA library for consumption
    - ***Recommend*** future resource allocation for key reusable SDAs
      - Expanded funding for key SDAs
      - Transfer of key SDAs to common SDA support group

SDA Distribution

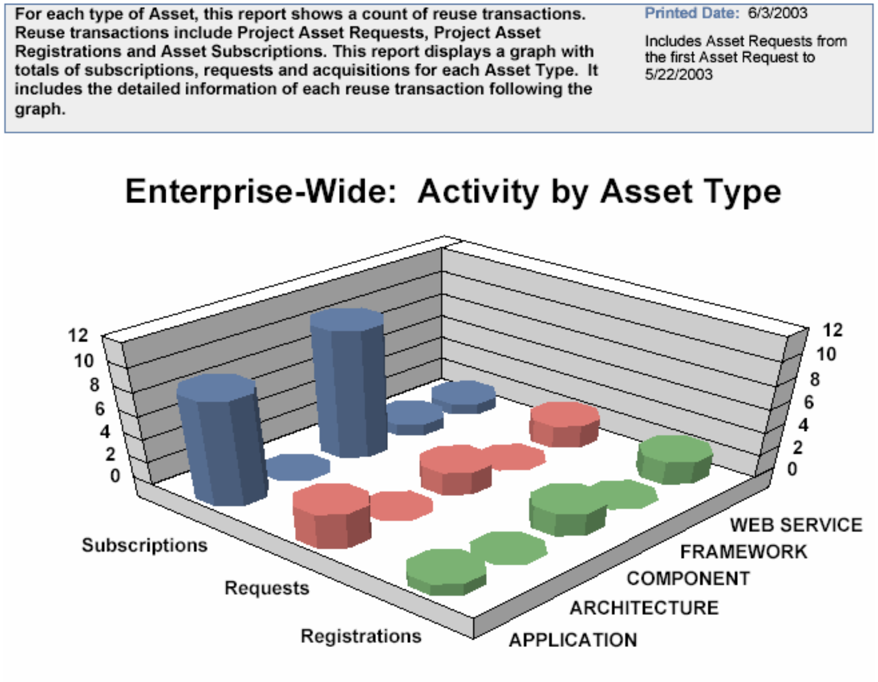
- Best practice recommendation:
  SDA library (e.g., Logidex) to distribute assets
  - Asset metadata ***assembly*** and ***validation***
    - Standardized metadata definition
    - Per-asset-type metadata validation and enforcement
  - Asset publication
    - Newly defined SDAs
    - Updated SDAs
    - New versions of existing SDAs

SDA Consumption Modes

- Best practice recommendation: ***named user consumption***
    - Project-scoped interactions/tracking
    - SDA acquisition
        - Project Manager and Asset Publisher approval
    - Project and asset-specific collaboration
        - Discussion forums
        - Persistent searches
        - Asset notifications
- Access alternative: ***anonymous consumption***
    - Suitable for lightweight/casual users
    - Read-only interaction with library
    - Restricted tracking and collaboration activities

SDA Reuse Management Example: Charles Schwab

- SDA ROI reporting examples:
  - “Big Picture” rollup
  - Project-specific SDA reuse ROI

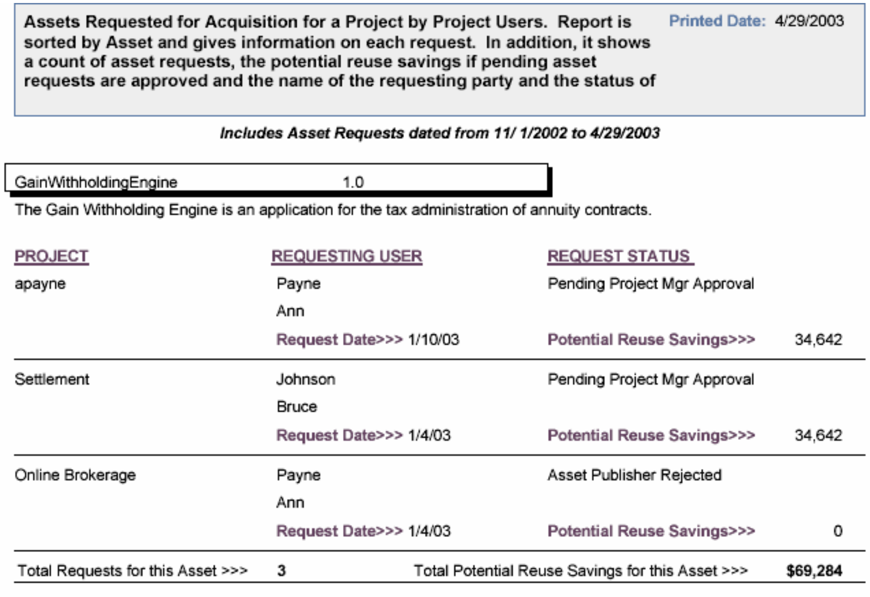


For each type of Asset, this report shows a count of reuse transactions. Reuse transactions include Project Asset Requests, Project Asset Registrations and Asset Subscriptions. This report displays a graph with totals of subscriptions, requests and acquisitions for each Asset Type. It includes the detailed information of each reuse transaction following the graph.

Printed Date: 6/3/2003

Includes Asset Requests from the first Asset Request to 5/22/2003

Enterprise-Wide: Activity by Asset Type



Assets Requested for Acquisition for a Project by Project Users. Report is sorted by Asset and gives information on each request. In addition, it shows a count of asset requests, the potential reuse savings if pending asset requests are approved and the name of the requesting party and the status of

Printed Date: 4/29/2003

Includes Asset Requests dated from 11/ 1/2002 to 4/29/2003

GainWithholdingEngine 1.0

The Gain Withholding Engine is an application for the tax administration of annuity contracts.

