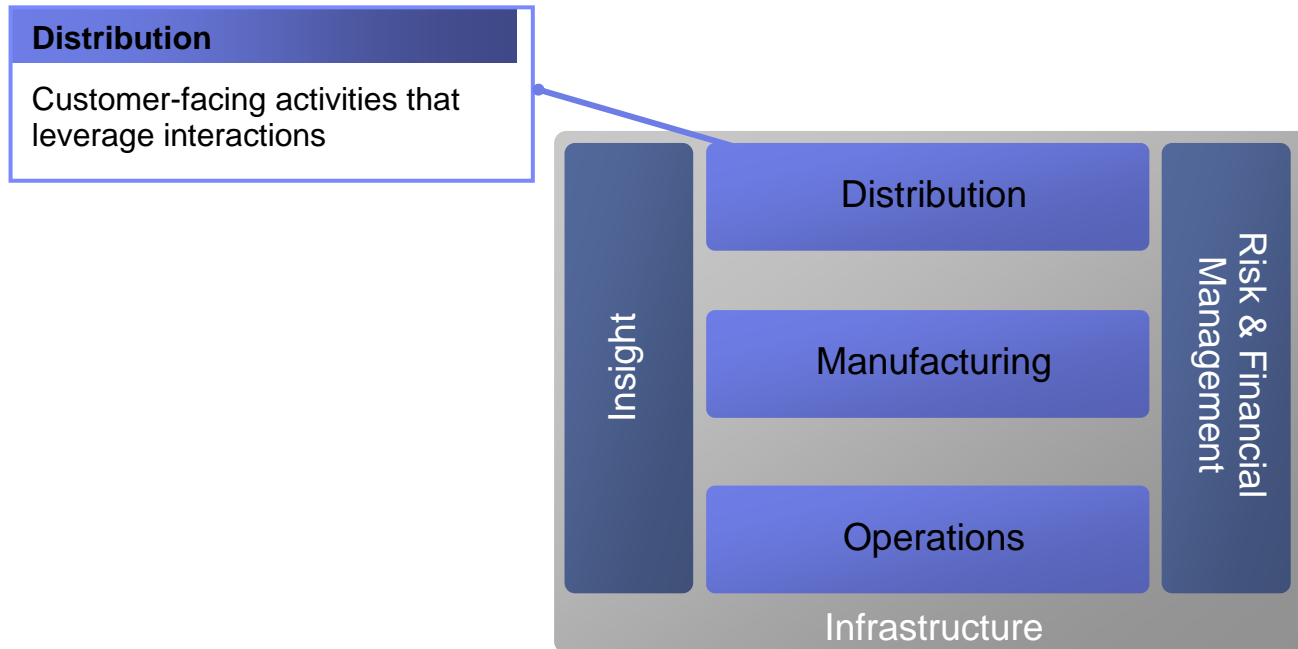


Six Financial Services Competencies

In addressing issues, financial institutions compete and excel across six common competencies, striving to deliver them **faster**, at **lower cost**, and with **higher quality** than the competition.



Source: IBM Institute for Business Value analysis

This presentation describes how IBM's products and assets help in the Distribution space

Lack of Integration across Channels Impedes Growth

Lack of continuity across channels

“I stopped by my branch with a query yesterday, but the call-center doesn't know anything about it.”



Lack of interoperation between applications

“Why is the Website asking me to type in information that they already have?”



Lack of consistency between channels

“The SMS service tells me I still have \$400 in this account !”



Why are the World's Leading Banks Transforming their Customer Experiences?

