IBM Software Group

# Building Middleware for Services Oriented Architecture

## *IBM Software Group Flexibility Through Componentization*

ON DEMAND BUSINESS™

*Ciarán Anthony DellaFera*
*STSM, Chief Architect SWG Componentization*

(C) Copyright IBM Corporation 2004, 2005

Copyright IBM Corporation 2004, 2005. All Rights Reserved.

The information in this presentation is IBM Confidential. It must not be disclosed to customers unless they have a Non-Disclosure Agreement with IBM.

This presentation should be treated as VENDOR CONFIDENTIAL under the practices in place in your firm and under any existing agreements with IBM regarding disclosure of confidential information.

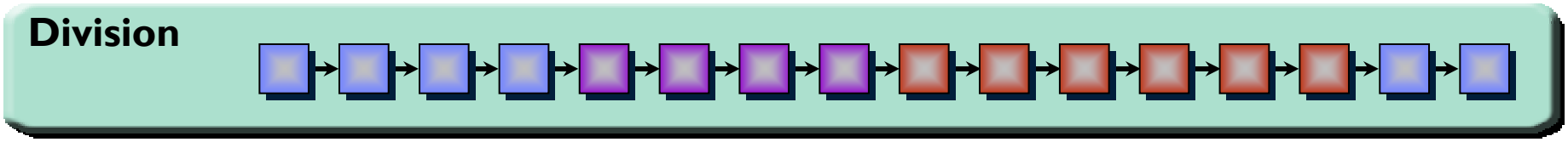
# Key Messages

- **The demands being placed on flexible business models require Flexible IT infrastructures. Services Oriented Architecture (SOA) will enable this transformation.**
- **IBM technologies are already successfully supporting businesses with Services Oriented Architecture.**
- **Componentization is a key enabler for On Demand.**