PROJECT	REQUESTING USER	REQUEST STATUS	
apayne	Payne Ann Request Date>>> 1/10/03	Pending Project Mgr Approval Potential Reuse Savings>>>	34,642
Settlement	Johnson Bruce Request Date>>> 1/4/03	Pending Project Mgr Approval Potential Reuse Savings>>>	34,642
Online Brokerage	Payne Ann Request Date>>> 1/4/03	Asset Publisher Rejected Potential Reuse Savings>>>	0
Total Requests for this Asset >>>	3	Total Potential Reuse Savings for this Asset >>>	**$69,284**

SDA Reuse ROI Example

Logidex Return On Investment (ROI)

ABC Company	Year 1	Year 2	Year 3
Environmental Data			
Total Research and Development Cost (est.)	$3,750,000	$7,725,000	$11,935,125
Total Number of Developers	25	50	75
Loaded Developer Cost	$3,750,000	$7,725,000	$11,935,125
Logidex Investment			
Cost of Logidex and the implementation	$ 70,000	$ 35,800	$ 40,300
Estimated ROI			
Savings from reusing components	$1,035,750	$3,362,400	$6,902,784
Additional savings from using Logidex to help write reusable assets	$11,250	$69,525	$107,416
Additional savings from using Logidex to find and understand assets	$144,000	$444,960	$916,618
Total Cost Savings	**$ 1,191,000**	**$ 3,876,885**	**$ 7,926,818**
Incremental Logidex Payback Time (in months)	**6.30**	**5.90**	**5.04**

The LogicLibrary Reuse Calculator is based on the reuse metric terms, return-on-investment (ROI) model, and “ReuCalc” calculator developed by Dr. Jeffrey S. Poulin. Details of this work appear in Dr. Poulin's book, Measuring Software Reuse: Principles, Practices, and Economic Models, published by Addison-Wesley (ISBN 0-201-63413-9) in 1997.

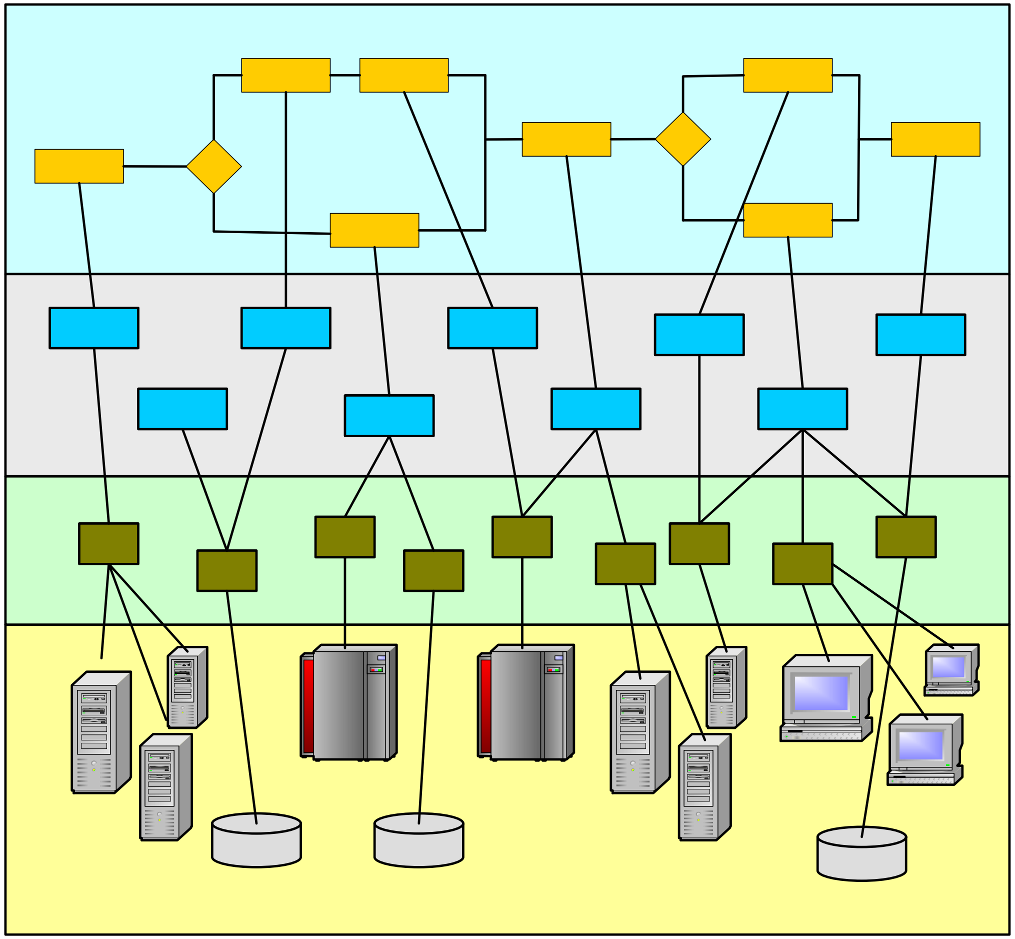
LogicLibrary

SDA Management/Governance Case Study

Services Lifecycle Case Study: CNA Insurance

- CNA Insurance
  - Global Insurance organization
  - 12 billion in revenue
  - 15,000 employees total, 1,600 in IT workforce
- CNA's Service-oriented Architecture Initiative
  - Led by Dmitry Tyomkin and Boris Lublinsky,
    Enterprise Architects, CNA's Solutions and Architecture Group
  - Objectives:
    - Build bridges between different systems and applications
    - Leverage existing IT assets
    - Create ability to respond quickly and efficiently to changes in the business environment