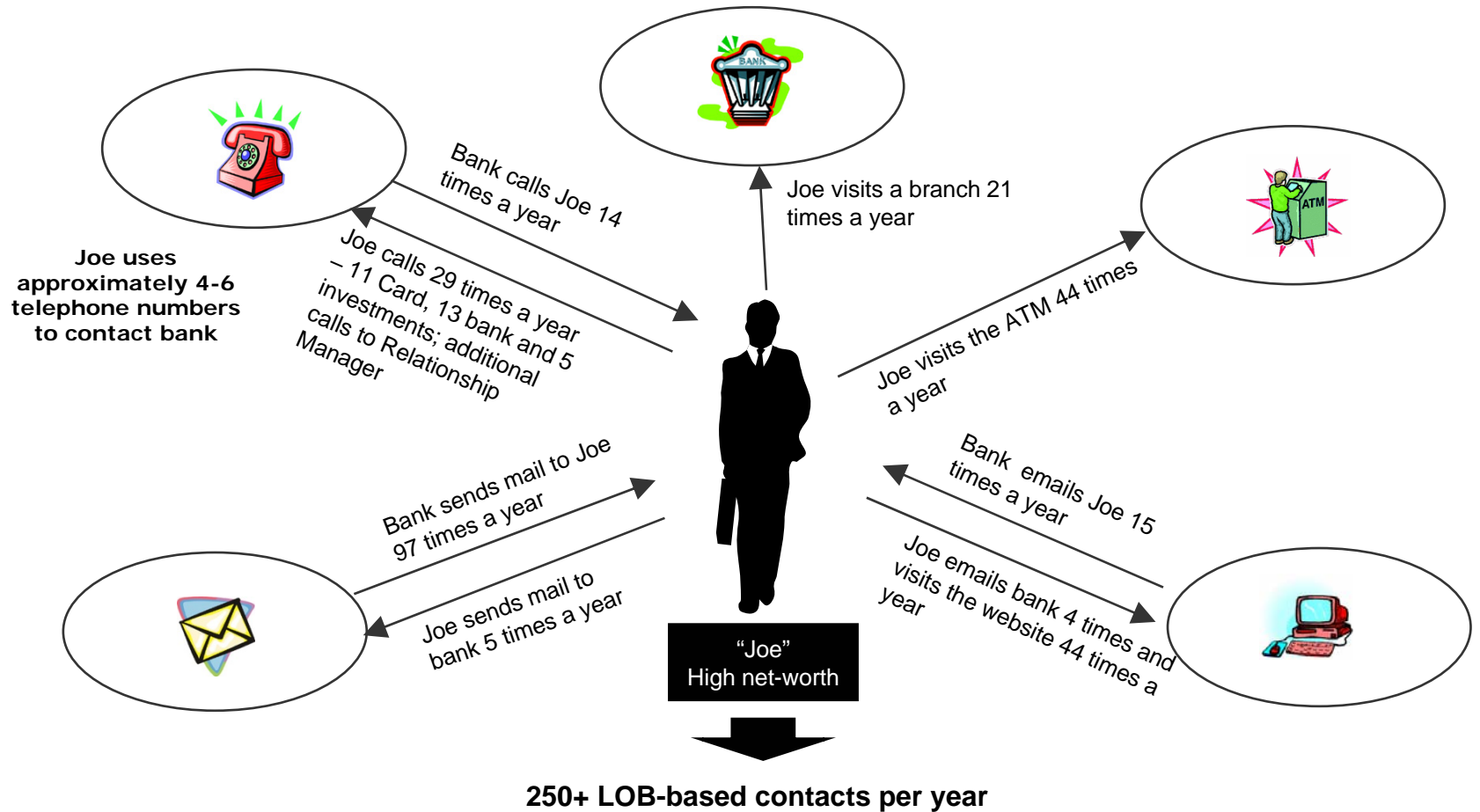


- Increased domestic and foreign **competition**
- Increased **choices** and ease of switching
- **Customer service** is improving in other industries (Retail)
- Increasing **multi channel** contacts are changing expectations
- **Constant Innovation** and improvement
- The **branch** is the hub of most banking activities and the most visible distribution outlet. The **online** experience is essential to managing the relationship and researching new opportunities.