(C) Copyright IBM Corporation 2004, 2005

# Where We Are Heading

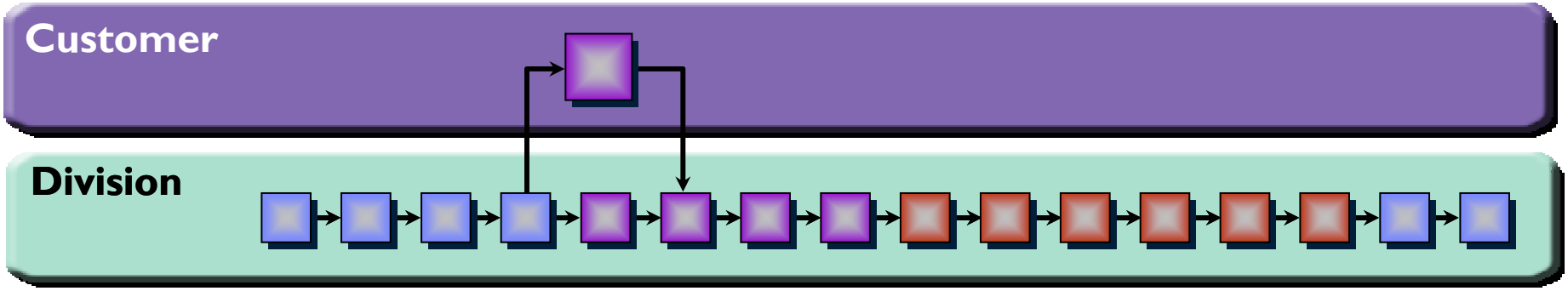
*Case Study: Procure to Pay Process*




(C) Copyright IBM Corporation 2004, 2005

# Where We Are Heading

## *Case Study: Procure to Pay Process*

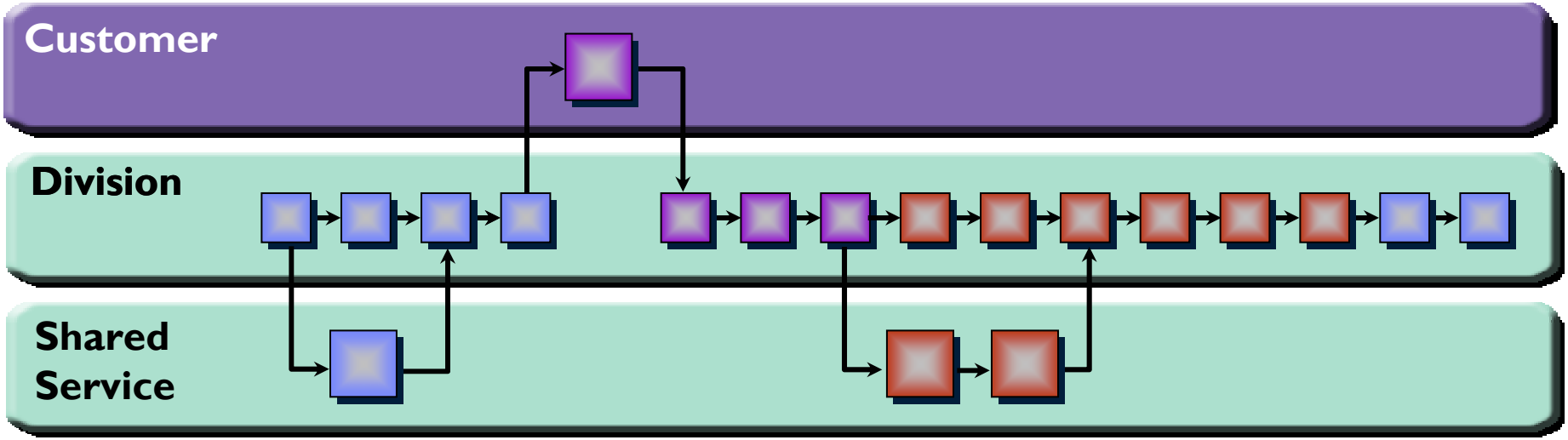



*Change: Customer Order Entry*

(C) Copyright IBM Corporation 2004, 2005

# Where We Are Heading

## *Case Study: Procure to Pay Process*

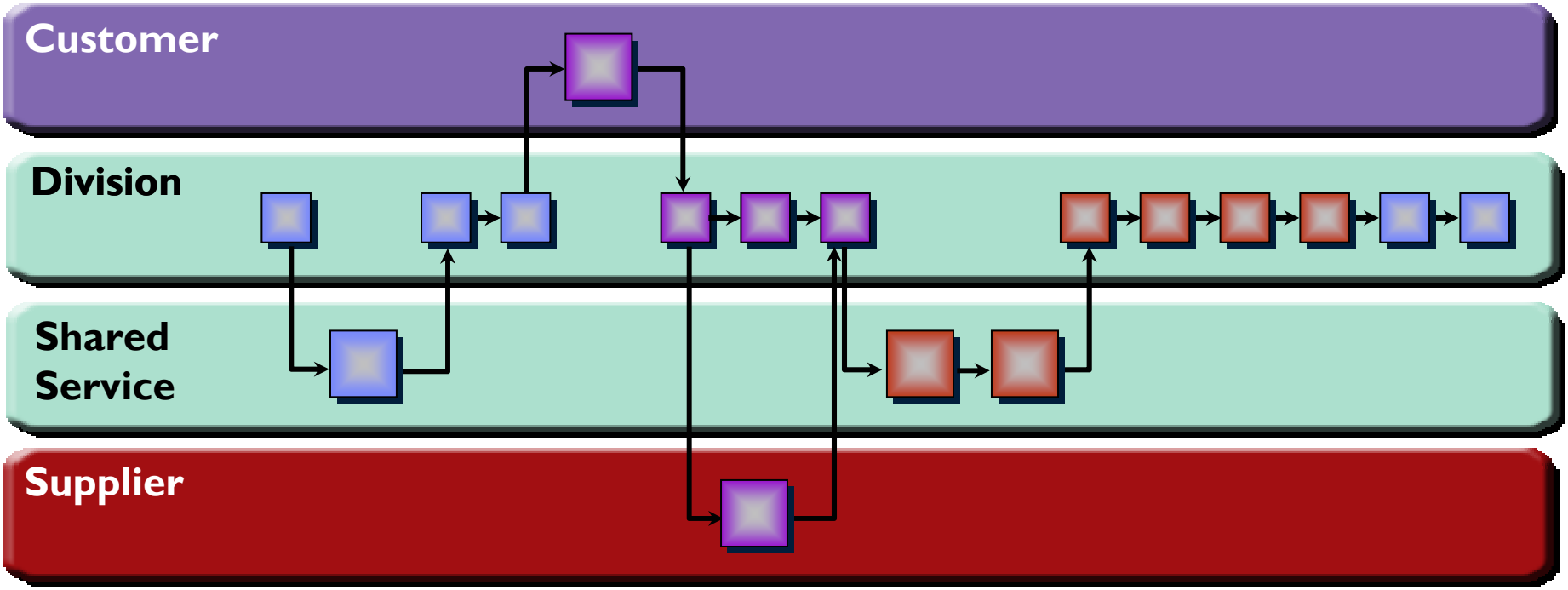



*Change: Shared Service – Marketing, Billing, Receivables*

(C) Copyright IBM Corporation 2004, 2005

# Where We Are Heading

## *Case Study: Procure to Pay Process*

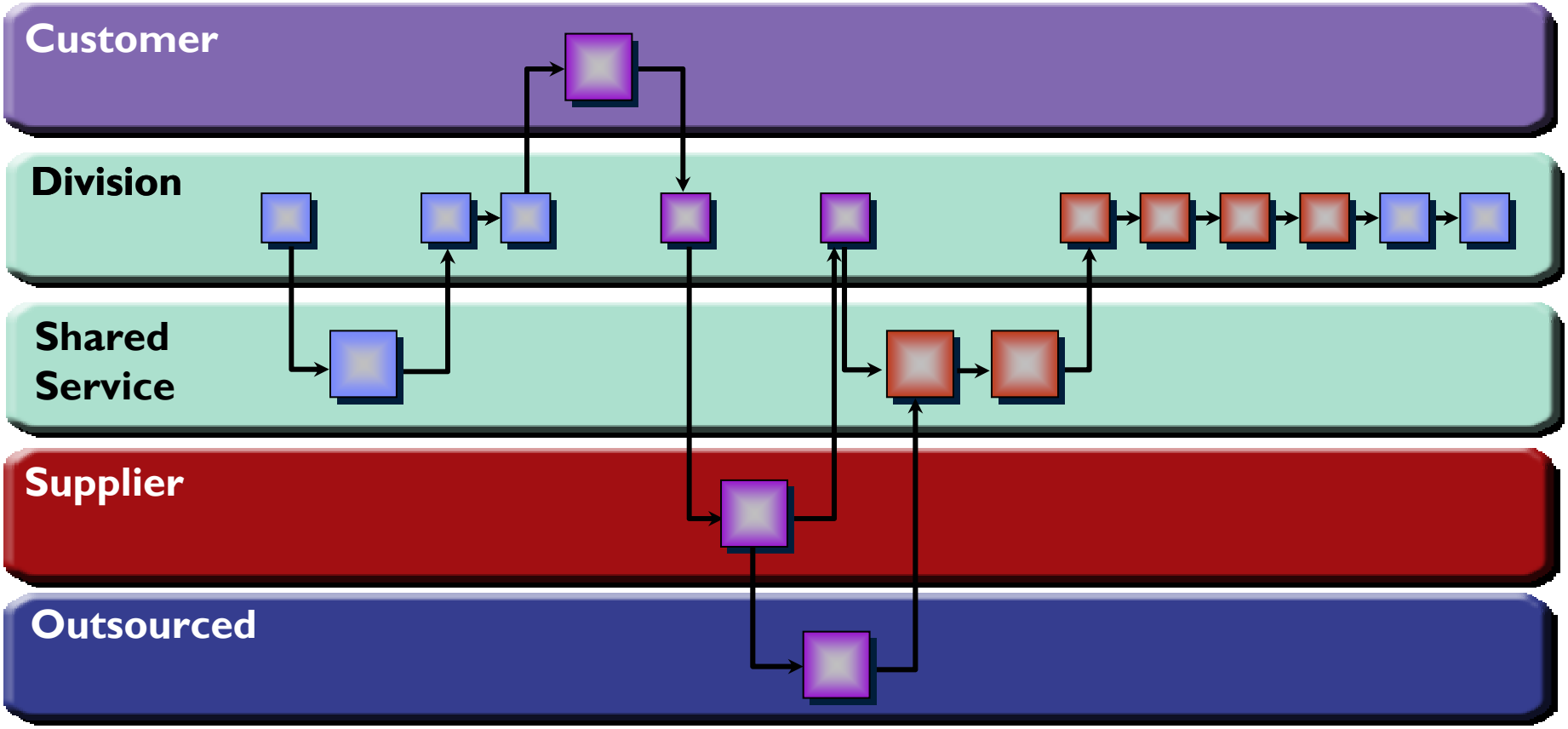



*Change: Supplier Handles Inventory (VMI)*

(C) Copyright IBM Corporation 2004, 2005

# Where We Are Heading

## *Case Study: Procure to Pay Process*

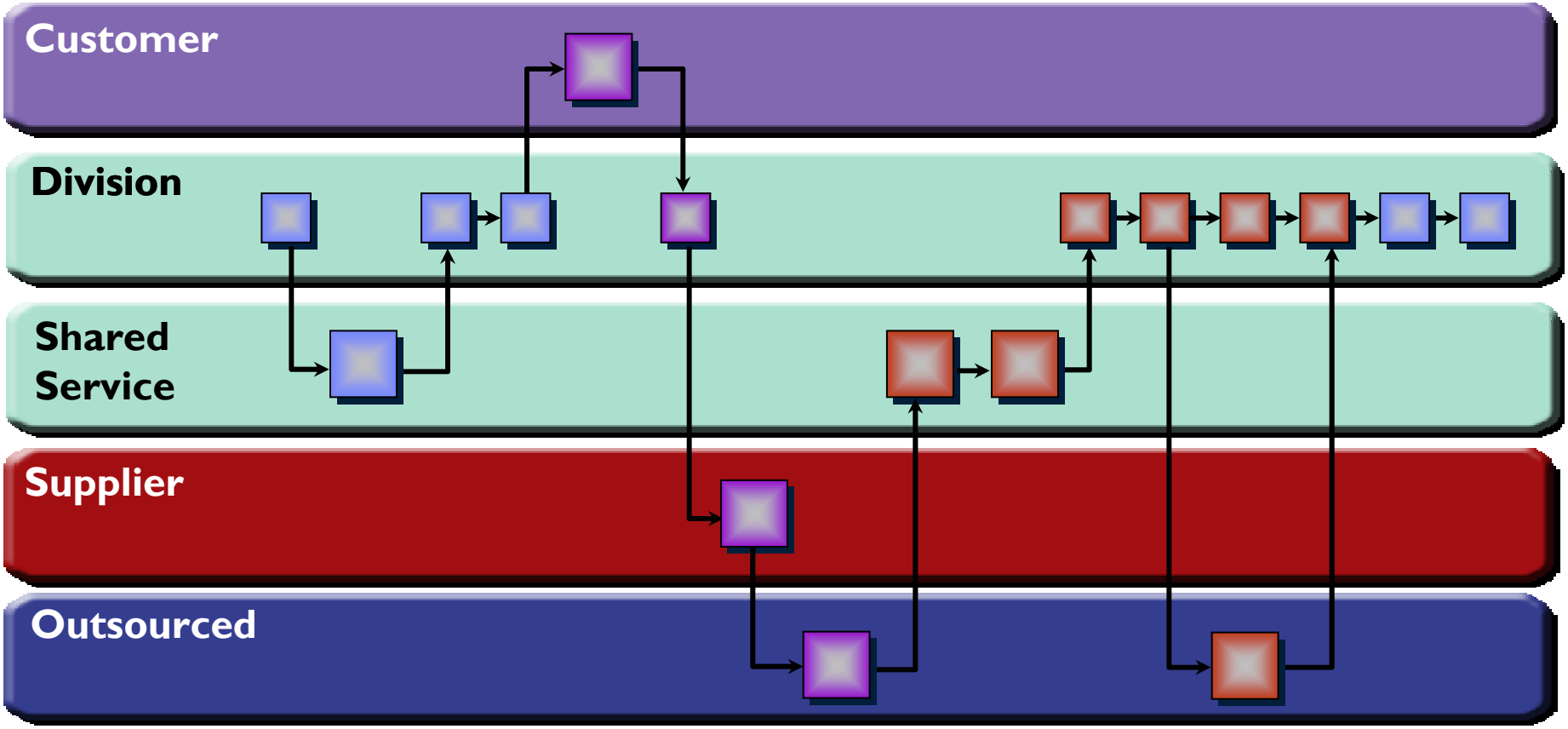



*Change: Shipping by FedEx, DHL or UPS*

(C) Copyright IBM Corporation 2004, 2005

# Where We Are Heading

## *Case Study: Procure to Pay Process*

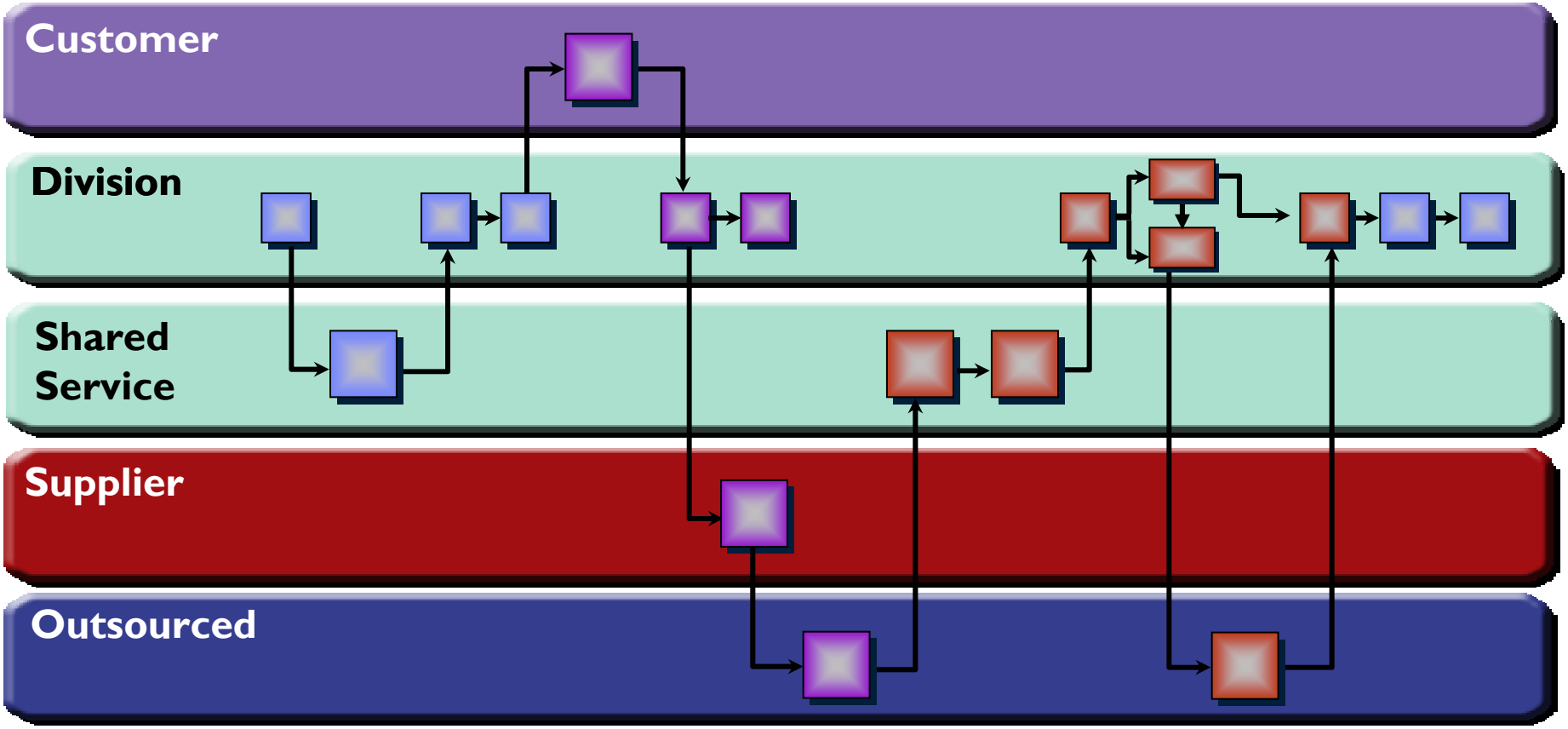



*Change: Collections Outsourced*

(C) Copyright IBM Corporation 2004, 2005

# Where We Are Heading

## *Case Study: Procure to Pay Process*




*Change: Process Optimization*

(C) Copyright IBM Corporation 2004, 2005

# Introducing Controlled Transformation

- One way to start is by deconstructing your business model—breaking it down into discrete business processes and functions
- These processes and functions are what we call service components
- Each service component serves a unique purpose and interacts with other service components in the business model, using agreed-upon cost structures and service levels

(C) Copyright IBM Corporation 2004, 2005

# Deconstruct & Conquer: The Component Business Model

**1** 2 3 4 First, break down your business into its components

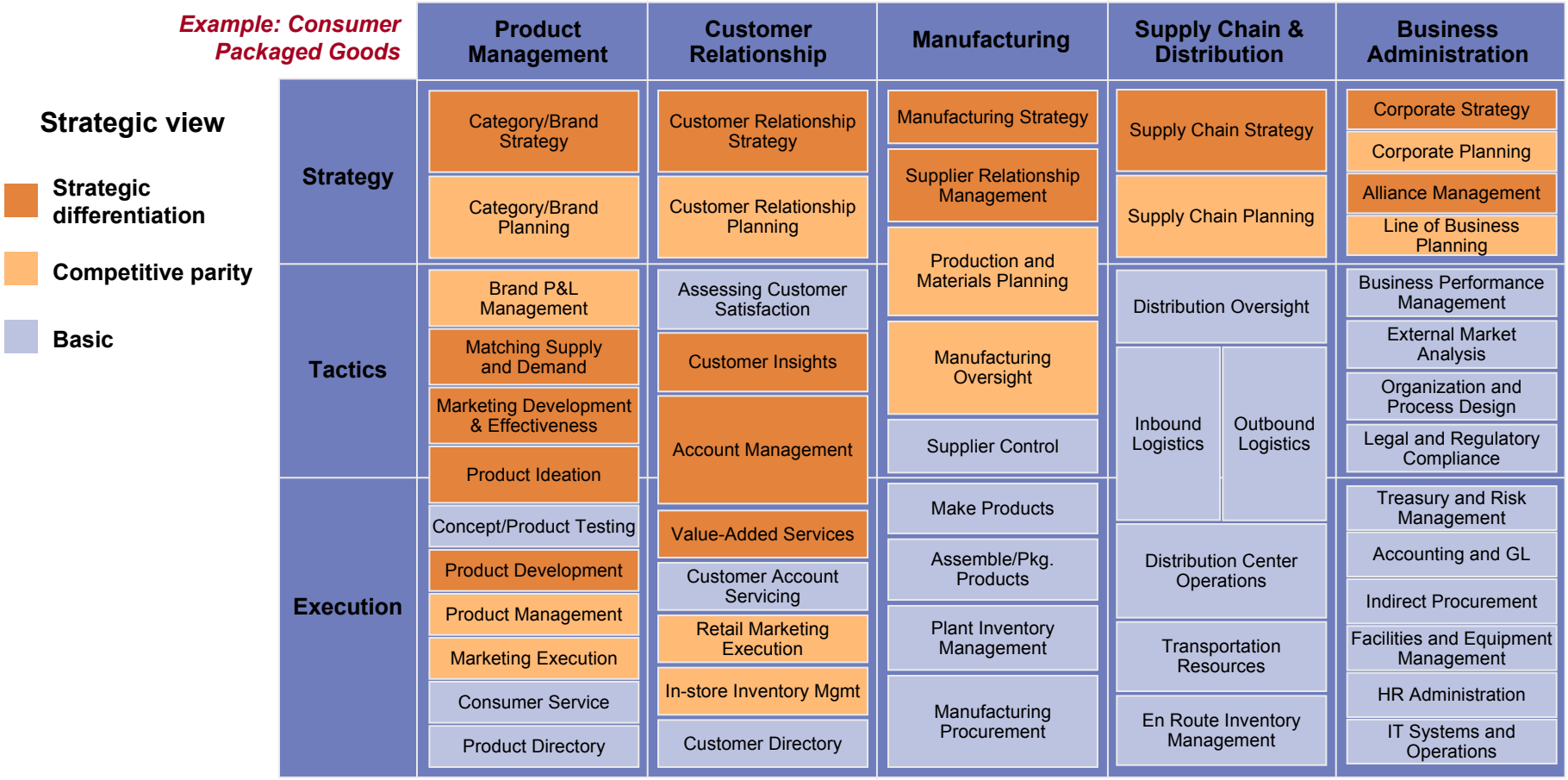
*Example: Consumer Packaged Goods*

| | Product Management | Customer Relationship | Manufacturing | Supply Chain & Distribution | Business Administration |
|---|---|---|---|---|---|
| **Strategy** | Category/Brand Strategy<br>Category/Brand Planning | Customer Relationship Strategy<br>Customer Relationship Planning | Manufacturing Strategy<br>Supplier Relationship Management | Supply Chain Strategy<br>Supply Chain Planning | Corporate Strategy<br>Corporate Planning<br>Alliance Management<br>Line of Business Planning |
| **Tactics** | Brand P&L Management<br>Matching Supply and Demand<br>Marketing Development & Effectiveness<br>Product Ideation | Assessing Customer Satisfaction<br>Customer Insights<br>Account Management | Production and Materials Planning<br>Manufacturing Oversight<br>Supplier Control | Distribution Oversight<br>Inbound Logistics<br>Outbound Logistics | Business Performance Management<br>External Market Analysis<br>Organization and Process Design<br>Legal and Regulatory Compliance |
| **Execution** | Concept/Product Testing<br>Product Development<br>Product Management<br>Marketing Execution<br>Consumer Service<br>Product Directory | Value-Added Services<br>Customer Account Servicing<br>Retail Marketing Execution<br>In-store Inventory Mgmt<br>Customer Directory | Make Products<br>Assemble/Pkg. Products<br>Plant Inventory Management<br>Manufacturing Procurement | Distribution Center Operations<br>Transportation Resources<br>En Route Inventory Management | Treasury and Risk Management<br>Accounting and GL<br>Indirect Procurement<br>Facilities and Equipment Management<br>HR Administration<br>IT Systems and Operations |

(C) Copyright IBM Corporation 2004, 2005

# Deconstruct & Conquer: The Component Business Model

1 **2** 3 4 Next, decide what's differentiating and what is simply operating




(C) Copyright IBM Corporation 2004, 2005

# Deconstruct & Conquer: The Component Business Model

1 2 **3** 4 Then, analyze costs

*Example: Consumer Packaged Goods*

**Financial view**

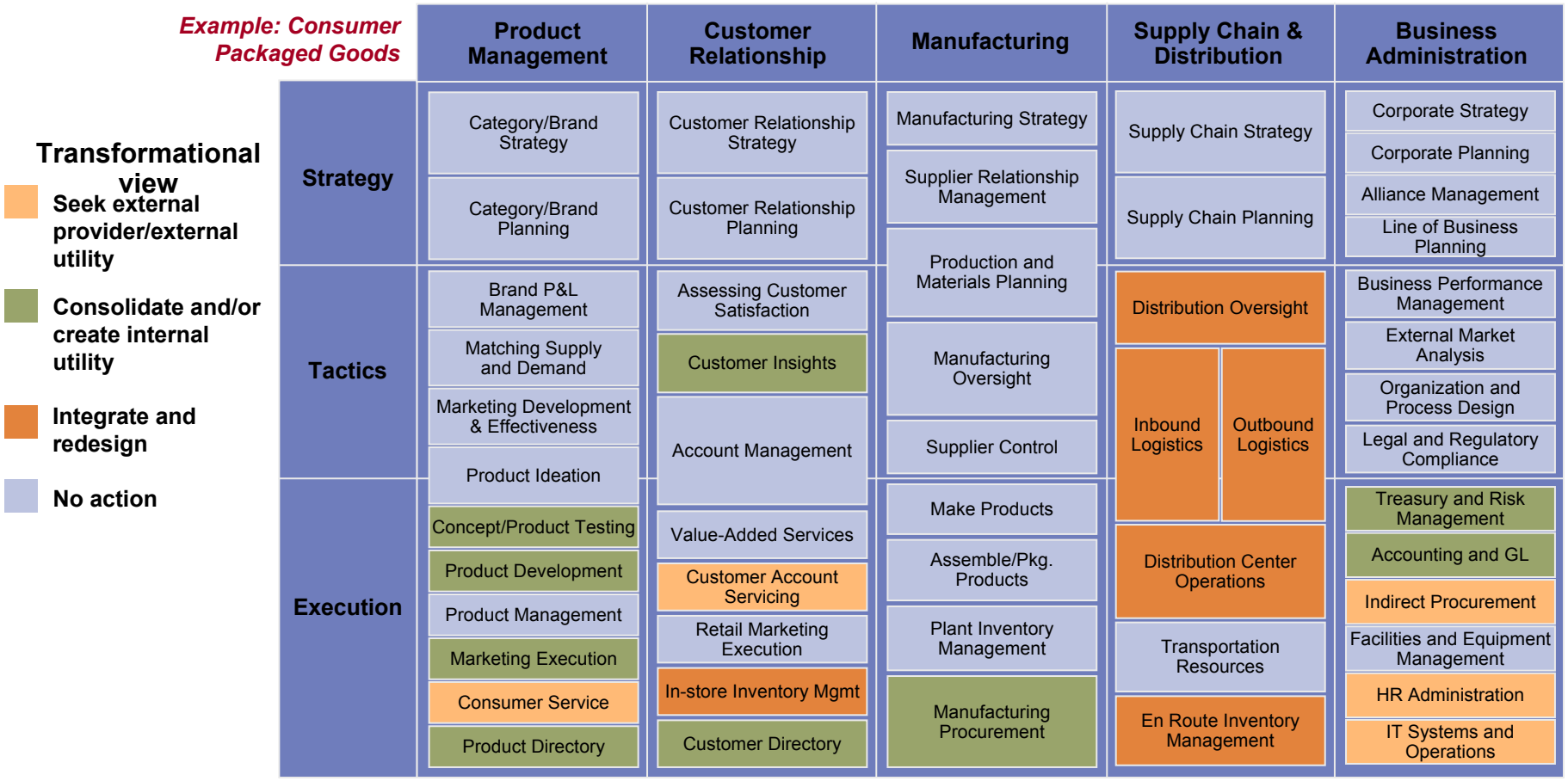
- **High capital area**
- **High cost area**
- **High cost & capital area**

| | Product Management | Customer Relationship | Manufacturing | Supply Chain & Distribution | Business Administration |
|---|---|---|---|---|---|
| **Strategy** | Category/Brand Strategy<br>Category/Brand Planning | Customer Relationship Strategy<br>Customer Relationship Planning | Manufacturing Strategy<br>Supplier Relationship Management<br>Production and Materials Planning | Supply Chain Strategy<br>Supply Chain Planning | Corporate Strategy<br>Corporate Planning<br>Alliance Management<br>Line of Business Planning |
| **Tactics** | Brand P&L Management<br>Matching Supply and Demand<br>Marketing Development & Effectiveness<br>Product Ideation | Assessing Customer Satisfaction<br>Customer Insights<br>Account Management | Manufacturing Oversight<br>Supplier Control | Distribution Oversight<br>Inbound Logistics<br>Outbound Logistics | Business Performance Management<br>External Market Analysis<br>Organization and Process Design<br>Legal and Regulatory Compliance |
| **Execution** | Concept/Product Testing<br>Product Development<br>Product Management<br>Marketing Execution<br>Consumer Service<br>Product Directory | Value-Added Services<br>Customer Account Servicing<br>Retail Marketing Execution<br>In-store Inventory Mgmt<br>Customer Directory | Make Products<br>Assemble/Pkg. Products<br>Plant Inventory Management<br>Manufacturing Procurement | Distribution Center Operations<br>Transportation Resources<br>En Route Inventory Management | Treasury and Risk Management<br>Accounting and GL<br>Indirect Procurement<br>Facilities and Equipment Management<br>HR Administration<br>IT Systems and Operations |