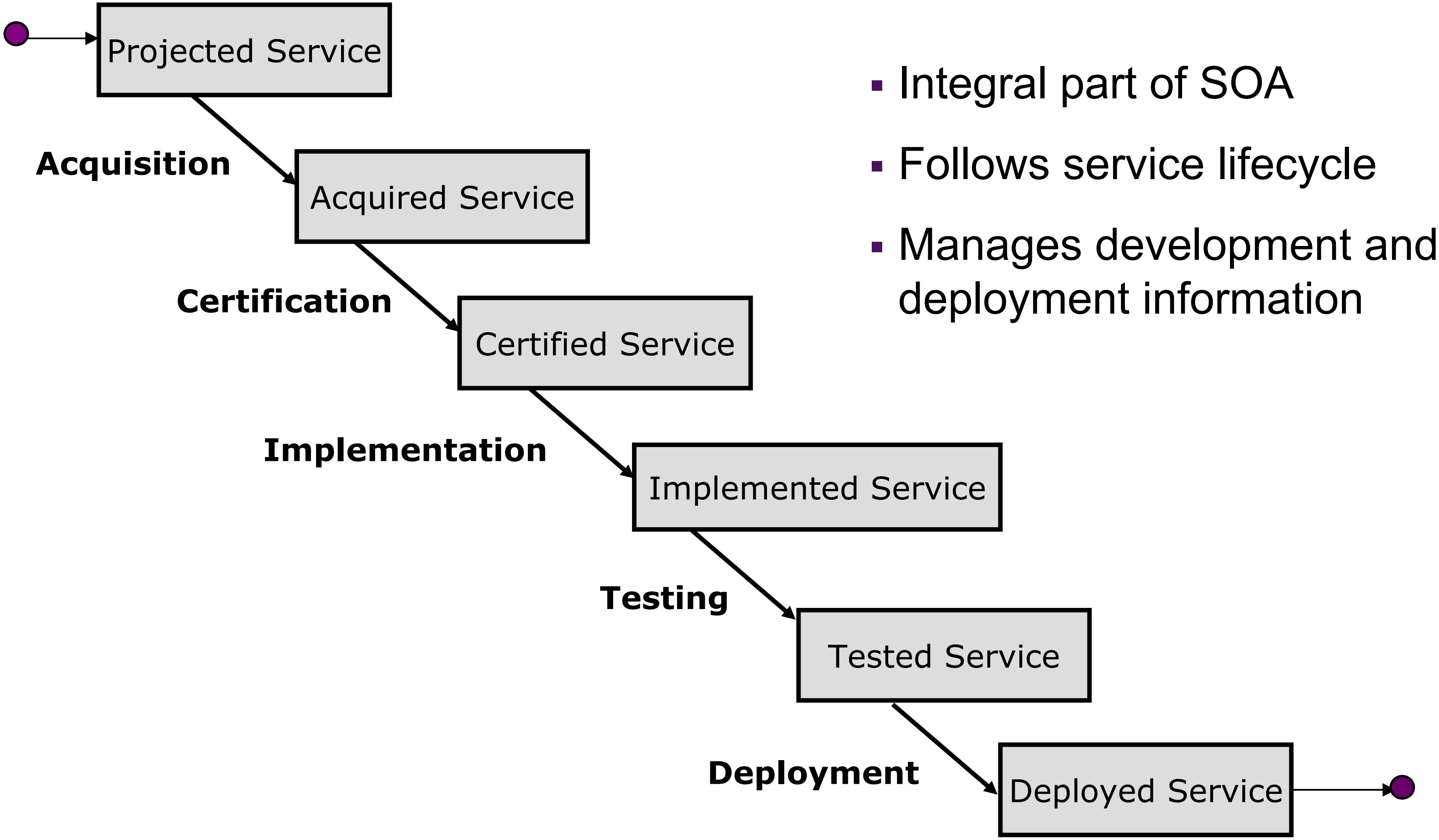
SOA in the Enterprise Architecture – CNA's View



Current SOA Components in CNA

- Consolidated Infrastructure
- Enterprise Service Bus
  - Integration Hub
  - Service Invocation and Execution Framework
  - Service Locator
- Services Implementation Framework
- Services Classification

CNA’s Service Catalog: LogicLibrary Logidex



- Integral part of SOA
- Follows service lifecycle
- Manages development and deployment information

Service Composition

- Services and Methods are managed independently
  - Services are composed of one or more Methods
  - Methods may be composed within one or more Services
    - Method will typically be composed within a single Service
  - Relationships between Services and Methods are managed with strongly-typed asset-to-asset relationships within Logidex-managed metadata
- Appropriate Methods identified as new Services are defined and projected
  - Projection process includes:
    - Selecting from predefined Methods
    - Specifying new candidate Methods
- Method-level granularity within Logidex enables:
  - More granular usage tracking
  - Flexible service composition based on domain needs