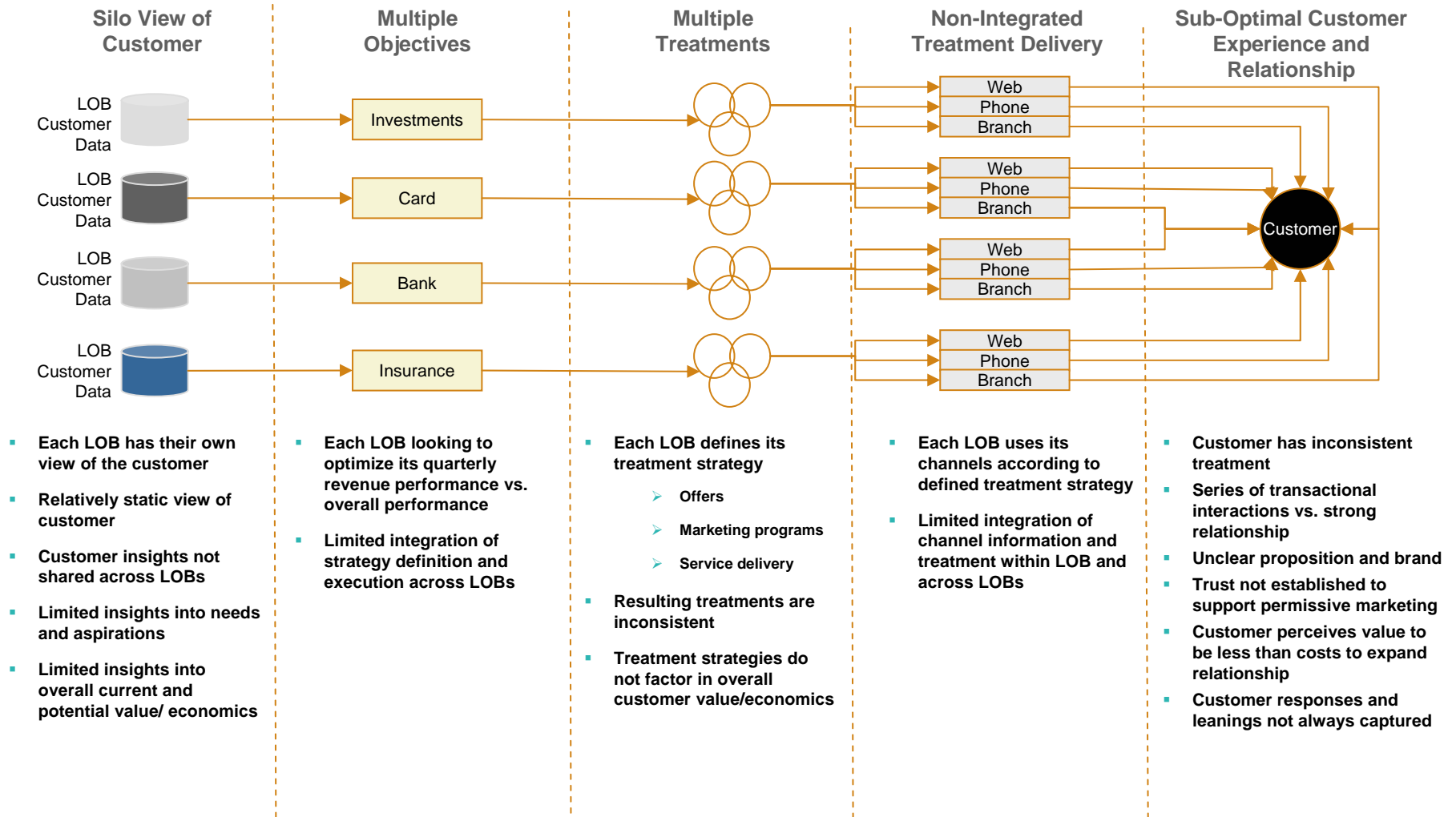
Current Challenges

- **Limited ability to focus on most profitable customers** while effectively serving mass market customers
- **Reactive sales approach** resulting in lost opportunities
- Staff must focus on **difficult processes**, not the customer
- Sales and service staff lack **customer management training and effective tools**
- Service leadership **hampered by processes**, checking reports, administrative duties
- Basic service, **referral and sales activities can't be tracked.**
- No **single view of customer** resulting in poor integration between branches, between channels

Lots of customer touches. Lots of opportunities.



Un-integrated LOBs drive fragmented customer interaction



- Each LOB has their own view of the customer
- Relatively static view of customer
- Customer insights not shared across LOBs
- Limited insights into needs and aspirations
- Limited insights into overall current and potential value/ economics