(C) Copyright IBM Corporation 2004, 2005

# Deconstruct & Conquer: The Component Business Model

1 2 3 **4** Finally, prioritize your transformation initiatives




(C) Copyright IBM Corporation 2004, 2005

# Why is this Scenario So Desirable?

*Business Monitoring & Optimization Leveraged by Innovation and Technology*

- A much more pragmatic approach to re-engineering efforts
  - Incremental deployments
  - Clearly tied to tangible business benefits
- Move function to the organization that can best meet your business needs – within the organization, to a business partner or completely outsource
- Allows for ongoing business tuning without widespread disruption – optimization in isolation

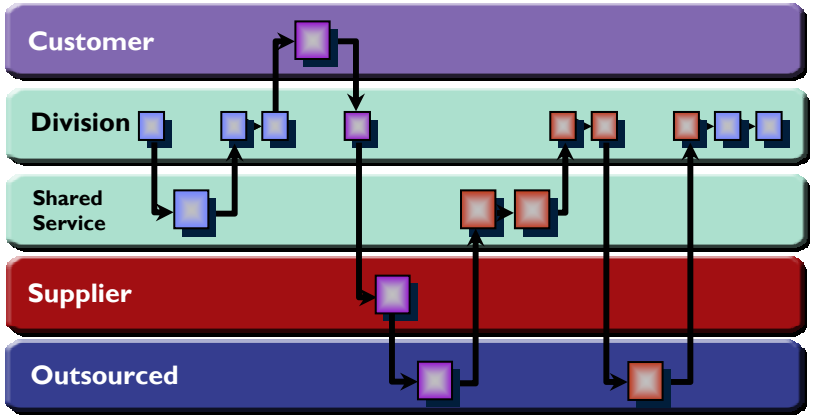
***The rate of change keeps increasing and the need for speed, flexibility, and adaptability to change is becoming that much more important***

(C) Copyright IBM Corporation 2004, 2005

# IT Architecture is a Choke Point for Business Innovation

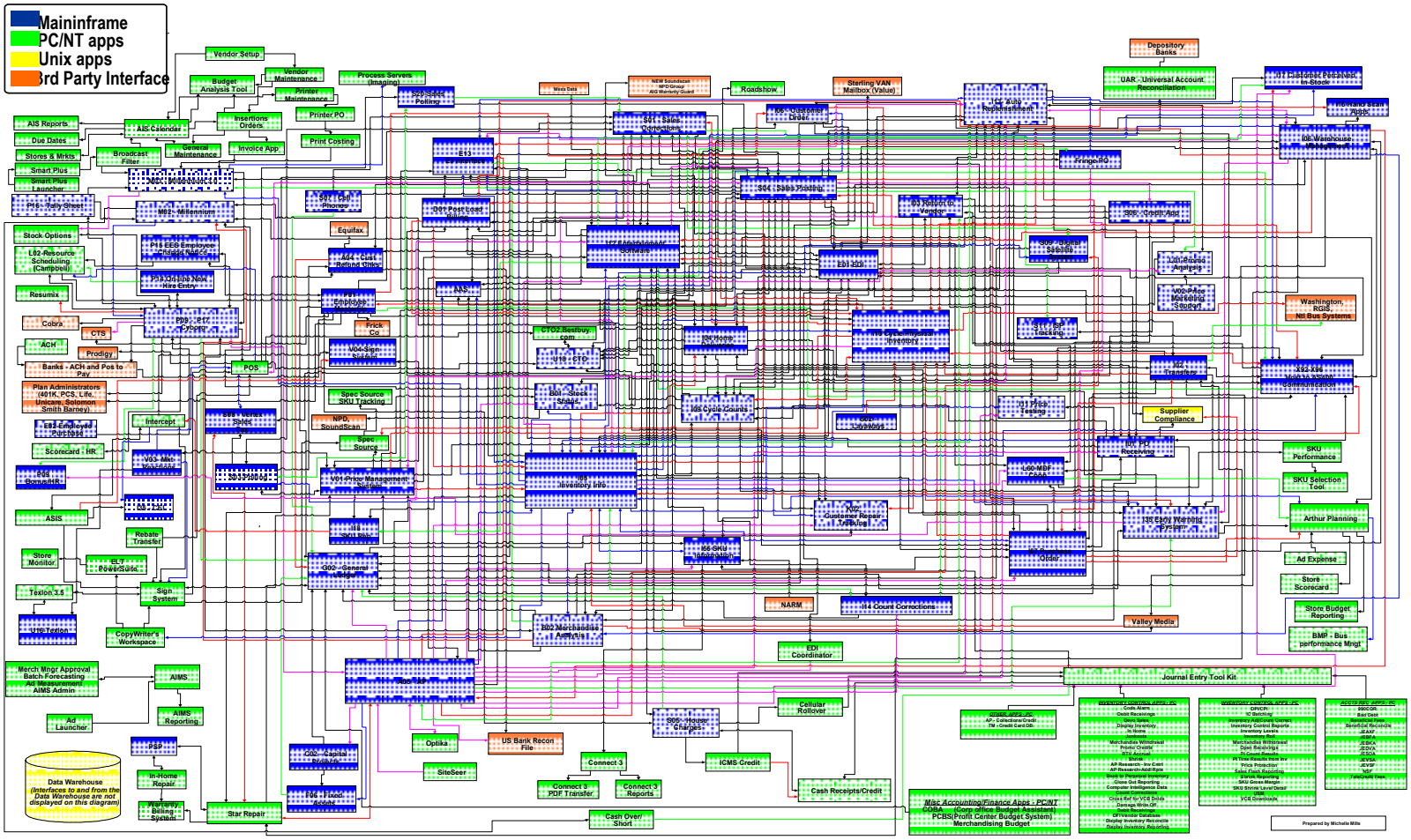
- Monolithic applications can't be reused
- Ad hoc integration creates connections that are difficult to change/maintain
- Lack of standards limits ability to deliver meaningful interoperability
- Rigidity of architecture makes small improvements impossible to cost justify

*Case Study: Procure to Pay Process*




(C) Copyright IBM Corporation 2004, 2005

# Complexity is Forcing Change

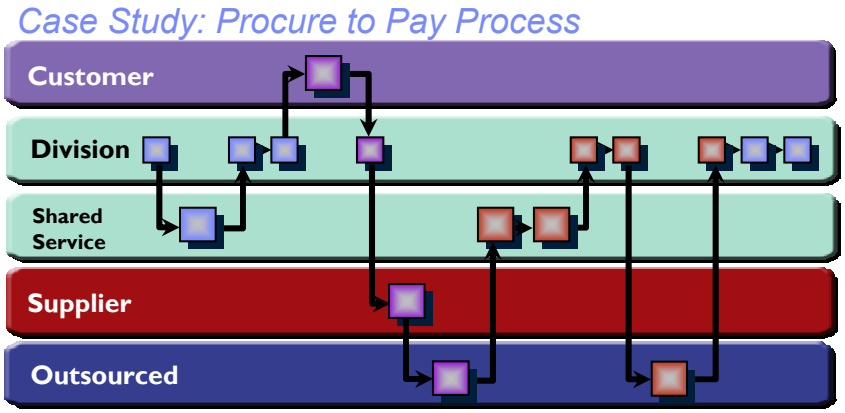



*Actual Application Architecture for Consumer Electronics Company*

(C) Copyright IBM Corporation 2004, 2005

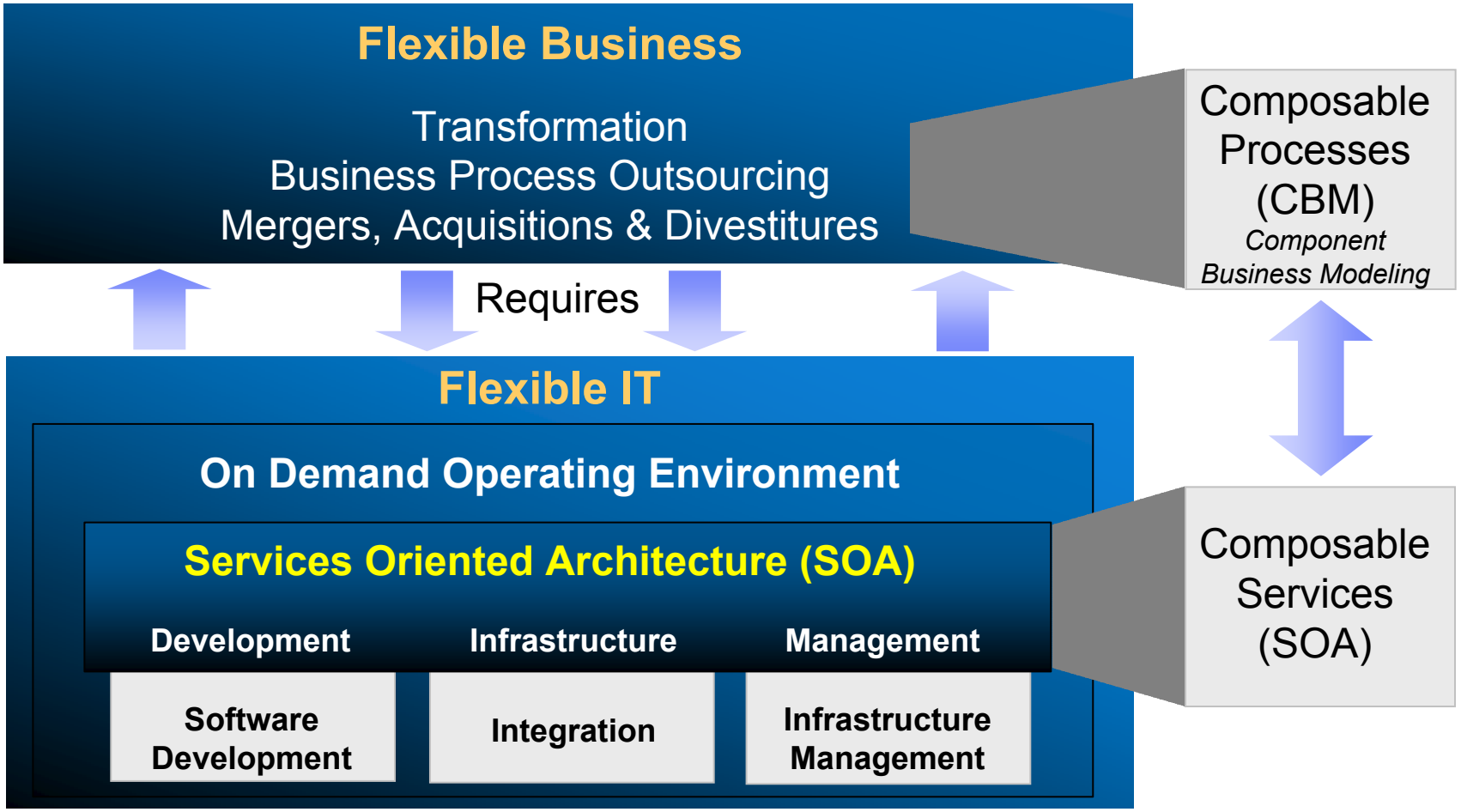
# But … Technology Applied Correctly can Pave the Way for Business Innovation

- Standards for interoperability
- Infrastructure that supports self-defined, loosely coupled interfaces emerging
- Tools incorporate existing assets through automation, virtualization, and integration



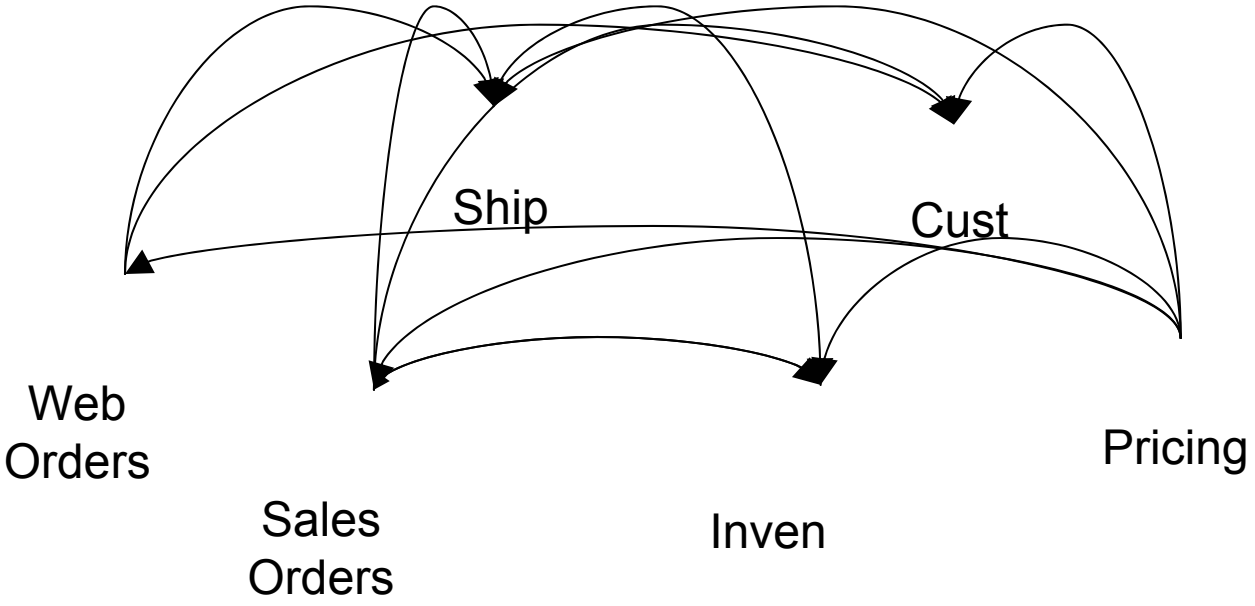

(C) Copyright IBM Corporation 2004, 2005

# Greater Flexibility Required from Business Models and the Supporting IT Architecture




(C) Copyright IBM Corporation 2004, 2005

# Component-based Architecture is not Enough

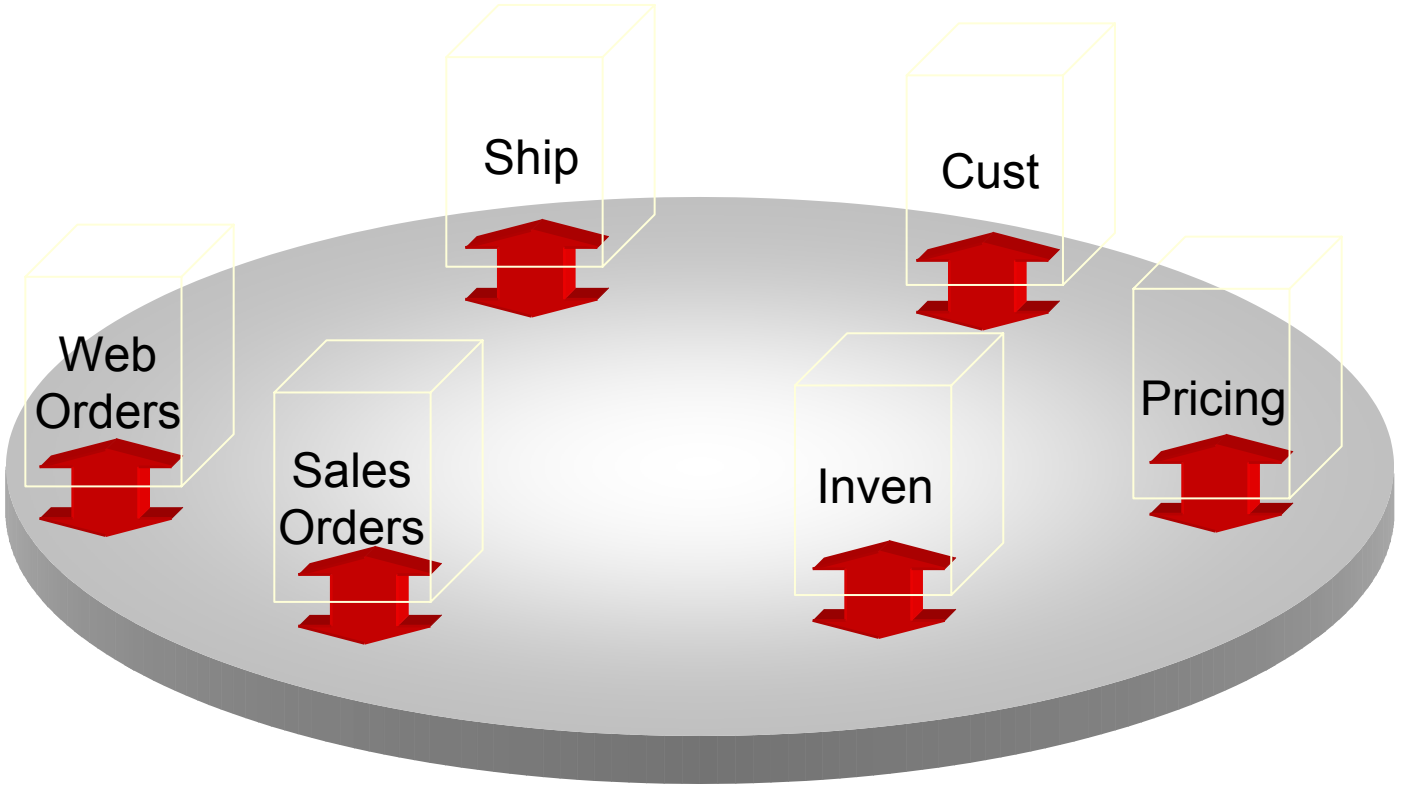



Services defined as units of business logic, but …

- Flow of control – bound into service logic
- Transformation of data formats bound into service logic
- Tight coupling between services makes them fragile