LogicLibrary™

Logidex Introduction

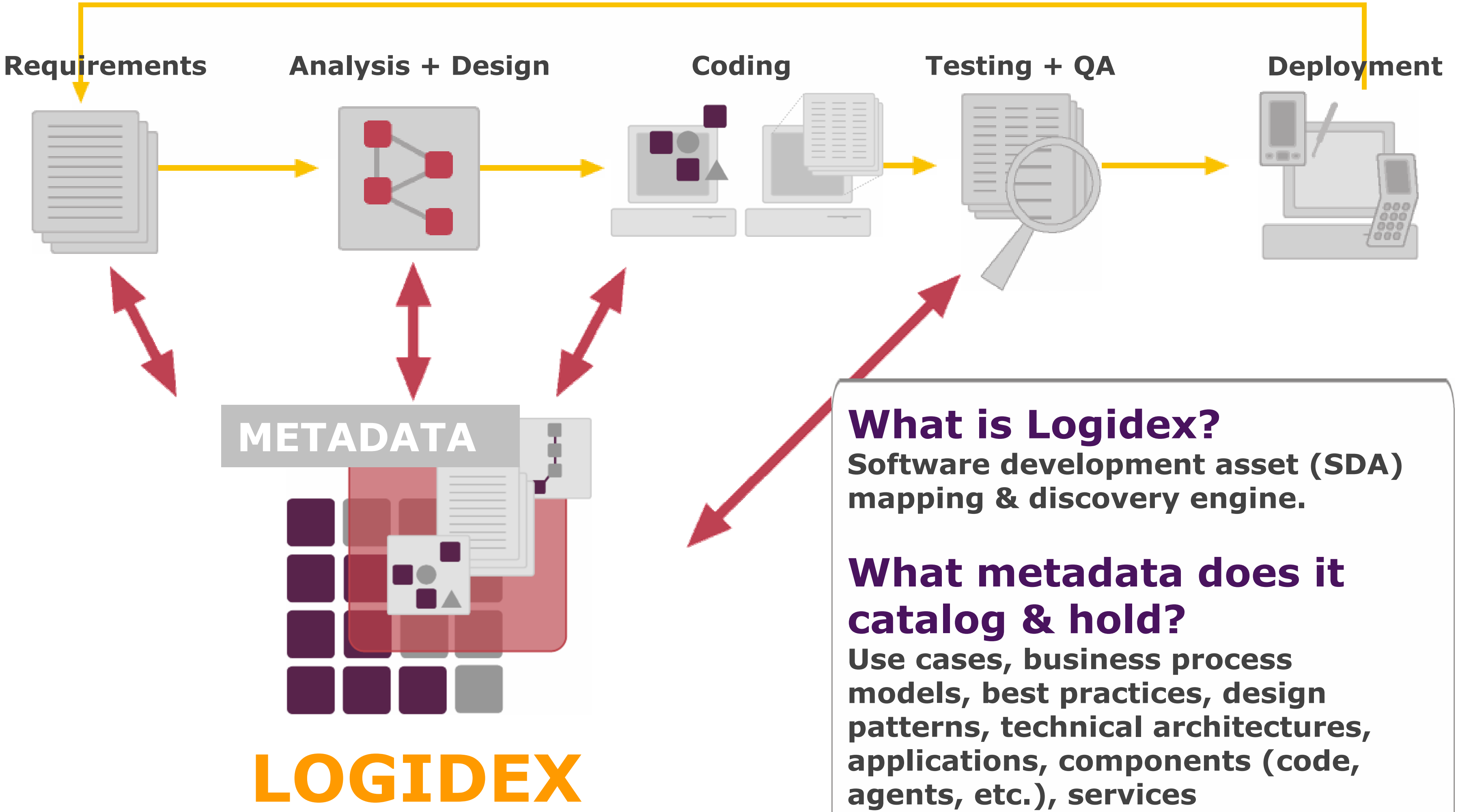
LogicLibrary Logidex

- What it is:
  - Catalog of software development asset (SDA) metadata
- Why it's different:
  - Powerful mapping, discovery and collaboration engine for identifying and using SDAs in all types of software development and integration

Clear Value Proposition:

Logidex enables faster and less costly consolidation, migration and/or integration of enterprise applications

Application Development Lifecycle: With Logidex



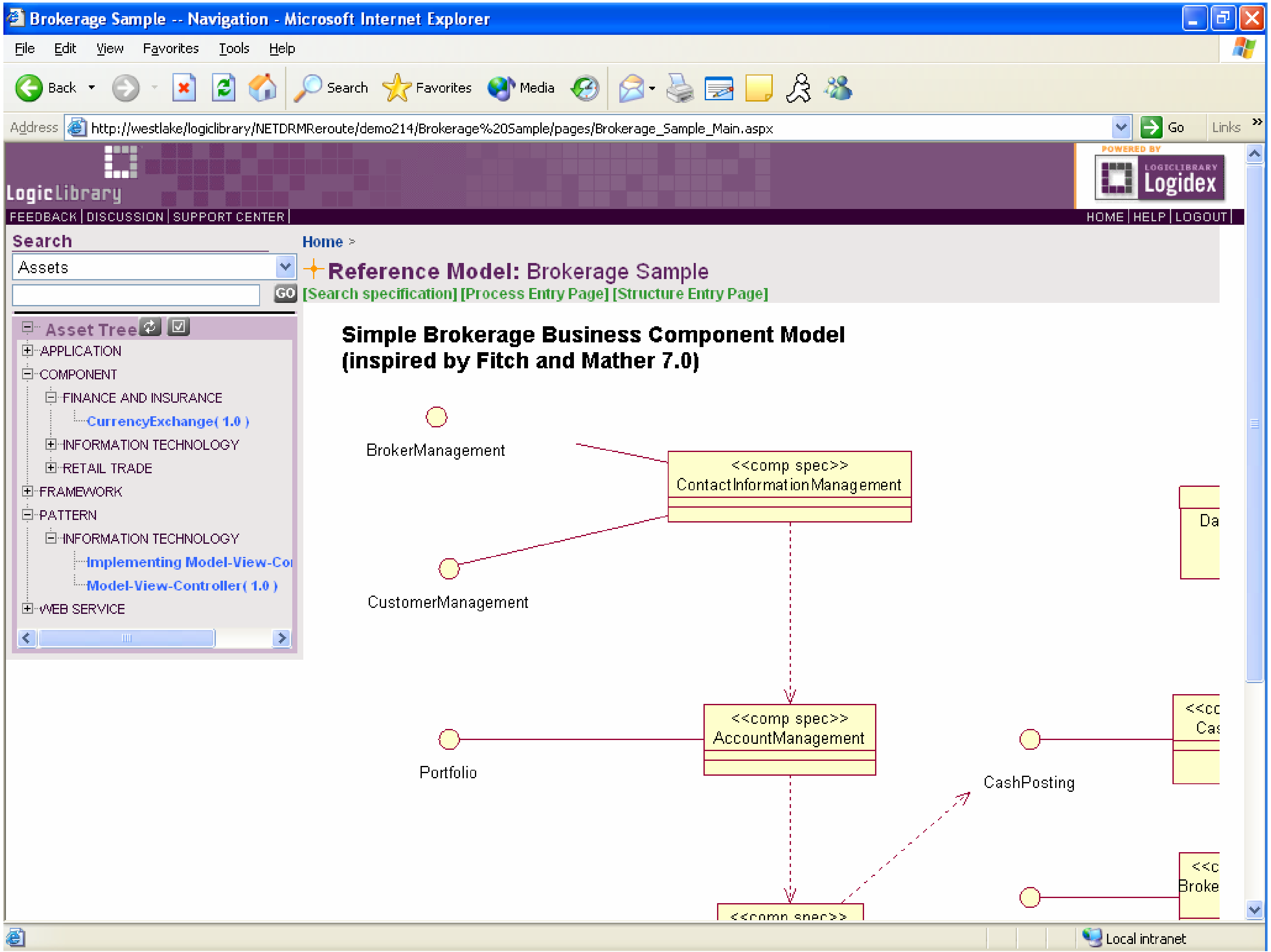
What is Logidex?

Software development asset (SDA) mapping & discovery engine.

What metadata does it catalog & hold?