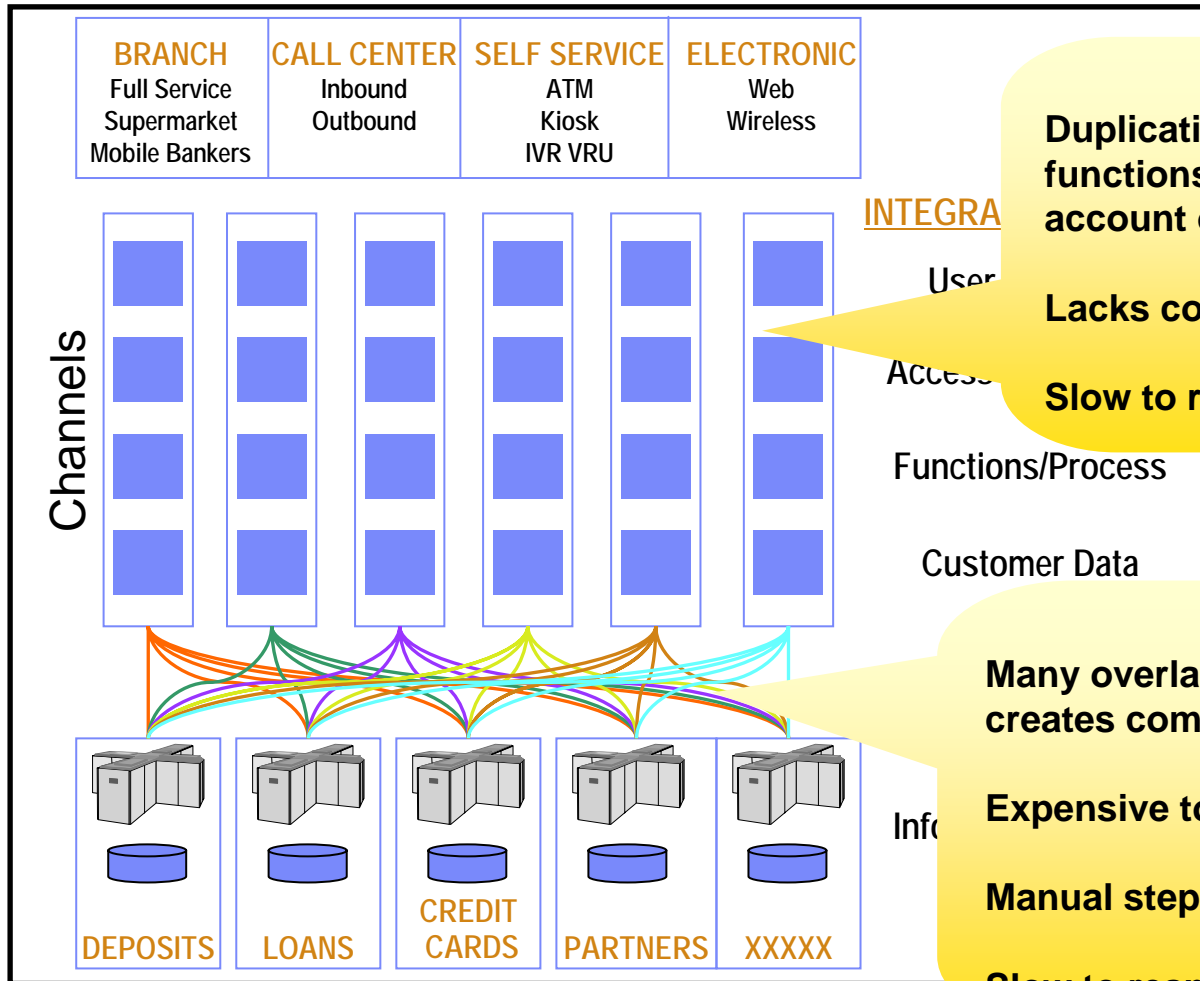
- Each LOB looking to optimize its quarterly revenue performance vs. overall performance
- Limited integration of strategy definition and execution across LOBs

- Each LOB defines its treatment strategy
 - Offers
 - Marketing programs
 - Service delivery
- Resulting treatments are inconsistent
- Treatment strategies do not factor in overall customer value/economics

- Each LOB uses its channels according to defined treatment strategy
- Limited integration of channel information and treatment within LOB and across LOBs

- Customer has inconsistent treatment
- Series of transactional interactions vs. strong relationship
- Unclear proposition and brand
- Trust not established to support permissive marketing
- Customer perceives value to be less than costs to expand relationship
- Customer responses and leanings not always captured

Most Banks' Channels: Line-of-Business & Product Orientated



BUSINESS CHARACTERISTICS

Duplication of common business functions. (e.g. many variations of account opening)

Lacks common customer view.

Slow to replicate good ideas.