(C) Copyright IBM Corporation 2004, 2005

# Move IT Logic out of Services



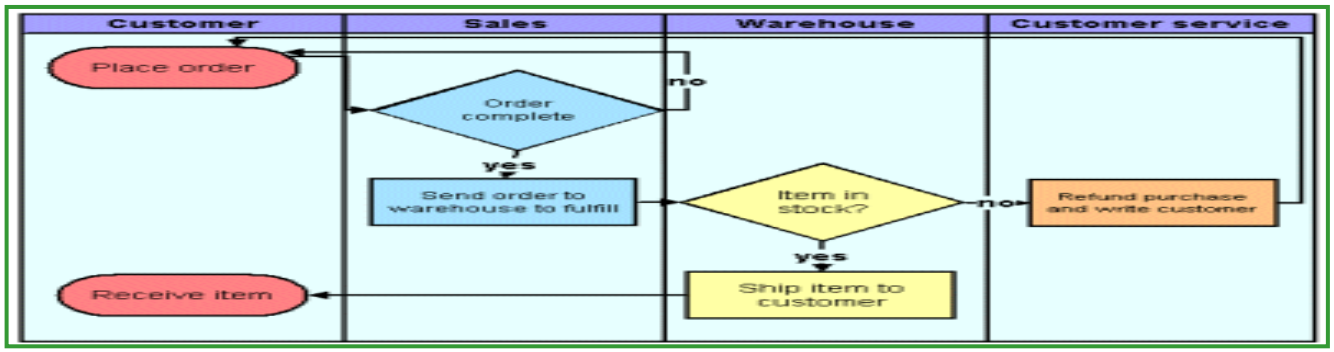

Services defined as units of business logic separated from:

- Flow of control and routing
- Data transformation and protocol transformation

(C) Copyright IBM Corporation 2004, 2005

# Aspects of the Enterprise Service Bus

***Higher-level function insulated from connection "details"***




***Customize interactions between services* (examples)**

Service Selection | Data Logging | Customized Routing | Format Translation

***Rich Comm patterns and QoS***

**Enterprise Service Bus**

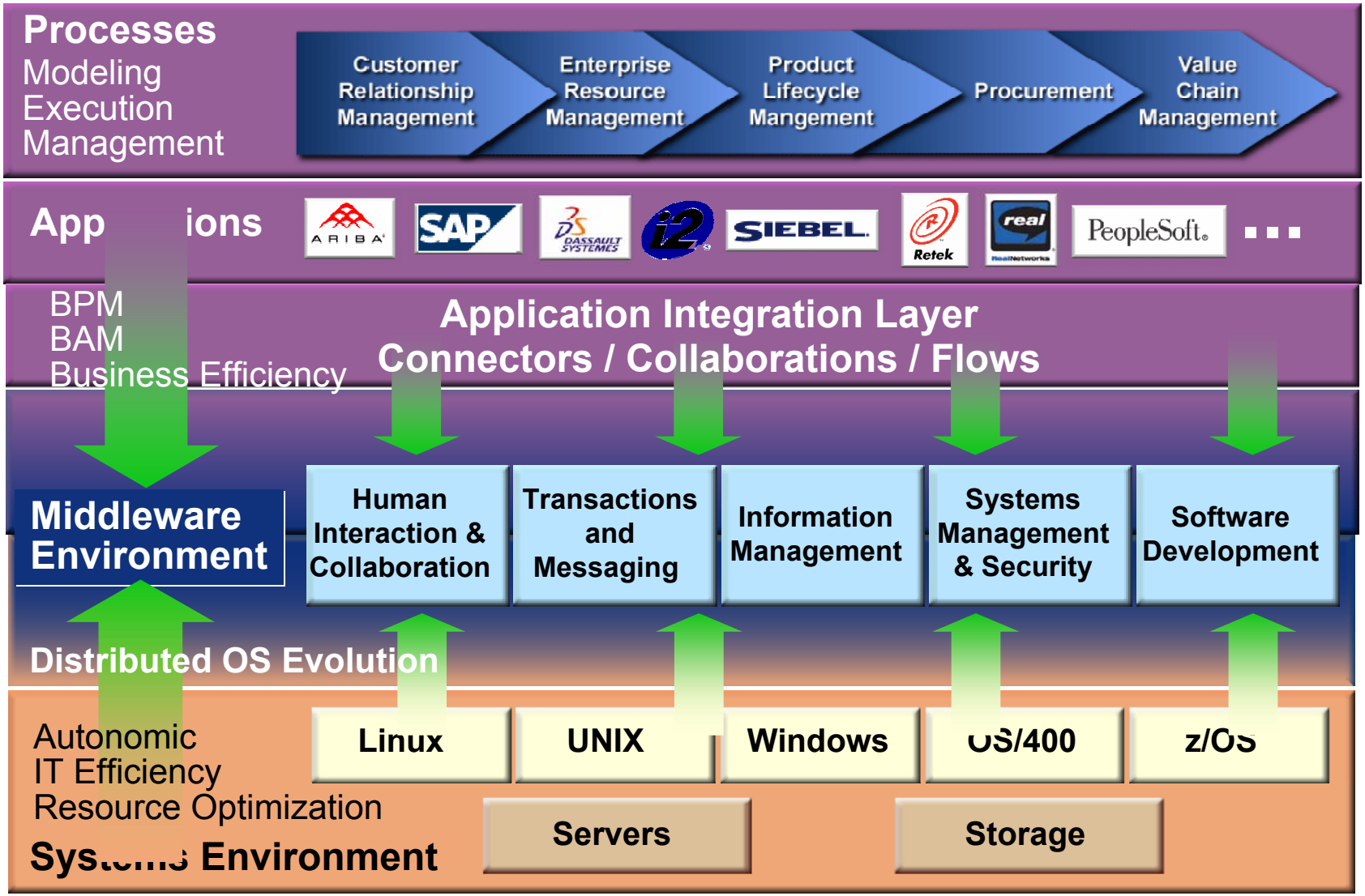
**Queues Pub/Sub Req/Rep Assured Secure Available**

***Connect anything* (examples)**

MQ SOAP/HTTP JMS WBI Adaptors CEI .NET

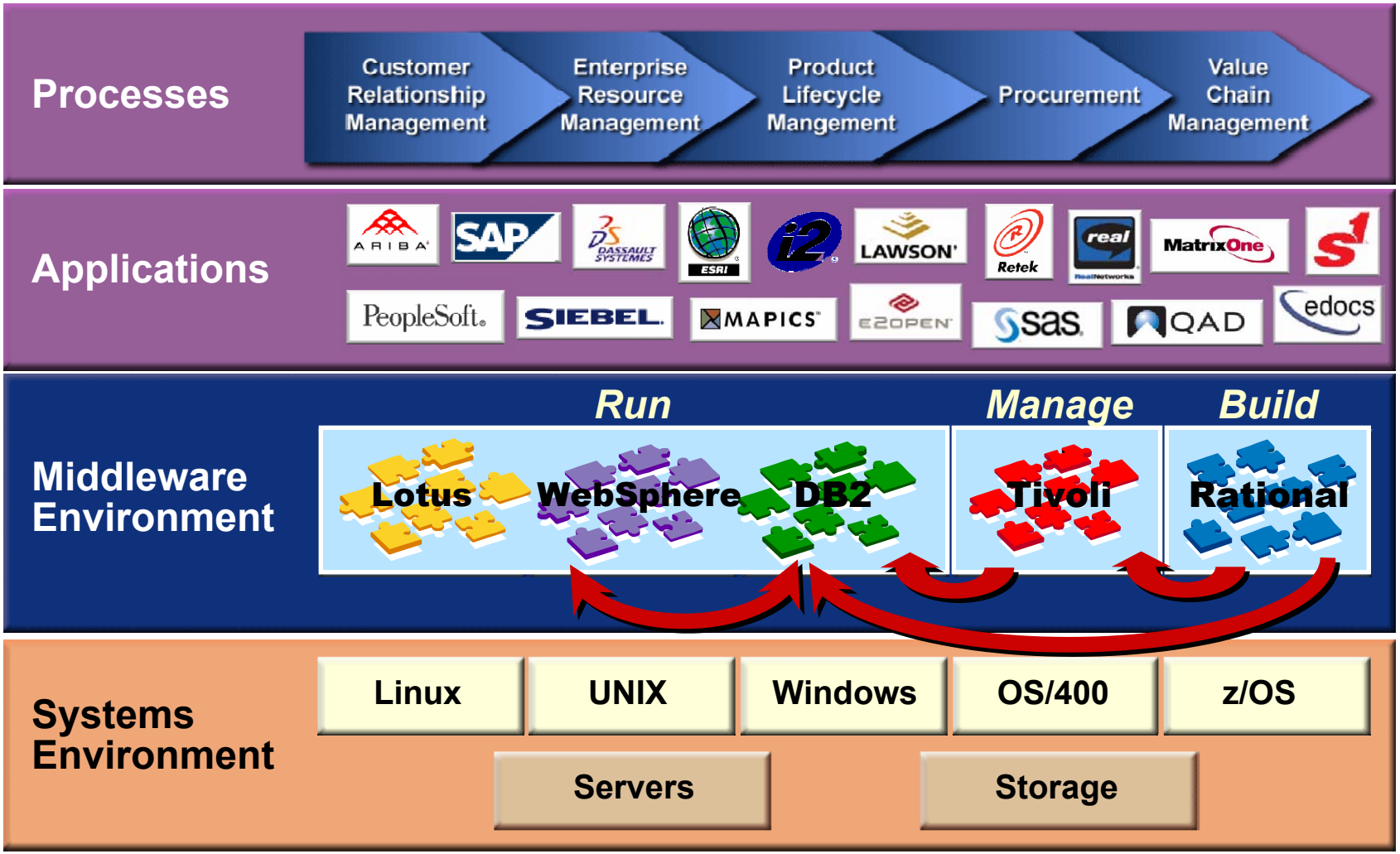
(C) Copyright IBM Corporation 2004, 2005

# Evolution Towards Middleware




(C) Copyright IBM Corporation 2004, 2005

# Middleware Platform – Componentization/Integration




(C) Copyright IBM Corporation 2004, 2005

# Product-based Development

## *Coarse Grained Reuse – WAS and UDB Technology*

- WAS technology **reused** and/or bundled by approximately **170 product offerings** (including multiple versions of some offerings) across IBM:
  - **WebSphere and BI**
    - Commerce
    - WSAD
    - WSSD
    - WSED
    - WDSC
    - WebServices Toolkit
    - Host Integration
    - Business Integration Connect
    - WBI Server Express
  - **Tivoli**
    - Policy Director
    - Privacy Manager
    - Service Level Advisor
    - Access Manager for e-business
    - Identity Manager
    - Monitoring
    - Storage Area Network Manager
    - Config Manager
  - **Lotus**
    - Portal
    - Domino
    - Workplace Messaging
    - Learning Management System
  - **Pervasive**
    - Voice Systems
    - Everyplace Access
  - **Data Management**
    - UDB Universal Dev Edition
    - DB2 Everyplace
    - Content Manager
    - Information Integrator
    - XML Registry
  - **Server Group**
    - AIX Bonus Pack
    - CICS transaction Server
  - **Systems Group**
    - Storage ESS
    - Total Storage Expert

- UDB technology **reused** and/or bundled by approximately **175 product offerings** (including multiple versions of some offerings) across IBM:
  - **WebSphere and BI**
    - WAS
    - Commerce
    - WSAD, WSADIE
    - WSSD
    - MQ Series Extended Security Edition
    - MQ Workflow
    - Business Integration Connect, Message Broker, Server
    - Everyplace Access
  - **Tivoli**
    - Policy Director
    - Configuration Manager
    - Privacy Manager
    - NetView
    - Service Level Advisor
    - Access Manager for e-business
    - Identity Manager
    - Monitoring
    - Storage Resource Mgr
    - Storage Area Network Mgr
    - Directory Server
    - Data Warehouse
    - Web Site Analyzer
  - **Lotus**
    - Portal
    - Domino
    - Workplace Messaging
    - Learning Mgmt System
    - Discovery Server
  - **Data Management**
    - DB2 Everyplace
    - Content Manager

(C) Copyright IBM Corporation 2004, 2005

# Product-based Development

## *Coarse Grained Reuse – Portal, IDS and Cloudscape*

- Portal technology has been **reused** and/or bundled by approximately **15 product offerings** across SWG, including:

- IBM Workplace for Business Control Reporting
- WebSphere Everyplace Access
- WebSphere Voice Application Access
- WebSphere Commerce Portal
- WebSphere Everyplace
- IBM Workplace
- WebSphere Everyplace Mobile Portal Extension
- Enterprise Search w/ WPI Struts Framework
- Domino
- WebSphere Everyplace Service Delivery
- iSeries HW and Portal SW bundleWebSphere Voice Application Access v5.0
- WSAD bundling WP Toolkit
- WCS bundling Likeminds
- WEA bundling WP
- WBI bundling WP

- IDS technology has been **reused** and/or bundled by approximately **17 product offerings** across SWG, including:

- AIX
- TIM
- TAM
- Tivoli Monitoring
- Visualization Engine
- WAS
- WebSphere Portal
- IBM Content Manager
- WebSphere Edge Server
- WebSphere Catalog Manager
- Lotus Learning Space/Mindspan
- IBM Workplace
- Lotus Sametime
- UDB DB2 Client

- **Cloudscape** technology has been reused and/or bundled by approximately **45+ products offerings** across SWG:

**WebSphere and BI**
- WebSphere Application Server
- WebSphere Application Server Express
- WebSphere Business Integration
- WSAD
- WSSD
- WBI Modeler
- WebSphere Business Integration Brokers - Configuration Manager

**Lotus**
- WebSphere Portal Server
- WebSphere Process Choreographer
- Lotus LearningSpace
- IBM Workplace Messaging (server side)
- IBM Workplace Client

**Tivoli**
- Tivoli Policy Manager
- Tivoli Management Portal
- Tivoli Storage Resource Manager
- Tivoli Storage Area Network Manager
- Tivoli Monitoring (ITM)
- Tivoli Monitoring for Network Performance (ITMNP)
- Tivoli Workload Scheduler
- Tivoli License Manager

**SWG Other**
- Integrated Solution Console
- Seneca (On Demand project)
- Autonomic Computing Toolbox
- Solution Install