Use cases, business process models, best practices, design patterns, technical architectures, applications, components (code, agents, etc.), services documentation, test cases

Logidex thin-client



Brokerage Sample -- Navigation - Microsoft Internet Explorer

Address: http://westlake/logiclibrary/NETDRMReroute/demo214/Brokerage%20Sample/pages/Brokerage_Sample_Main.aspx

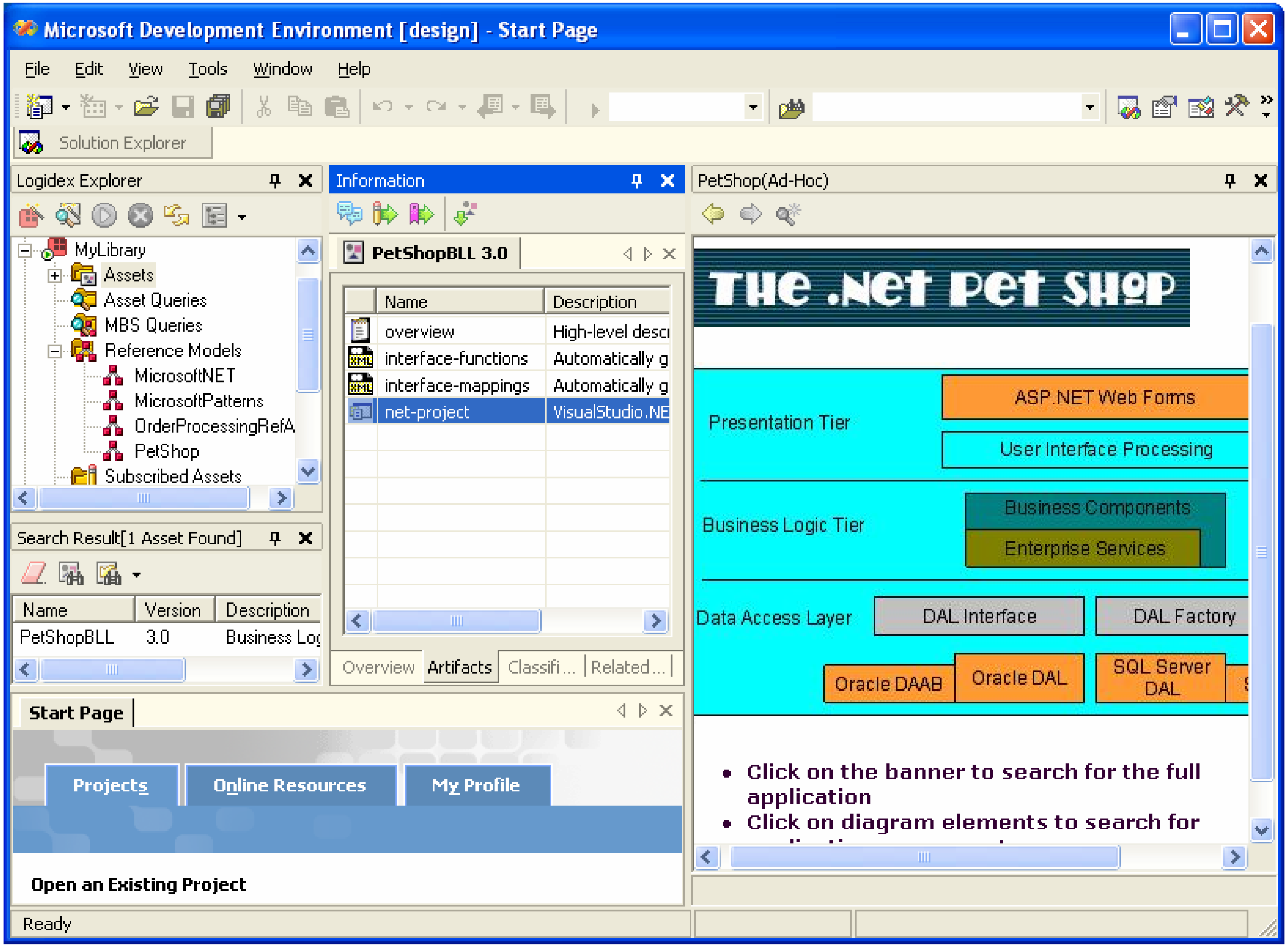
Reference Model: Brokerage Sample

[Search specification] [Process Entry Page] [Structure Entry Page]

Simple Brokerage Business Component Model (inspired by Fitch and Mather 7.0)

©2004 LogicLibrary, Inc.

Logidex Visual Studio .NET Add-in



Microsoft Development Environment [design] - Start Page

- MyLibrary
  - Assets
  - Asset Queries
  - MBS Queries
  - Reference Models
    - MicrosoftNET
    - MicrosoftPatterns
    - OrderProcessingRefA
    - PetShop
  - Subscribed Assets

Search Result[1 Asset Found]

Name	Version	Description
PetShopBLL	3.0	Business Lo

PetShopBLL 3.0

Name	Description
overview	High-level desc
interface-functions	Automatically g
interface-mappings	Automatically g
net-project	VisualStudio.NE