TECHNOLOGY CHARACTERISTICS

Many overlapping points of integration creates complexity:

Expensive to maintain

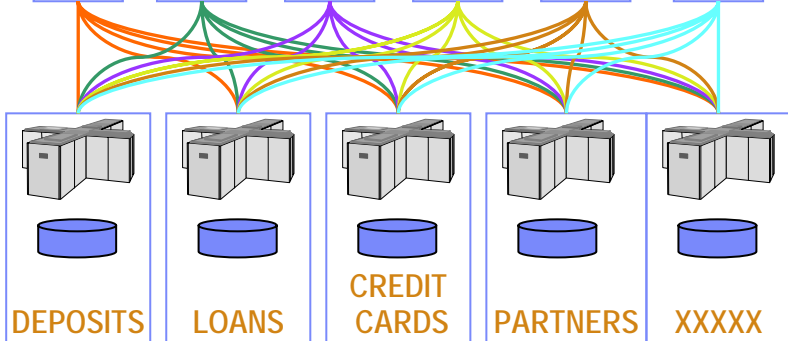
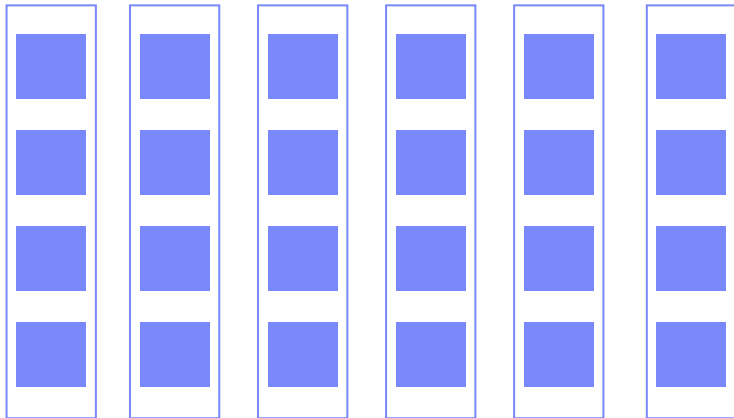
Manual steps in end to end processes

Slow to respond to business change.

Multi-channel Transformation to SOA

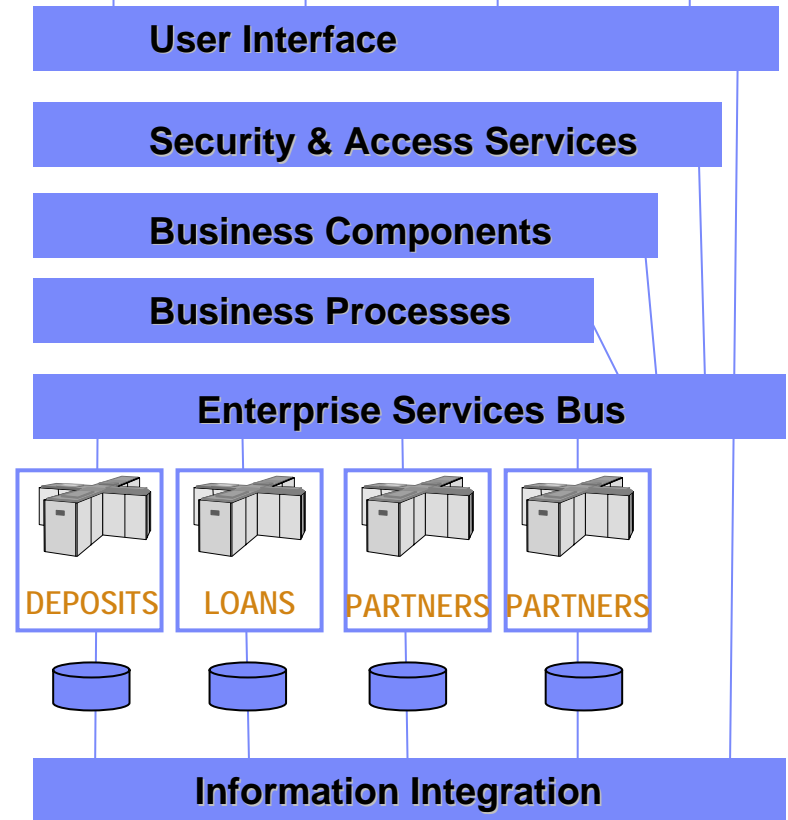


BRANCH	CALL CENTER	SELF SERVICE	ELECTRONIC
Full Service Supermarket Mobile Bankers	Inbound Outbound	ATM Kiosk IVR VRU	Web Wireless