**Data Management**
- DB2 Content Manager
- Jetstream

(C) Copyright IBM Corporation 2004, 2005

# Software Continues to Evolve

*The New Programming Model*


**Open, standards-based**

**Event Orientation** **Flexibility**

**Service Orientation**

**Incremental Integration**

(C) Copyright IBM Corporation 2004, 2005

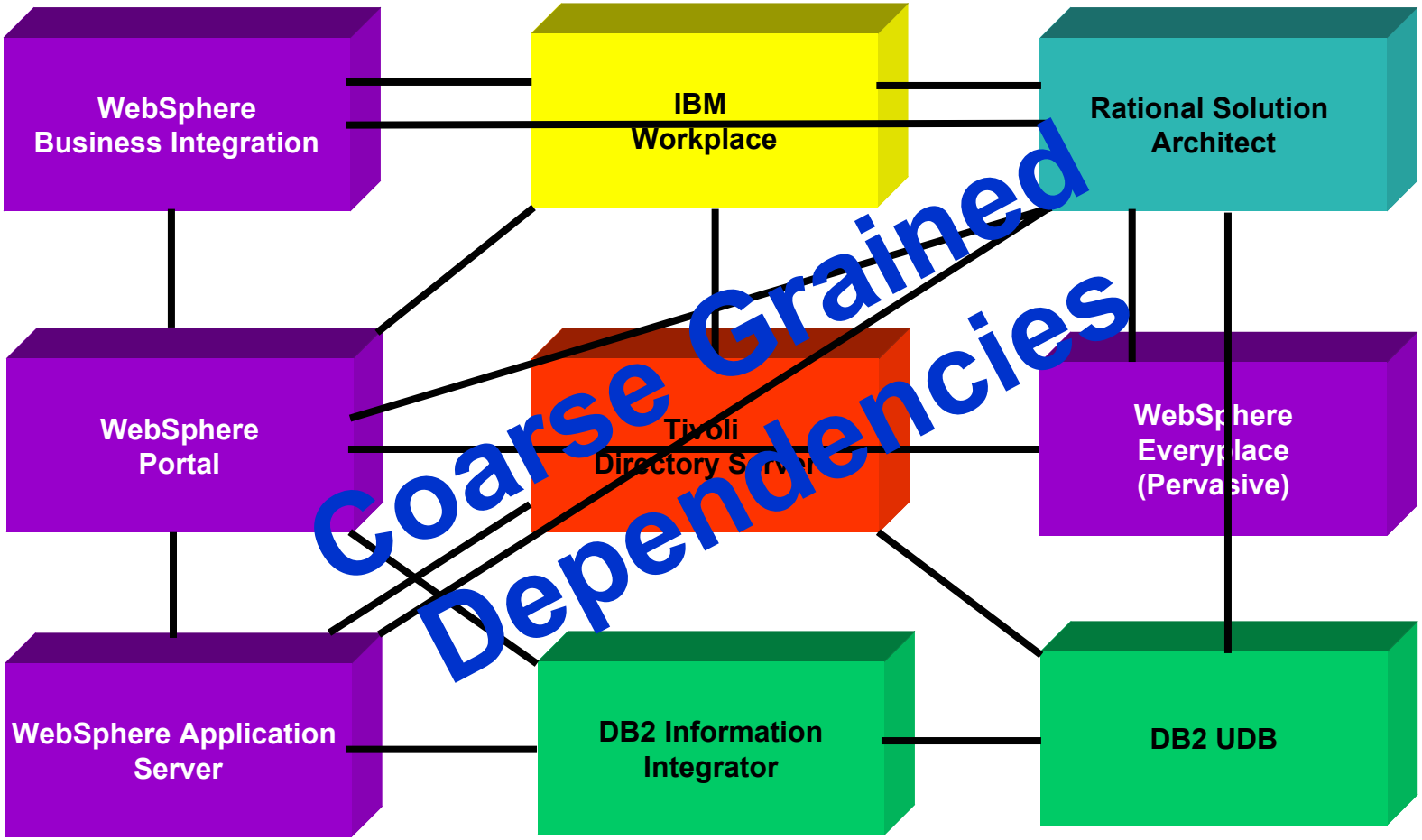
# Key Messages

- The demands being placed on flexible business models require Flexible IT infrastructures. Services Oriented Architecture (SOA) will enable this transformation.
- IBM technologies are already successfully supporting businesses with Services Oriented Architecture.
- **Componentization is a key enabler for On Demand.**

(C) Copyright IBM Corporation 2004, 2005

# Product-based Development

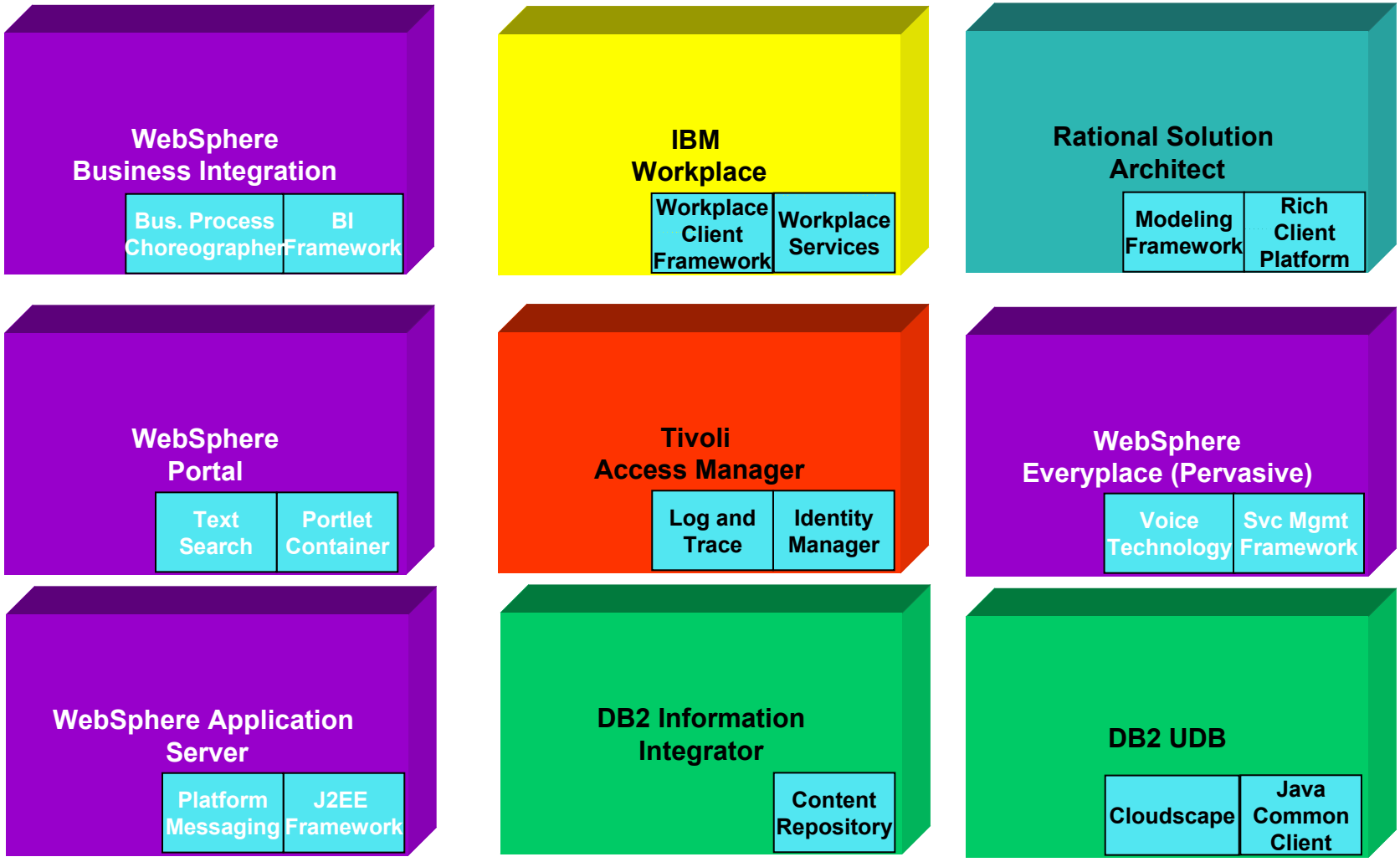
## *Product-to-Product Dependencies*




(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

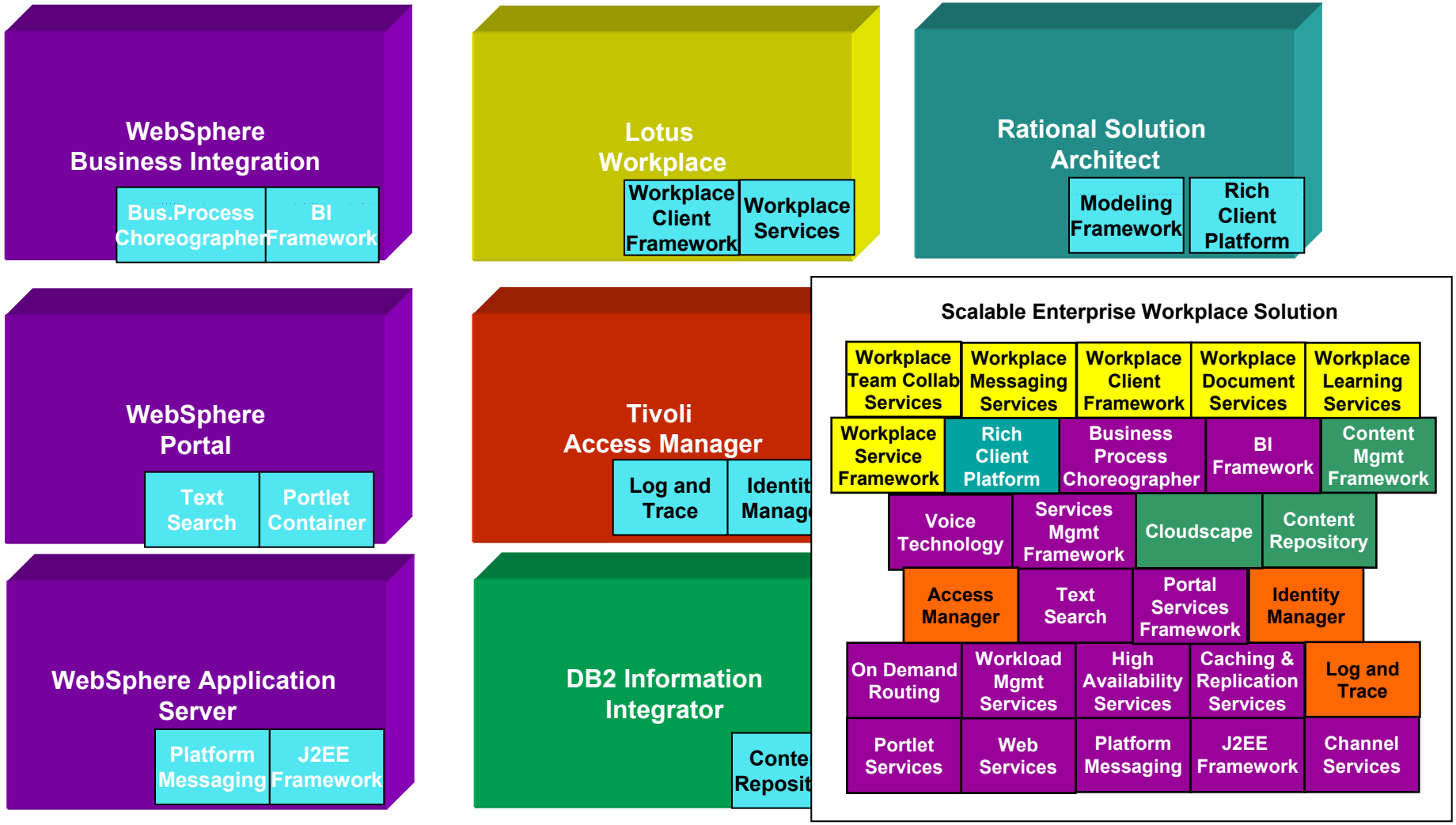
*Refactor Existing Products Into Best-of-Breed Components*




(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *Assemble Industry-Leading Solutions*



# Component-based Development
## *Benefits*

- **Introduce consistent behavior and increased simplicity through common components**
  - Autonomic components (Solution Install, Integrated Solutions Console, Common Event Infrastructure)
  - 35+ products built on Eclipse IDE Core and RCP. 35 Additional products built using Eclipse Help System, EMF or Hyades
- **Reduce Redundant Development & Redundant Processes**
  - Reuse of the best function rather than reinvention of similar (but different) function
  - Reuse of Common Criteria certifications for GSKit → reduced efforts for consumers and cost savings for consumers.
  - Eclipse is at the core of Rational/WebSphere tools, IBM Workplace and New DB2 Tool Strategy
- **Reduced maintenance & service costs**
  - Common parts are driven to higher quality sooner
  - Common architecture will enable better problem resolution – PD should get easier through a consistent approach
  - Tivoli reuse of L2 tool component from Alphaworks to improve serviceability → cost savings.
- **Accelerate Higher Quality**
  - Reuse drives higher initial quality of new marketplace offerings
  - Reuse of WAS self-certification test suite for new/modified Operating Systems → reduced test cycle from 20PM to 9PM saving $183K per platform for three-plus platforms.
- **Deliver to Market Faster**
  - Leveraging inventory of shared components will reduce time to market
  - Reuse of common tooling and components across WAS offerings → WAS XD reduced delivery cycle by 18 months.
- **Rapidly Adjust to Changing Marketplace**
  - Ability to compose solutions that are more directly targeted at specific or unique market opportunities. Allow for new offerings to be built quicker by leveraging existing inventory. Enable an ecosystem
  - Reuse of open standard components (Eclipse for LPW/Tools/PvC, OSGi-based Runtime for WAS/Eclipse/PvC, BPEL/JSR-170 in Workplace/Portal/CM) → solid and flexible industry standard foundation for growth across IBM offerings and with our partners.

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

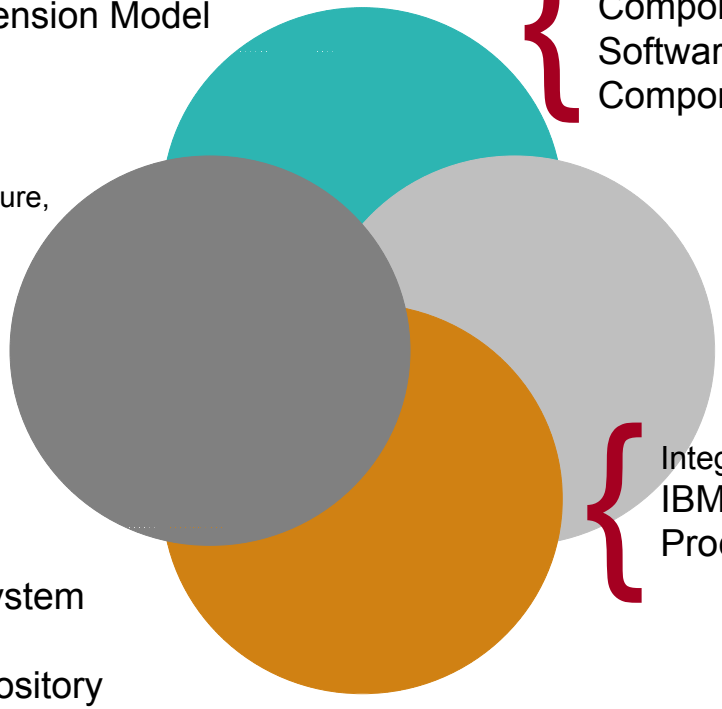
*State-of-the-art development model encompassing IBM management systems, processes, infrastructure and technology to create, share and reuse software components for a more effective and efficient development engine across IBM.*

Management Systems

- Integrated Component Mgmt Team
- Component Leadership Community
- Software Developers Leadership Community
- Component Development Teams

Technology

- Architected Platform Extension Model
- Component Models
- Reference Architectures
- SWG AB Workgroups
  - (Componentization Architecture, Common Components, Versioning and Modularity)



Processes

- Integrated Component Development Process
- IBM Rational Unified Process
- Process Workgroups