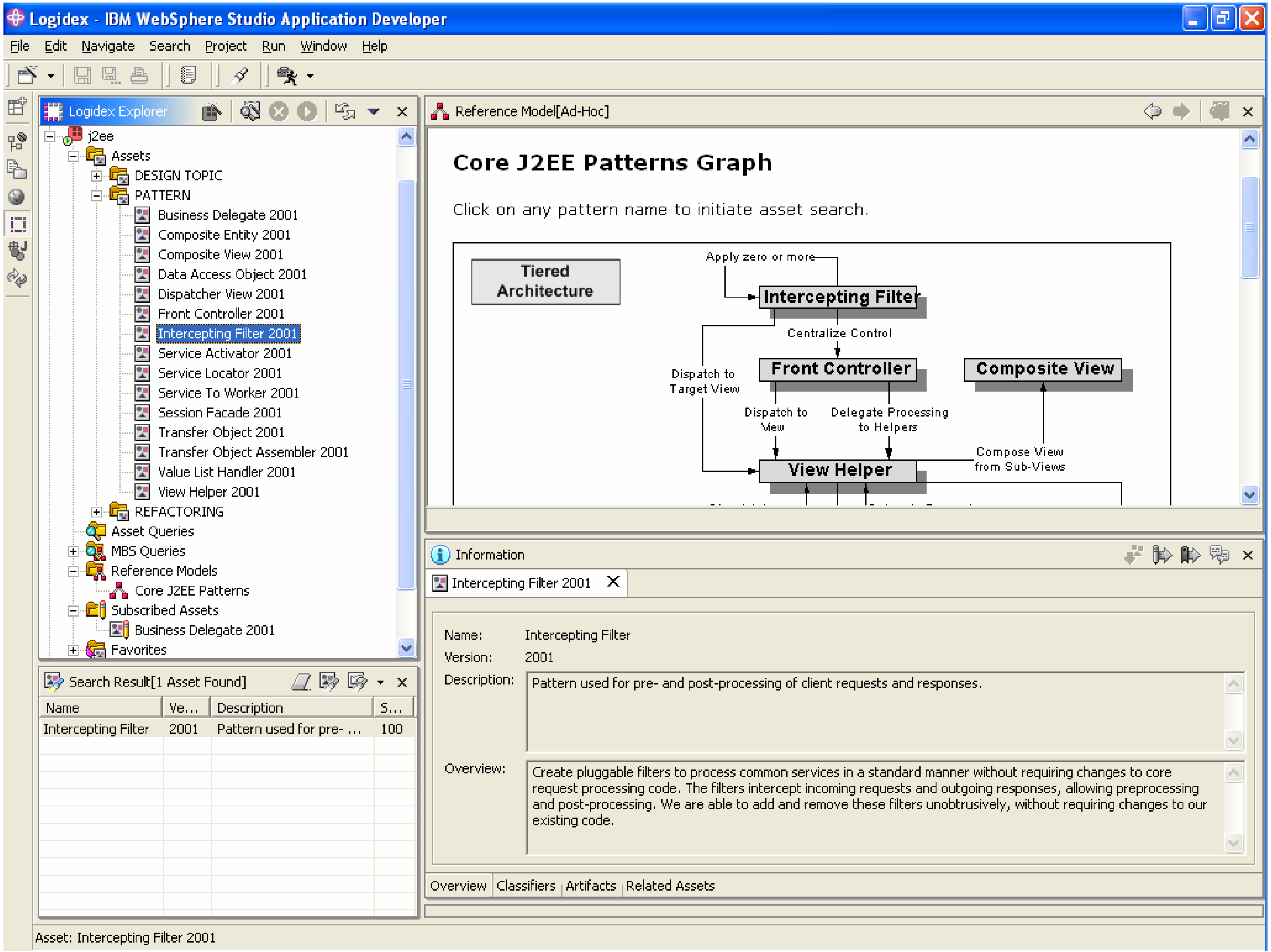
Open an Existing Project

Ready

The .Net Pet Shop

- Click on the banner to search for the full application
- Click on diagram elements to search for

Logidex WSAD/Eclipse plug-in



- j2ee
  - Assets
    - DESIGN TOPIC
    - PATTERN
      - Business Delegate 2001
      - Composite Entity 2001
      - Composite View 2001
      - Data Access Object 2001
      - Dispatcher View 2001
      - Front Controller 2001
      - Intercepting Filter 2001
      - Service Activator 2001
      - Service Locator 2001
      - Service To Worker 2001
      - Session Facade 2001
      - Transfer Object 2001
      - Transfer Object Assembler 2001
      - Value List Handler 2001
      - View Helper 2001
    - REFACTORING
  - Asset Queries
  - MBS Queries
  - Reference Models
    - Core J2EE Patterns
  - Subscribed Assets
    - Business Delegate 2001
  - Favorites

Core J2EE Patterns Graph

Click on any pattern name to initiate asset search.

Name	Ve...	Description	S...
Intercepting Filter	2001	Pattern used for pre- ...	100

Intercepting Filter 2001

Name: Intercepting Filter

Version: 2001

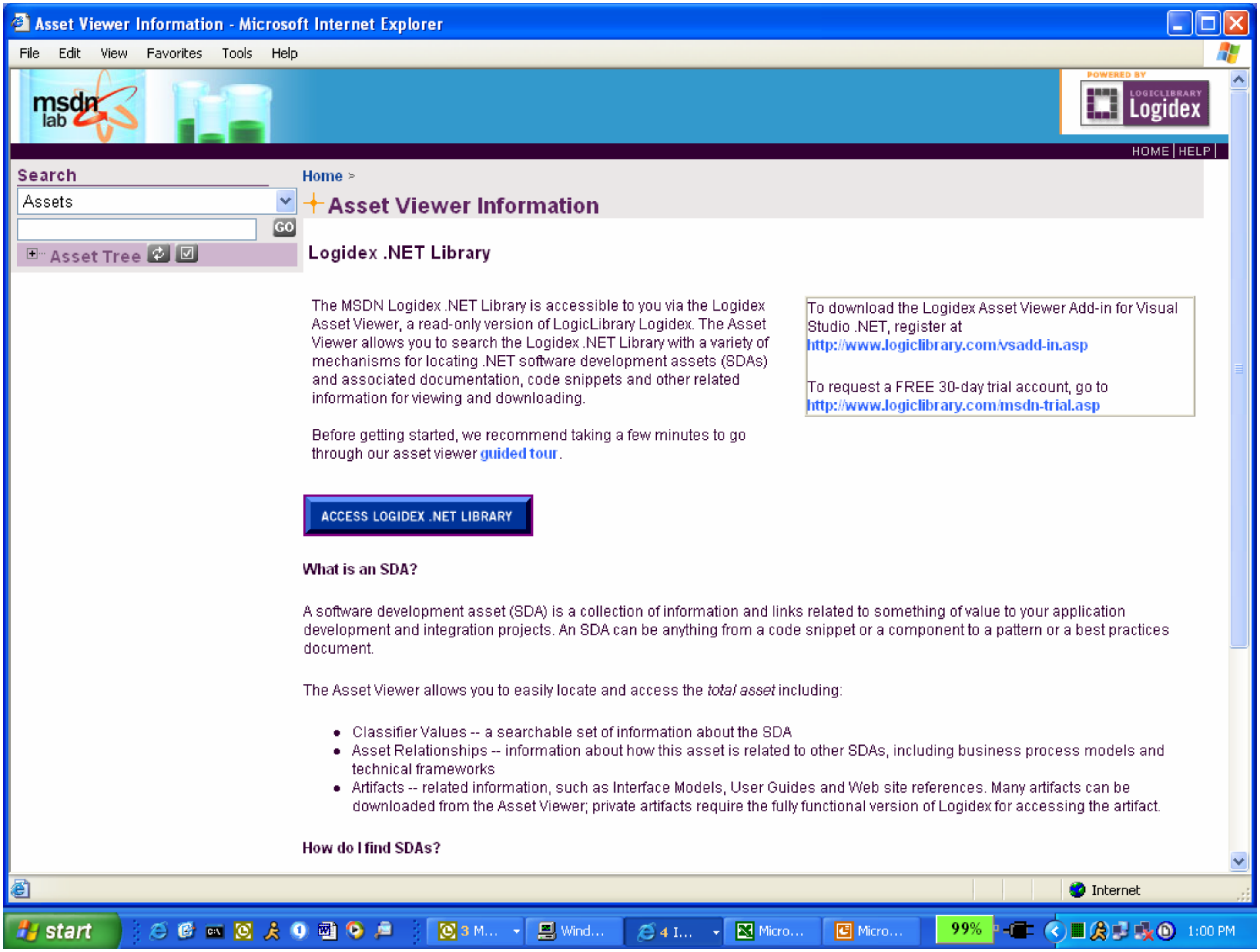
Description: Pattern used for pre- and post-processing of client requests and responses.