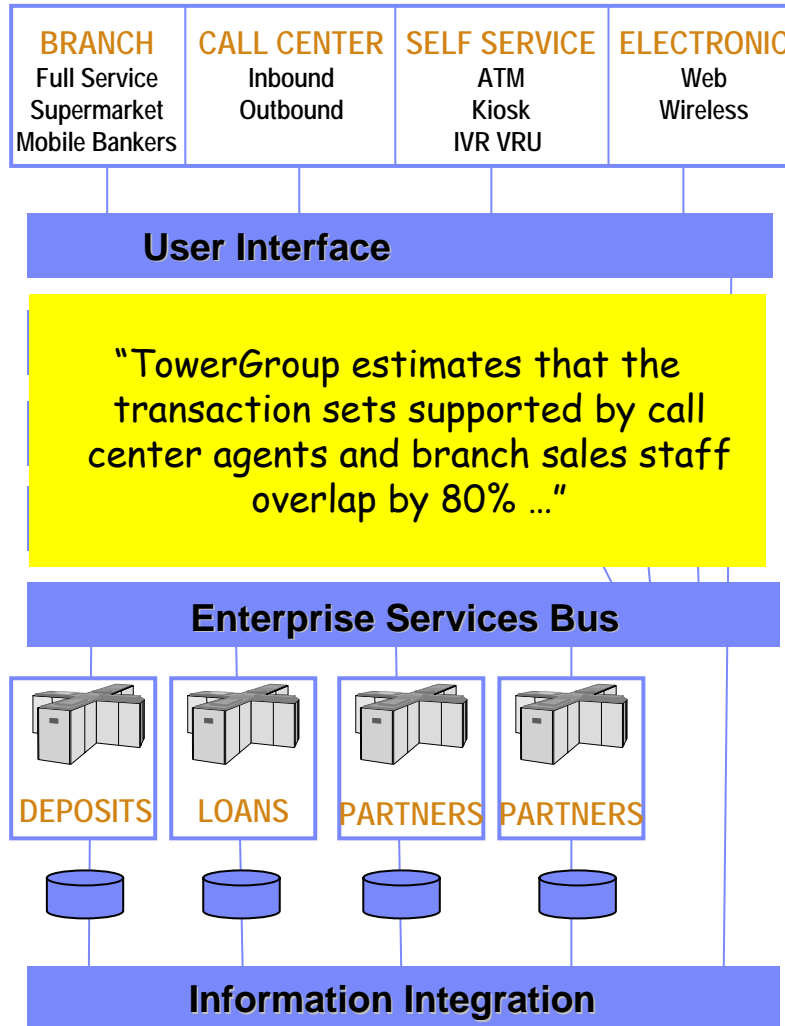


Multi-Channel Architecture (MCA)

BRANCH	CALL CENTER	SELF SERVICE	ELECTRONIC
Full Service Supermarket Mobile Bankers	Inbound Outbound	ATM Kiosk IVR VRU	Web Wireless



An Integrated Channel Infrastructure



CHARACTERISTICS

- Consistent look & feel across channels anytime, anywhere, anyway
- Common views of customer information
- Reuse of functions, processes & technology
- Ability to respond quickly to market conditions

REQUIRES

- An end-to-end, cross-silo solution that leverages current core competencies
- An infrastructure that is open, flexible, and secure
- Tools & platforms that permit reuse, support standards, and provide quick development
- Preservation & leverage of existing customer investments

IBM's Front Office Approach



▪ **Better Rich and Thin**

▪ **Build or buy business logic components shared across all channels ... reuse**

▪ **Information as a common service.**

What we mean by “*Solution for Channel Infrastructure*”

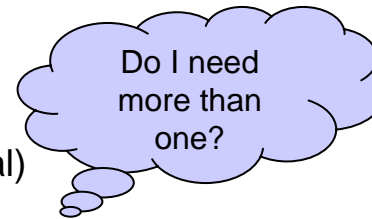
- Considerations for building channel applications
 - ✓ Client platforms (Thin / Rich / Fat / Phone / e-Mail / Mobile / ATM / ...)
 - ✓ Unified desktops for Sales and Service applications – Designed around the user / role
 - ✓ Efficient Teller applications
 - ✓ Specialized development tooling
 - ✓ ISV applications
 - ✓ Support for printing, scanning and financial devices
 - ✓ At-the-glass integration strategy
 - ✓ Security and Single Sign On (SSO)
 - ✓ Role-based capabilities
 - ✓ Operations with low bandwidth or unreliable networks
 - ✓ Channel Application Architecture
 - ✓ Server Platforms (p-series vs. z-series)
 - ✓ Business Process Management
 - ✓ EAI and SOA Infrastructure
 - ✓ Access to Core Systems
 - ✓ Common Enterprise Data (Customers, Account opening, Product data...)