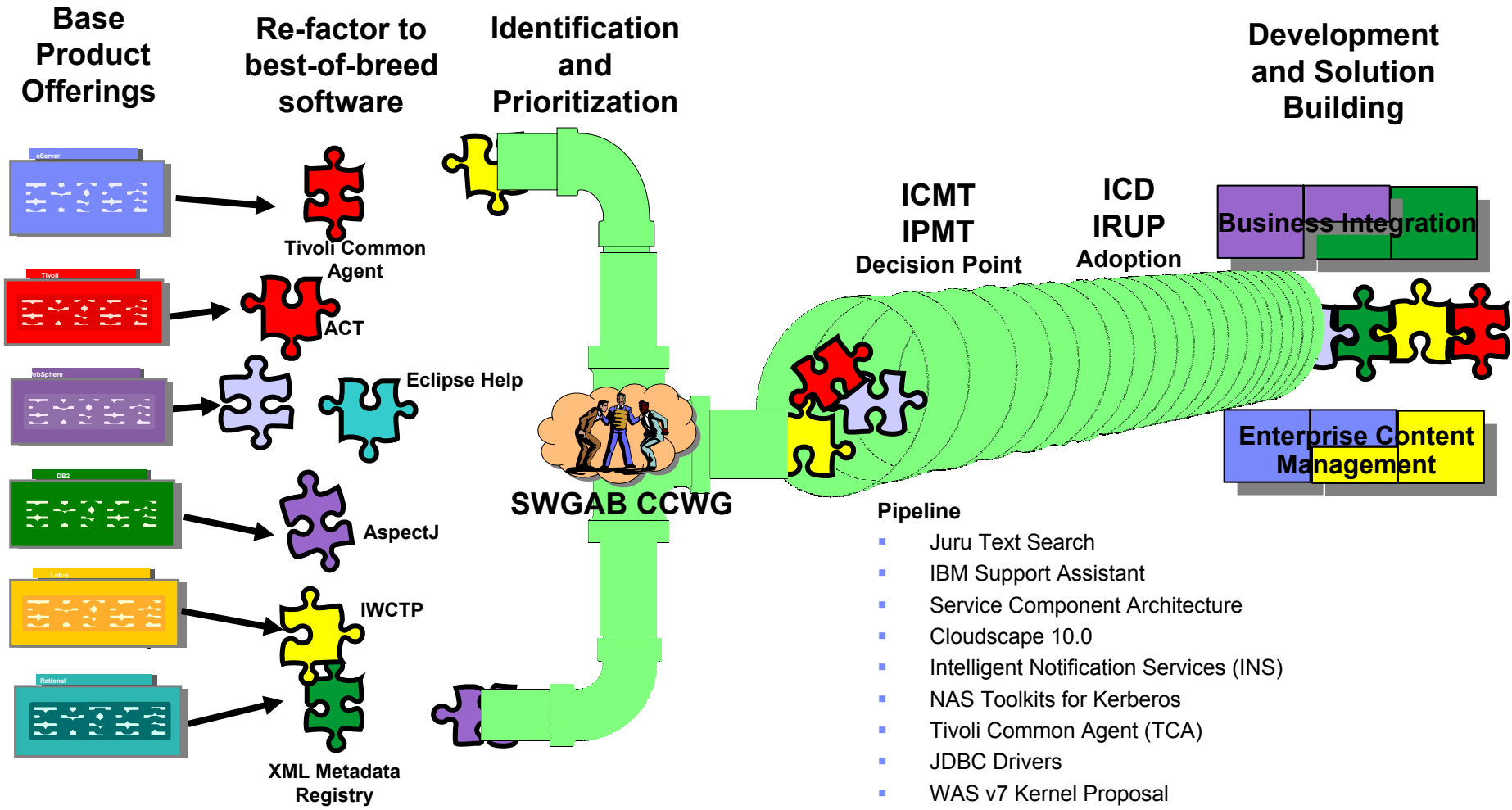
Infrastructure

- Community Source
- Asset Management System
- Clearing House
- Component Build Repository

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *Identification and Prioritization of Components*




**Pipeline**

- Juru Text Search
- IBM Support Assistant
- Service Component Architecture
- Cloudscape 10.0
- Intelligent Notification Services (INS)
- NAS Toolkits for Kerberos
- Tivoli Common Agent (TCA)
- JDBC Drivers
- WAS v7 Kernel Proposal
- Scalable Vector Graphics (SVG)

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development
## *Governing Bodies and Processes*

- Technical Leadership
  - Software Group Architecture Board (SWGAB) Common Components Workgroup (CCWG)
- Offering Development
  - Integrated Portfolio Management Team (IPMT)
  - Integrated Product Development (IPD) Process
- Component Development
  - Integrated Component Management Team (ICMT)
  - Integrated Component Development (ICD) Process
  - IBM Rational Unified Process (IRUP)

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *Commissioned Component Prioritization Criteria*

1. Enables SWG Programming Model adoption through simplification of SWG platform as a whole
2. Enables a quantum leap in our customer's experience of our offerings as well integrated, seamless, and consistent
3. Enables customers and the services team to integrate offerings into solutions and incrementally add platform capabilities
4. Enables strategic market positioning of offerings and solutions (e.g., enabling success with government accounts such as ICC in GSKit/JCE for FIPS 140-2 compliance, etc.)
5. Is critical to the success of one or more strategic customer scenarios
6. Is used pervasively across the platform and is therefore critical to the success of the platform

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *Commissioned Components Governed by the ICMT*

1. Active Correlation Technology (ACT)
2. Autonomic Computing Log & Trace (AC L&T)
3. Business Process Choreographer (BPC)
4. Common Event Infrastructure (CEI)
5. Embedded version of IBM WebSphere Application Server
6. GSKit
7. IBM Eclipse SDK (IES)
8. IBM Java SDK
9. IBM Workplace Client Technology
10. Integrated Solutions Console (ISC)
11. Java Content Repository (JCR)
12. *Java Security components for SDK*
13. OmniFind Search Component
14. Platform Messaging (Jetstream)
15. Solution Installation for Autonomic Computing (SI)
16. Tivoli Common Audit & Report Services (CARS)
17. Tivoli WebSphere Identity Mgr (WIM)

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development
## *Components Recommended for Commissioning*

1. Juru Text Search (considered a peer with OmniFind above)
2. IBM Support Assistant (ISA -- formerly eSupport Client)
3. Service Component Architecture (SCA -- formerly JService)
4. Cloudscape
5. Tivoli Common Agent (TCA)
   - OSGi SMF Runtime
   - Extension Services for WebSphere Everyplace (ESWE)
   - Tivoli Common Agent Bundles (not separable Common Component)
   - Tivoli Agent Manager
6. Intelligent Notification Services (INS)
7. JDBC Drivers
8. NAS C Toolkit for Kerberos
9. Scalable Vector Graphics (SVG)

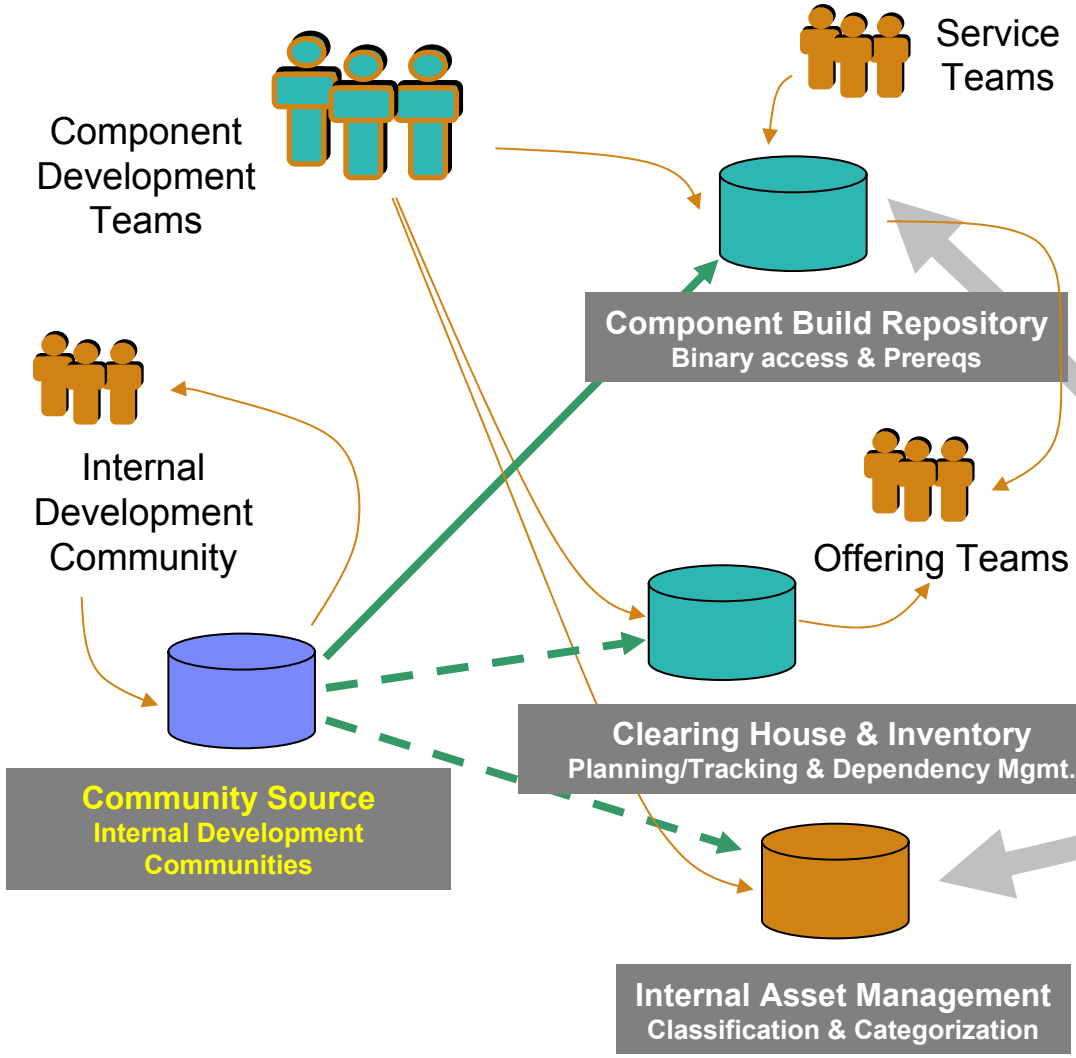
(C) Copyright IBM Corporation 2004, 2005

# Component-based Development
## *Upcoming Component Proposals*

1. Workplace Client Technology - Micro Edition (WCT-ME)
   - OSGi SMF Runtime
   - Extension Services for WebSphere Everyplace (ESWE)
2. WS-Policy4J
3. Eclipse Help System
4. Session Initiation Protocol (SIP) Container
5. Relationship Registry (formerly CMDB)
6. LPG Parser Generator
7. Launchpad and First Steps
8. Java Content Repository (JCR) UI / Search
9. LanguageWare
10. IBM HTTP Server
11. International Components for Unicode (ICU)
12. Abstract User Interface Markup Language (AUIML) Rendering Portlet
13. Common Console Interface (CCI) Library
14. AspectJ

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

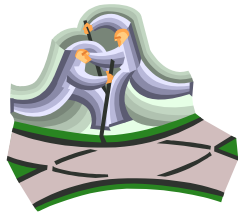



**Community Source**

- Internal development communities
- Avoids duplication of effort
- Encourages reuse over reinvention
- Leverages broader IBM community
- Collaborative "open source" development environment

**Asset Management System**

- Metadata for each prerequisite
- User profiles
- Discussion forums
- Subscription services



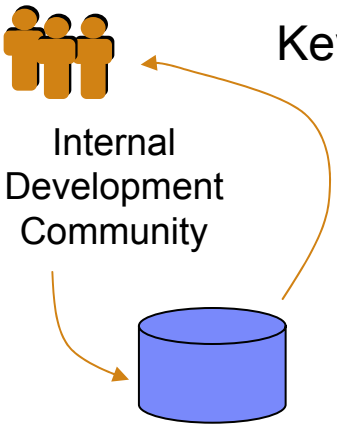
**Integrated Component Mgmt Team**

- Provides a forum for **conflict resolution**
- Addresses delivery and fulfillment planning problems
- Addresses inconsistencies and integration problems

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development
## *Community Source*

- Provides a collaborative environment accessed via a Web-based portal for shared, distributed development and testing, with over 50 projects participating to date
- Controlled “white box” approach for development of component software
- Explore shared development of components intended for reuse
- Seeded by respective development organizations




Key Features:

- Access Control
- Product builds, fixes and test drivers
- Discussion Forums
- Reference information (API specs, programming documentation, education, demo, etc.)
- Defect Reporting
- Feature Requests
- Code Storage and Version Control

## Benefits

- Encouraging reuse over reinvention
- Improving information flow between teams (availability of source materials, decisions and discussions)
- Leveraging broader IBM community skills (technical and non-technical communities)
- Improving quality through peer reviews and user feedback (defects and forums)
- Positively impacting our ability to deliver more function on shorter schedule (collaboration and contribution)
- Most valuable assets get the most attention (based on reuse)
- Facilitate development

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *Using Components to Implement SOA Services*

***The architecture defines a method for modeling the software components that implement services as part of IBM's On Demand Services Oriented Architecture (SOA).***

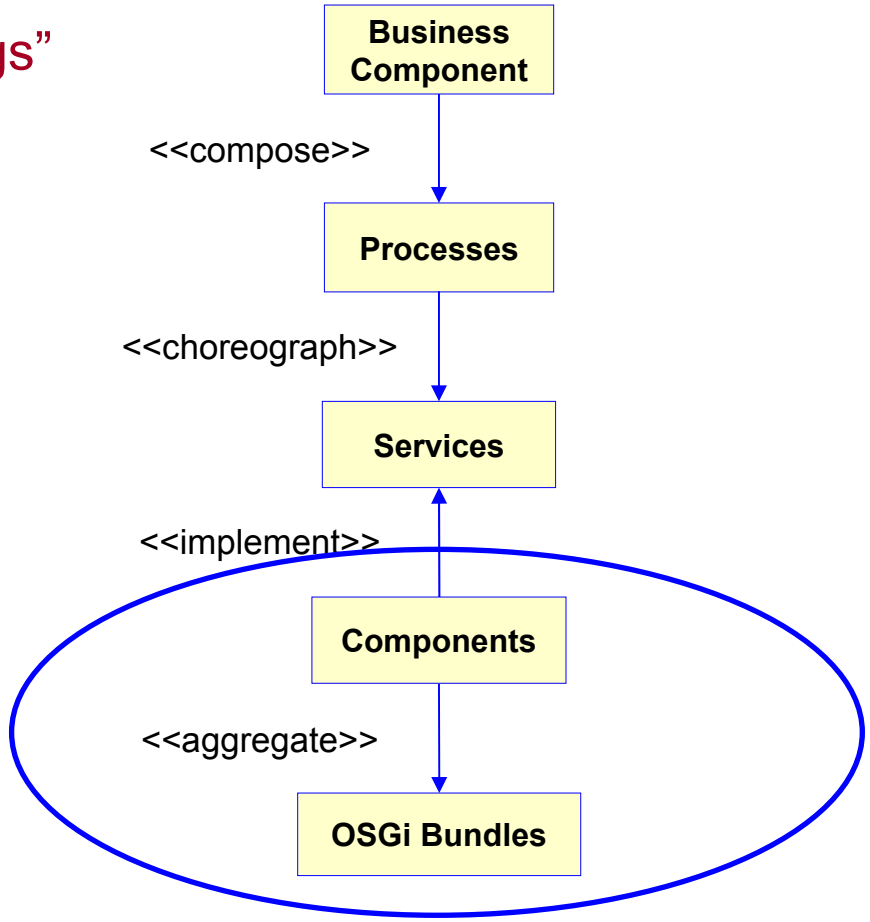
Business Value from “Things”

How “Things” are used

What “Things” do

“Things”