Overview: Create pluggable filters to process common services in a standard manner without requiring changes to core request processing code. The filters intercept incoming requests and outgoing responses, allowing preprocessing and post-processing. We are able to add and remove these filters unobtrusively, without requiring changes to our existing code.

Asset: Intercepting Filter 2001

How can you try Logidex?

MSDN Logidex .NET Library



Asset Viewer Information - Microsoft Internet Explorer
File Edit View Favorites Tools Help
msdn lab
POWERED BY LOGICLIBRARY Logidex
HOME | HELP |
Search
Assets
GO
Asset Tree
Home >
Internet
start
1:00 PM

Asset Viewer Information

Logidex .NET Library

The MSDN Logidex .NET Library is accessible to you via the Logidex Asset Viewer, a read-only version of LogicLibrary Logidex. The Asset Viewer allows you to search the Logidex .NET Library with a variety of mechanisms for locating .NET software development assets (SDAs) and associated documentation, code snippets and other related information for viewing and downloading.

Before getting started, we recommend taking a few minutes to go through our asset viewer guided tour.

To download the Logidex Asset Viewer Add-in for Visual Studio .NET, register at http://www.logiclibrary.com/vsadd-in.asp

To request a FREE 30-day trial account, go to http://www.logiclibrary.com/msdn-trial.asp

ACCESS LOGIDEX .NET LIBRARY

What is an SDA?

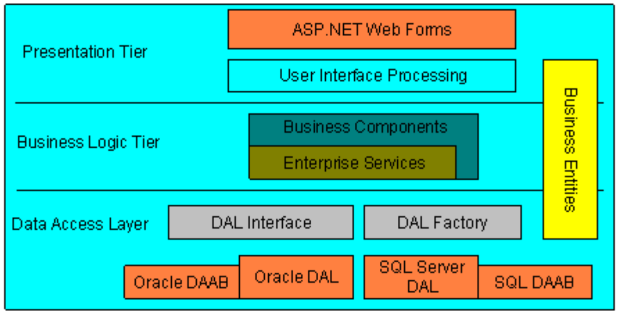
A software development asset (SDA) is a collection of information and links related to something of value to your application development and integration projects. An SDA can be anything from a code snippet or a component to a pattern or a best practices document.

The Asset Viewer allows you to easily locate and access the *total asset* including:

- Classifier Values -- a searchable set of information about the SDA
- Asset Relationships -- information about how this asset is related to other SDAs, including business process models and technical frameworks
- Artifacts -- related information, such as Interface Models, User Guides and Web site references. Many artifacts can be downloaded from the Asset Viewer; private artifacts require the fully functional version of Logidex for accessing the artifact.

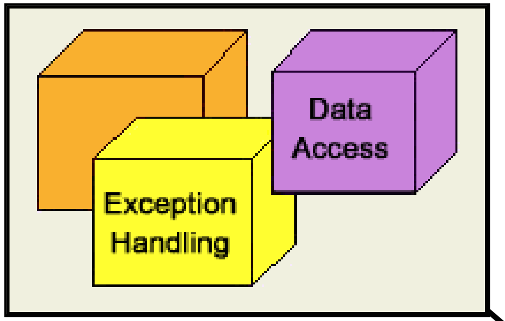
How do I find SDAs?