How “Things” are made

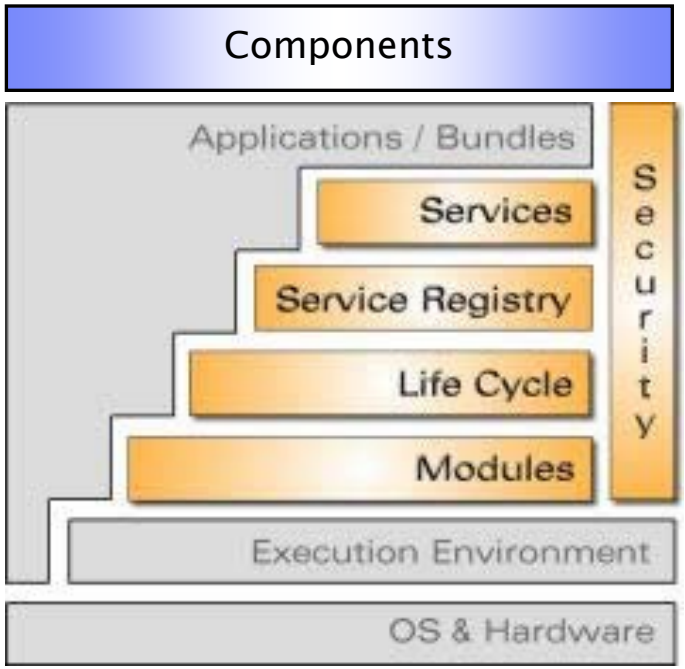



(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

*Using OSGi Technology*

- Leverage OSGi bundles and plug-in technology as a foundation for implementing components and aggregation:
  - Testing and reuse code across products
  - Reduce foot-print by only loading what you need to run
  - Dynamic provisioning of application environment
  - Leveraging (plug-into) open source activities
  - Supporting reliable Update and Versioning
  - Consistent Install and Maintenance




(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *SPX Component Modeling*

***S**oftware **P**latform e**X**tension*

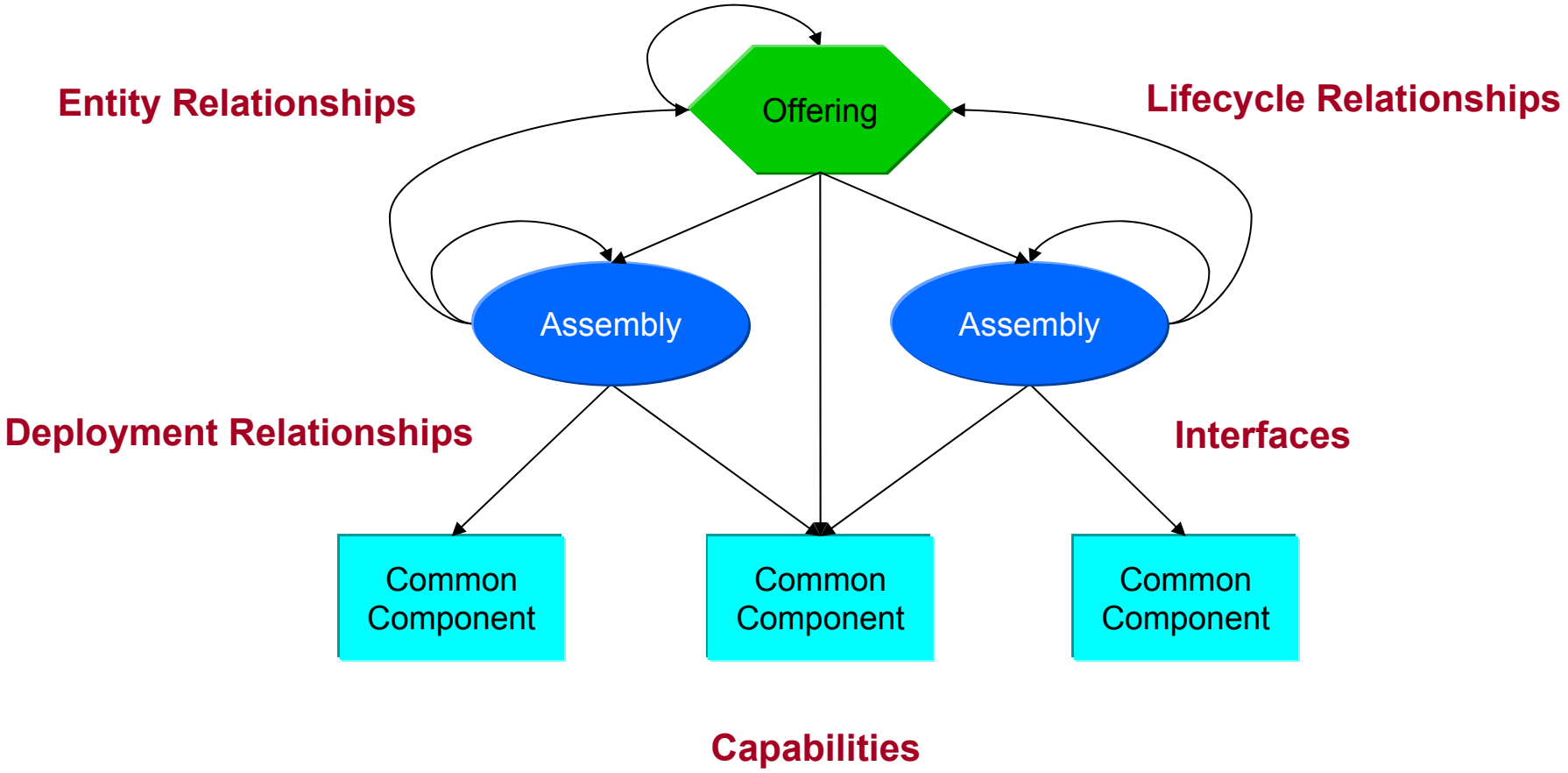
# *SPX*

The name for the architecture work that supports IBM's software componentization and software reuse strategy

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *SPX Component Modeling*

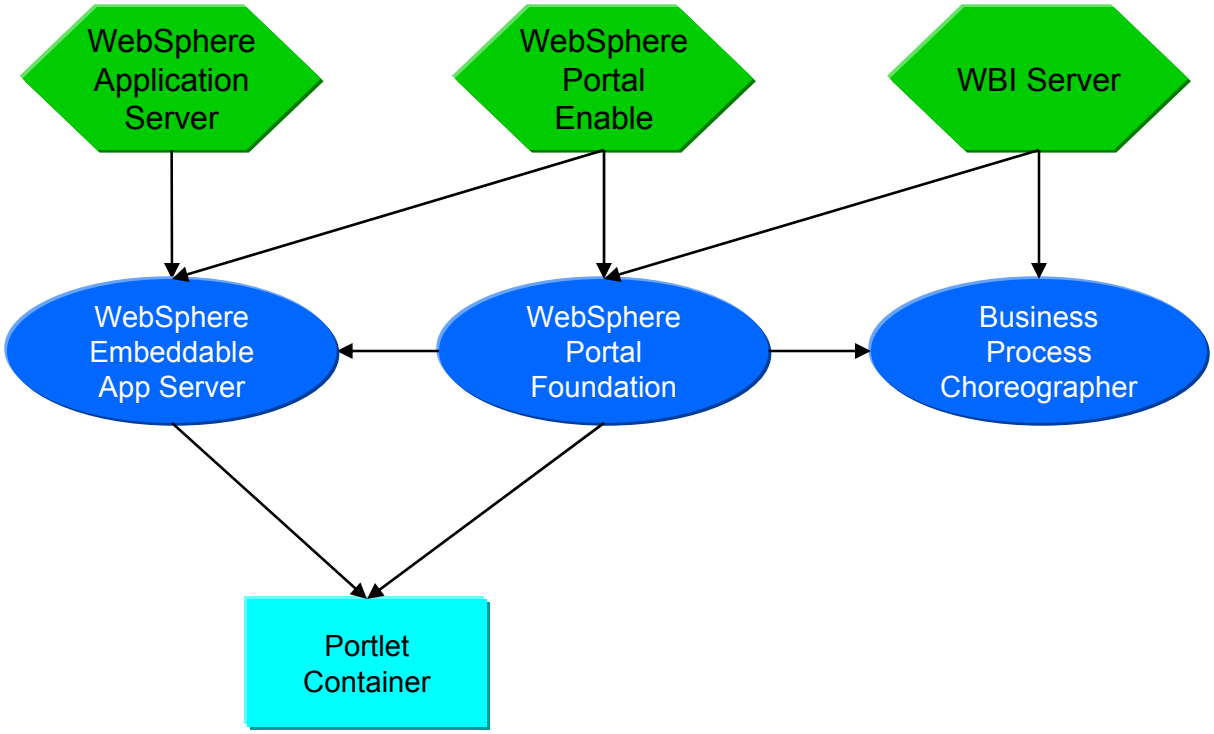



(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *SPX Component Modeling*

### Example: Subset of Component Sharing Across Offering Families



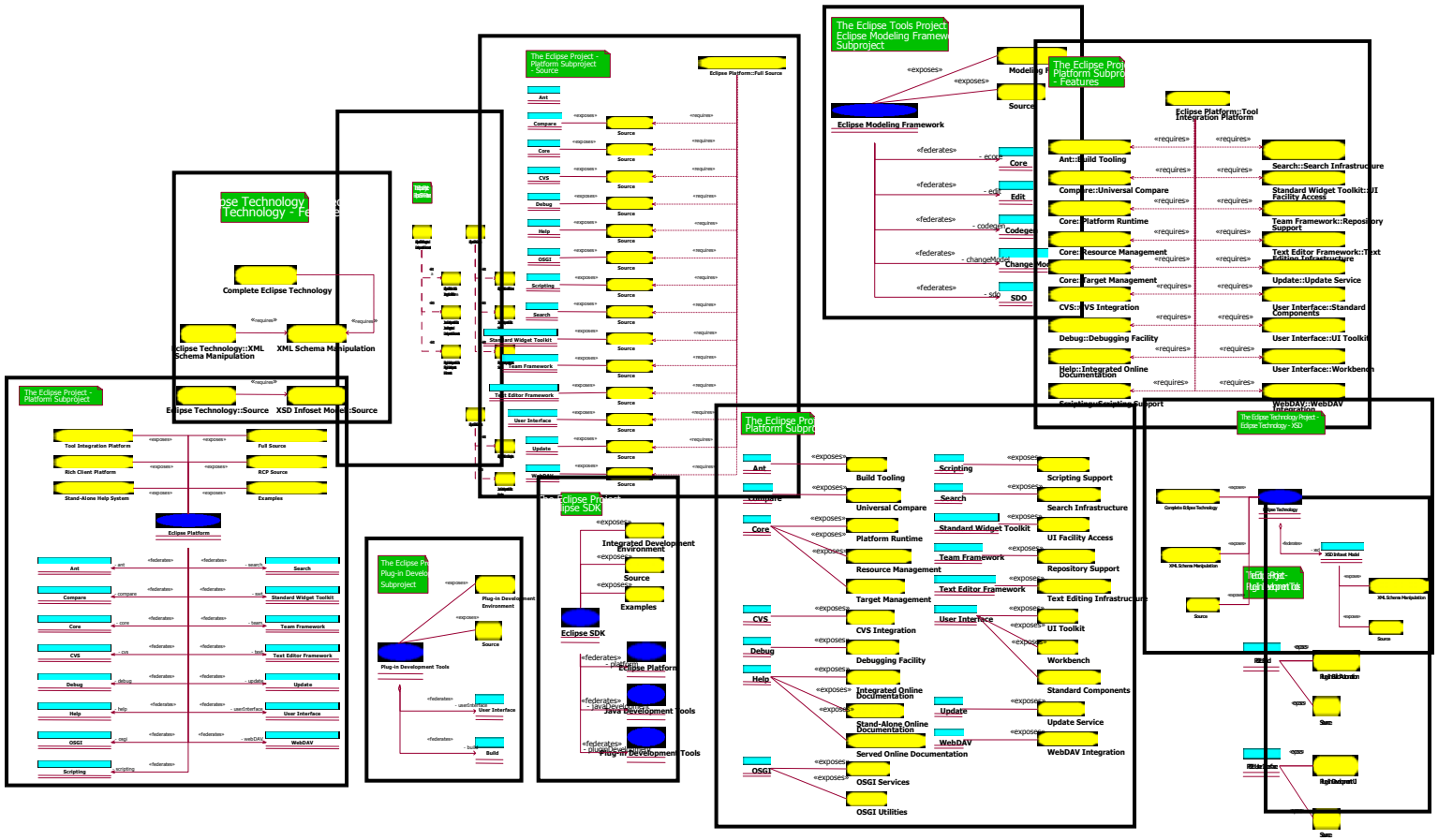

Offering | Assembly | C-Component

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *SPX Component Modeling*

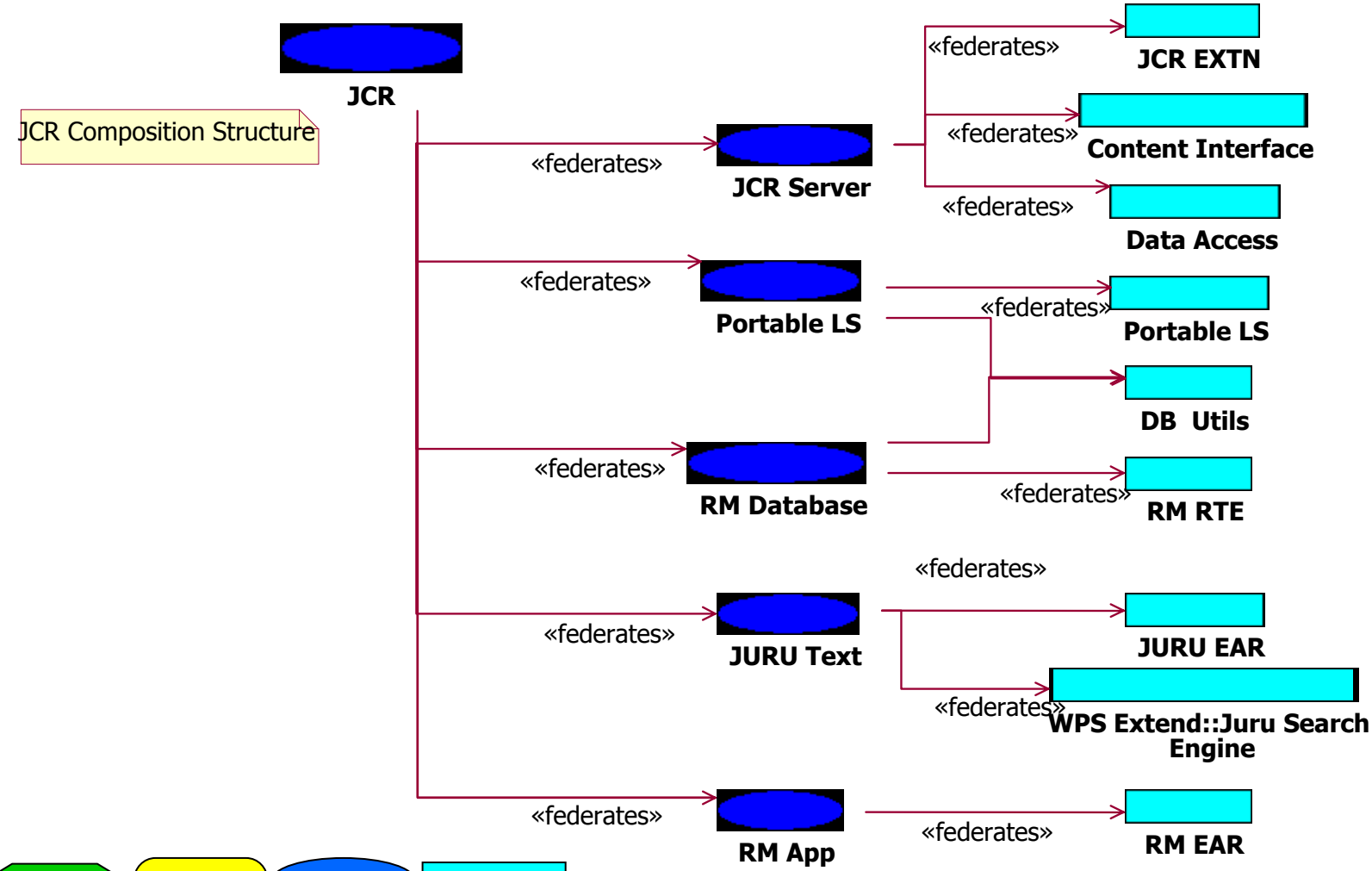
### Many SPX Component Models making up one SWG Component Model




(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

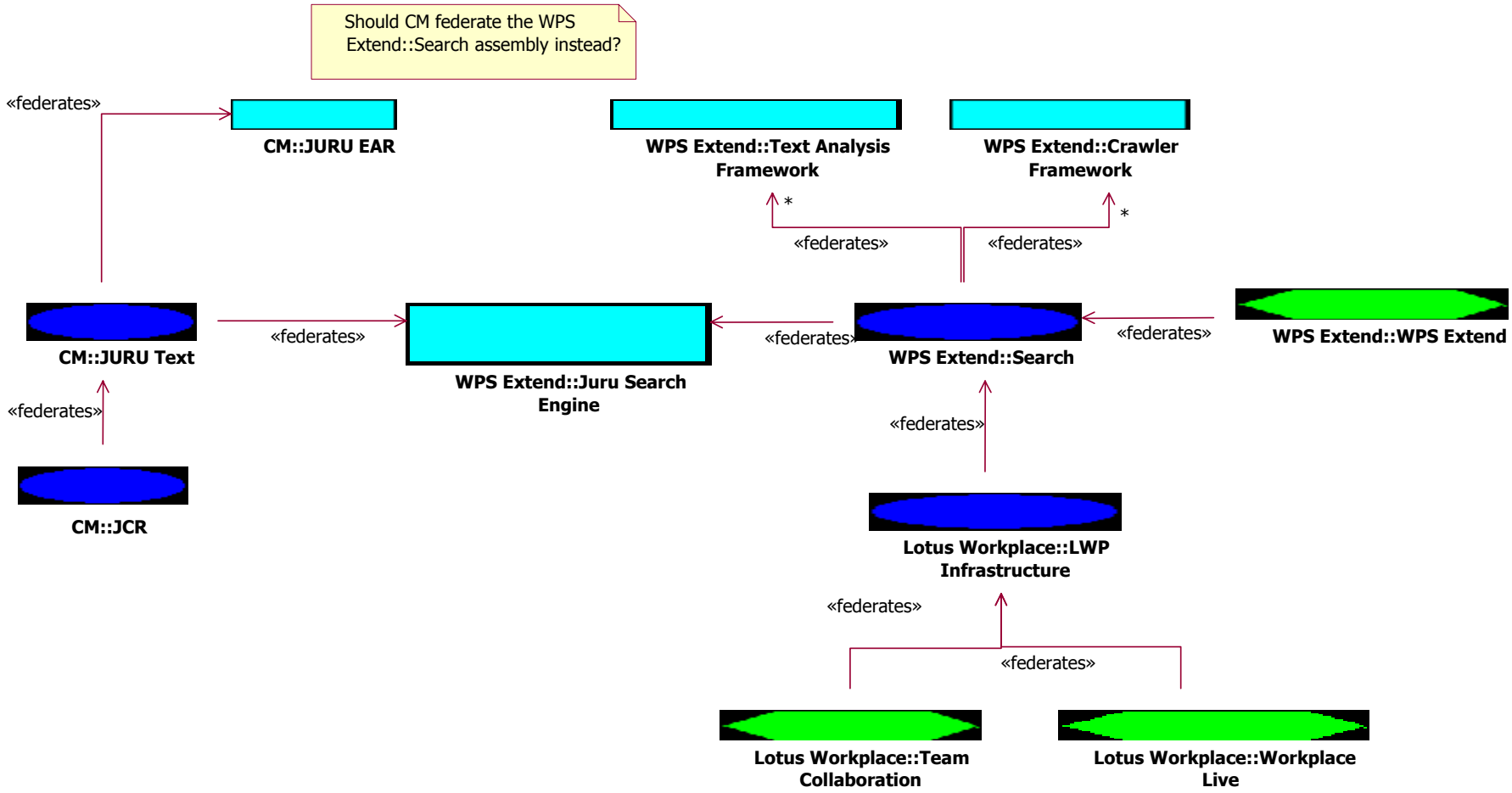
*What is the composition of the CM::JCR assembly?*




(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *Who uses WPS Extend::Juru Search Engine component?*

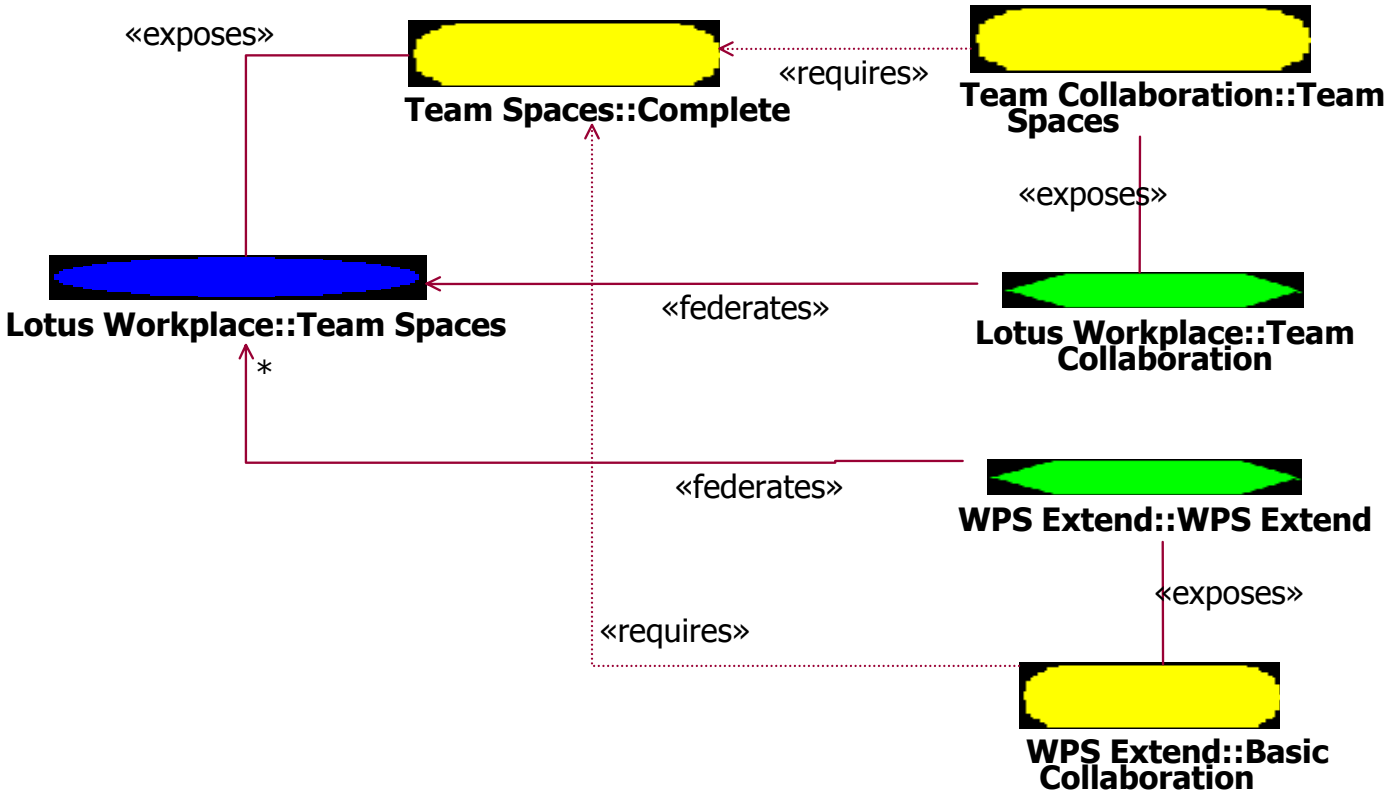



Offering | Feature | Assembly | C-Component

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *Who uses Lotus Workplace::Team Spaces assembly?*

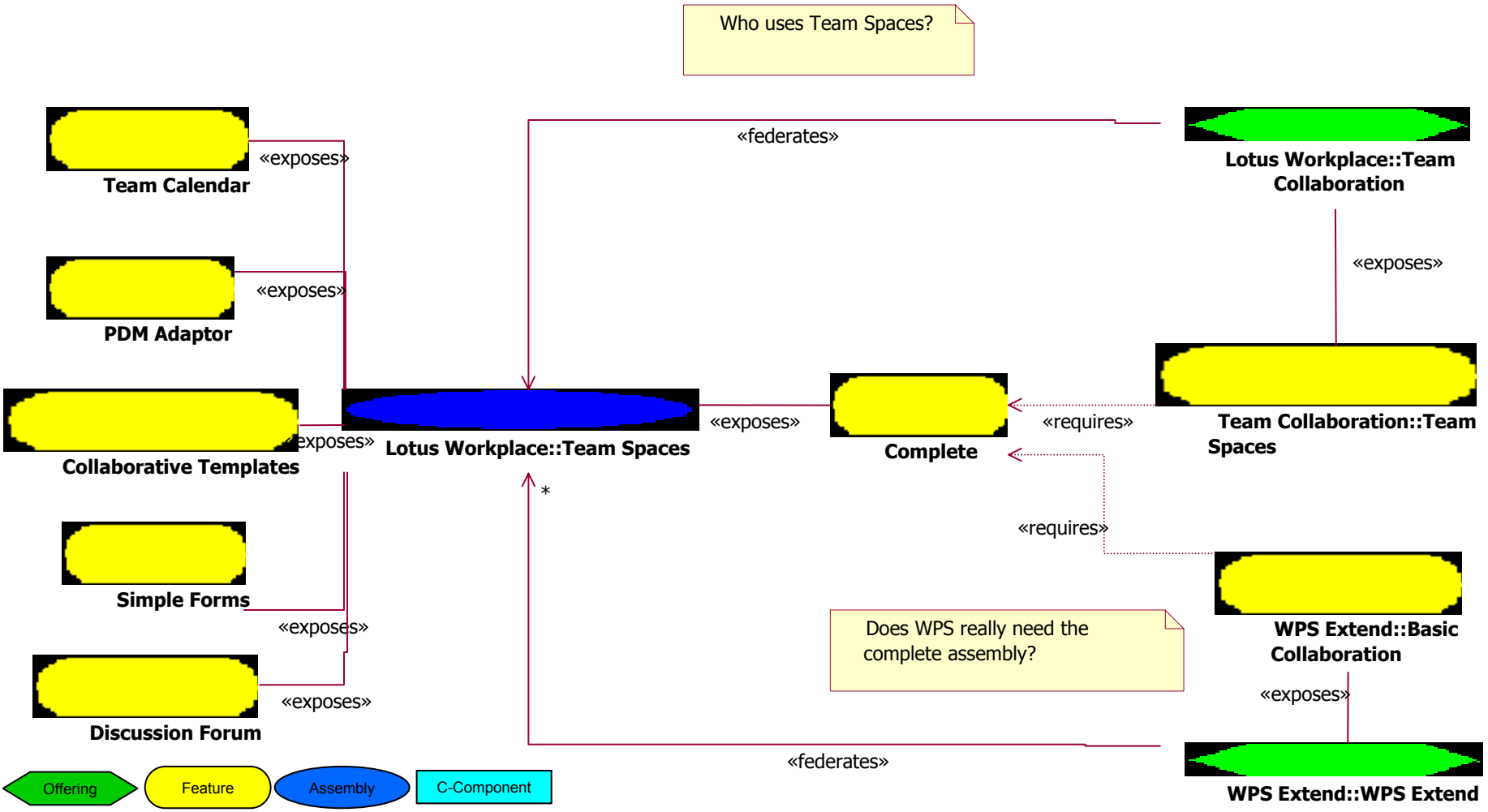



Offering | Feature | Assembly | C-Component

(C) Copyright IBM Corporation 2004, 2005

# Component-based Development

## *Does WPS Extend need the complete Lotus Workplace::Team Spaces assembly?*




(C) Copyright IBM Corporation 2004, 2005

# Summary

## Incremental and Integrated

**Enable customers** to easily, independently, and incrementally acquire and install IBM software platform capabilities through packaged offerings that provide a set of seamless software platform extensions.

## Agile and Composable

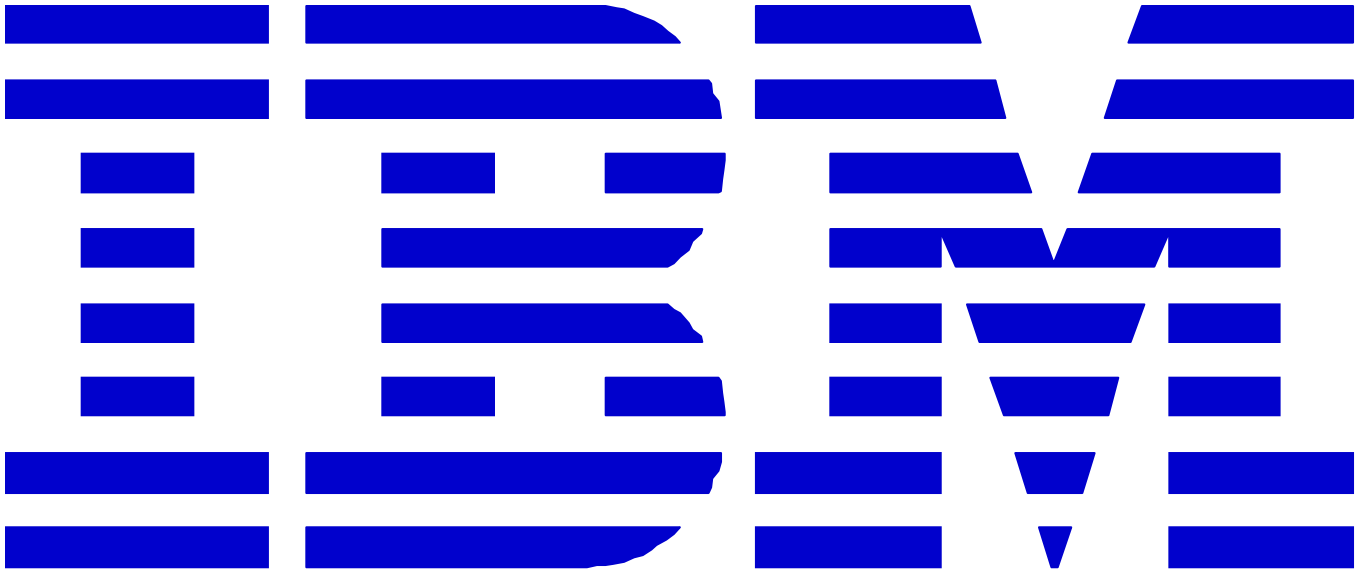
**Enable IBM and our partners** to rapidly adapt to shifting market pressures (enable market agility) by delivering packaged offerings and solutions to market through the assembly and reassembly of software platform capabilities into new and flexible configurations.

(C) Copyright IBM Corporation 2004, 2005

# Key Messages

- **The demands being placed on flexible business models require Flexible IT infrastructures. Services Oriented Architecture (SOA) will enable this transformation.**
- **IBM technologies are already successfully supporting businesses with Services Oriented Architecture.**
- **Componentization is a key enabler for On Demand.**

(C) Copyright IBM Corporation 2004, 2005



*ciaran_dellafera@us.ibm.com*

(C) Copyright IBM Corporation 2004, 2005

# SOA: The What and the Whys

- **It is an architecture, not a product**
- **It starts with the business requirements, allows for their changeability, and flexible nature**
- **It requires a framework or platform that connects the business processes to technical execution**
- **It is ubiquitous and agnostic; it connects one to many, many to many, irrespective of their native environments**
- **It creates an "ecosystem" predicated on "standardized" rules of participation**
- **By its nature, it is adaptive, and dynamic; it is designed to scale and extends the "usability" of any existing IT infrastructure**
- **Finally, it is a paradigm change, utilizing an integration based technology or "service bus", that effectively anticipates and optimizes IT**

- **With nearly 18 months of deep design and development, working with over 30 key clients from every key industry around the world, *here's why our clients are moving towards Services Oriented Architecture:***
  - Most businesses have grown organically, this has resulted in an amalgam of architectures, solutions, applications, data structures, messaging capabilities, often lacking any coherent governance, standardized programming or development models;
  - Businesses are immersed in a new era of mergers and acquisitions; this has significantly challenged time to completion deadlines;
  - Business pressures have created new accelerated "time to market" deadlines, two chief obstacles … lack of coherent business processes; lack of coherent IT execution.
  - These challenges are ubiquitous, they are industry and sector independent; while the specific semantic issues vary, the fundamental problem requires similar solutions
  - Any solution must, itself, be part of an agile environment, complementing, not further complicating the business and IT challenges

(C) Copyright IBM Corporation 2004, 2005