Logidex .NET Library Contents

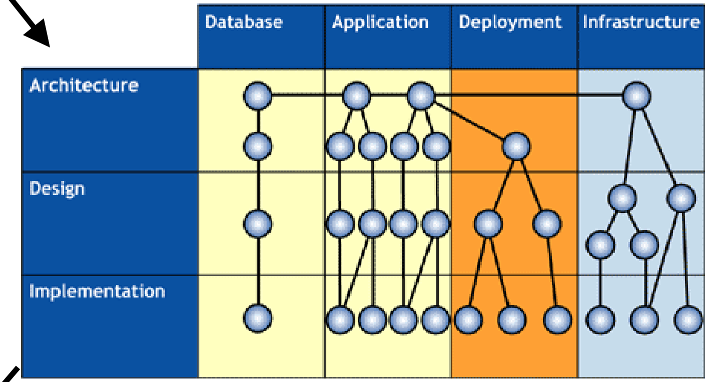


Reference Application

The .NET Pet Shop



Application Blocks

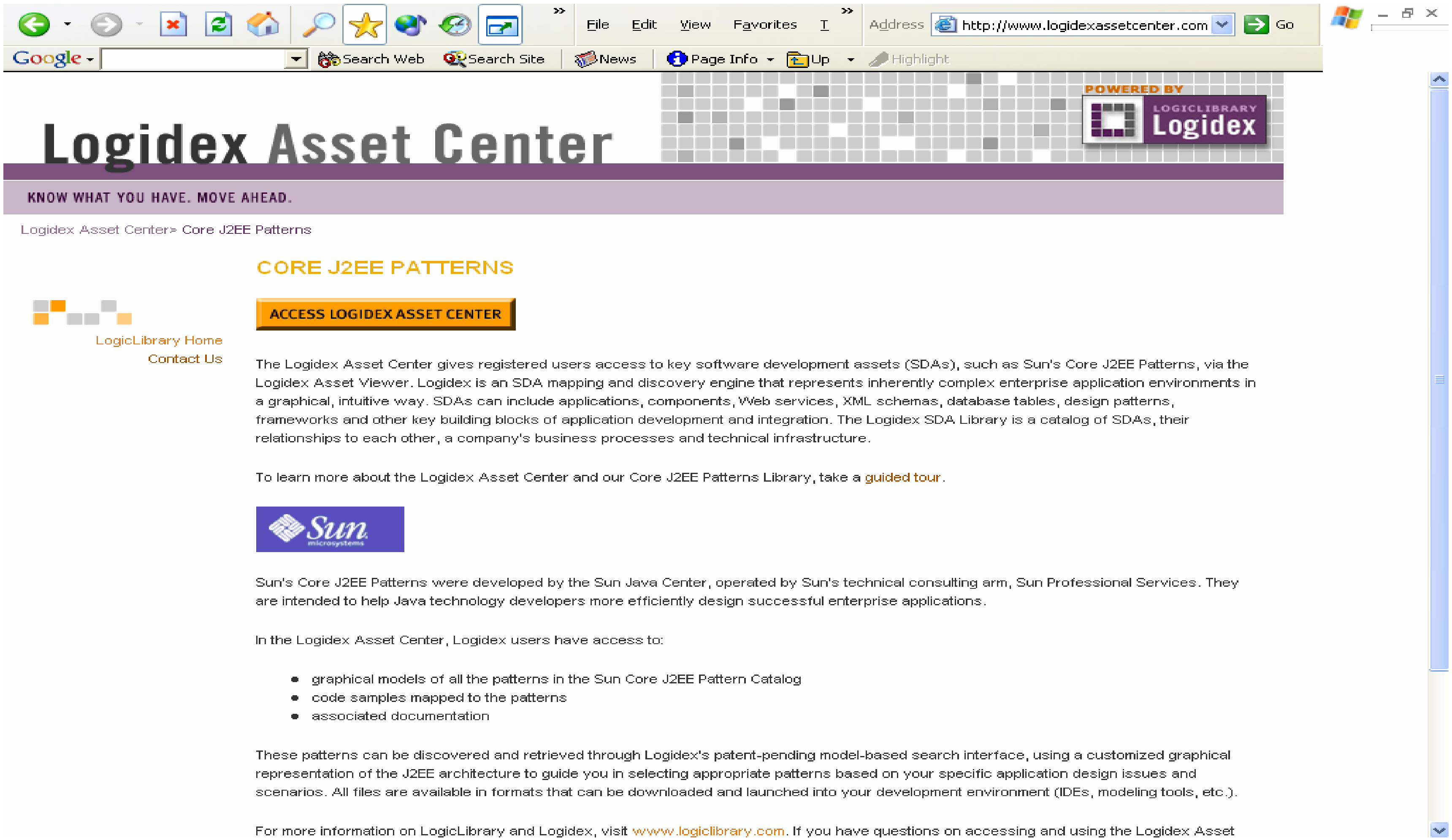


Enterprise Solution Patterns

Microsoft .net Framework

Logidex Asset Center

Core J2EE Patterns, RosettaNet PIPs, and FEA Reference Models and Assets



Logidex Asset Center

KNOW WHAT YOU HAVE. MOVE AHEAD.

Logidex Asset Center> Core J2EE Patterns

LogicLibrary Home
Contact Us

CORE J2EE PATTERNS

ACCESS LOGIDEX ASSET CENTER

The Logidex Asset Center gives registered users access to key software development assets (SDAs), such as Sun's Core J2EE Patterns, via the Logidex Asset Viewer. Logidex is an SDA mapping and discovery engine that represents inherently complex enterprise application environments in a graphical, intuitive way. SDAs can include applications, components, Web services, XML schemas, database tables, design patterns, frameworks and other key building blocks of application development and integration. The Logidex SDA Library is a catalog of SDAs, their relationships to each other, a company's business processes and technical infrastructure.

To learn more about the Logidex Asset Center and our Core J2EE Patterns Library, take a guided tour.

Sun microsystems

Sun's Core J2EE Patterns were developed by the Sun Java Center, operated by Sun's technical consulting arm, Sun Professional Services. They are intended to help Java technology developers more efficiently design successful enterprise applications.

In the Logidex Asset Center, Logidex users have access to:

- graphical models of all the patterns in the Sun Core J2EE Pattern Catalog
- code samples mapped to the patterns
- associated documentation

These patterns can be discovered and retrieved through Logidex's patent-pending model-based search interface, using a customized graphical representation of the J2EE architecture to guide you in selecting appropriate patterns based on your specific application design issues and scenarios. All files are available in formats that can be downloaded and launched into your development environment (IDEs, modeling tools, etc.).

For more information on LogicLibrary and Logidex, visit www.logiclibrary.com. If you have questions on accessing and using the Logidex Asset

Accessing Logidex

- MSDN Logidex .NET Library
  - Free access to thin client and Logidex add-in
    - Thin client: **http://lab.msdn.microsoft.com/logidex**
    - Add-in: **http://www.logiclibrary.com/vsadd-in.asp**
  - Trial account to put your own assets side by side with the MSDN .NET Assets in a private ASP-hosted library
    - **http://www.logiclibrary.com/msdn-trial.asp**

- Logidex Asset Center
  - http://www.logidexassetcenter.com

Thank you!