Selecting a Channel Infrastructure Solution

What set of client platforms will be supported?

- Internal Users
 - ✓ Client platform
 - Browser based (e.g. WebSphere Portal)
 - Rich (e.g. Lotus Expeditor)
 - Legacy (VB, Java Swing, 3270)
 - ✓ Workstation OS
 - Windows
 - Linux
 - ✓ Virtualization

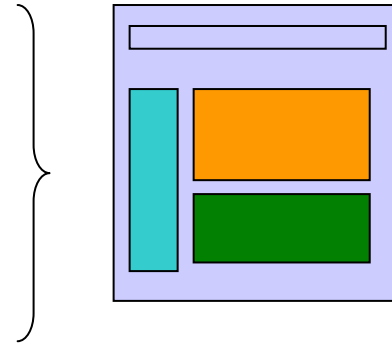
- External Users
 - Internet
 - Phone
 - Mobile
 - ATM



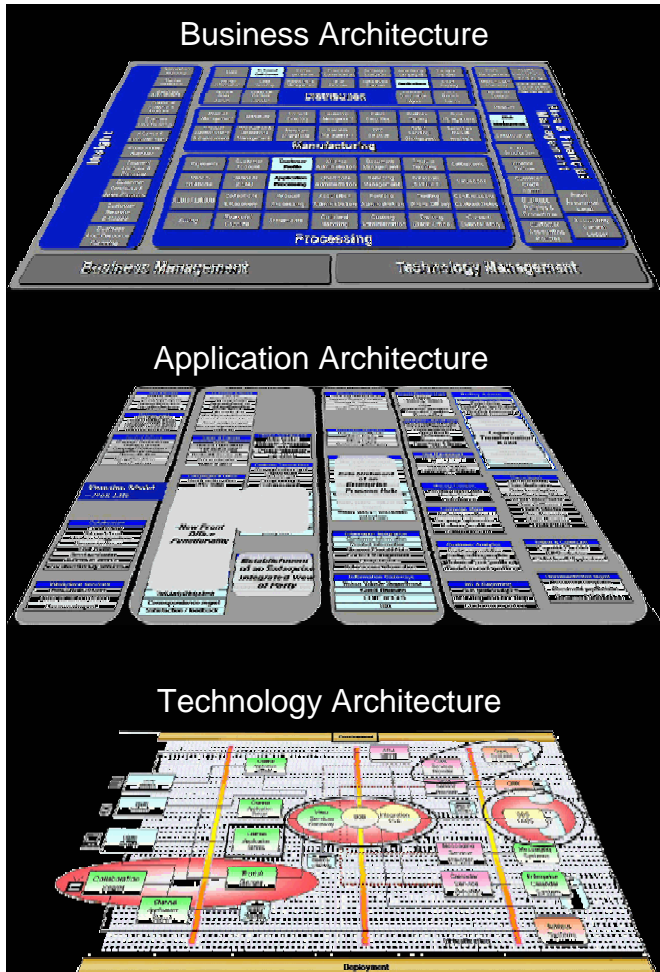
Building an Effective **Multi-Channel** Infrastructure

How will the desktop be optimized?

- Unified desktop applications
 - ✓ At the glass integration strategy
 - Visual Basic
 - Java Swing / SWT
 - HTML
 - Web 2.0
 - Portlets
 - ✓ How will legacy desktop applications be integrated to...
 - Reducing the time spent switching between applications
 - Minimize duplicate data entry
 - Provide as seamless of a user experience as possible
 - Provide smooth transition as new applications are rolled out
 - ✓ Provide for Single Sign-on and role-based authorizations and user experience



The End-to-end IBM Approach: Models, Tools, Middleware



Component Business Models

Information Framework (IFW)

WebSphere Business Modeler

Rational Application Developer

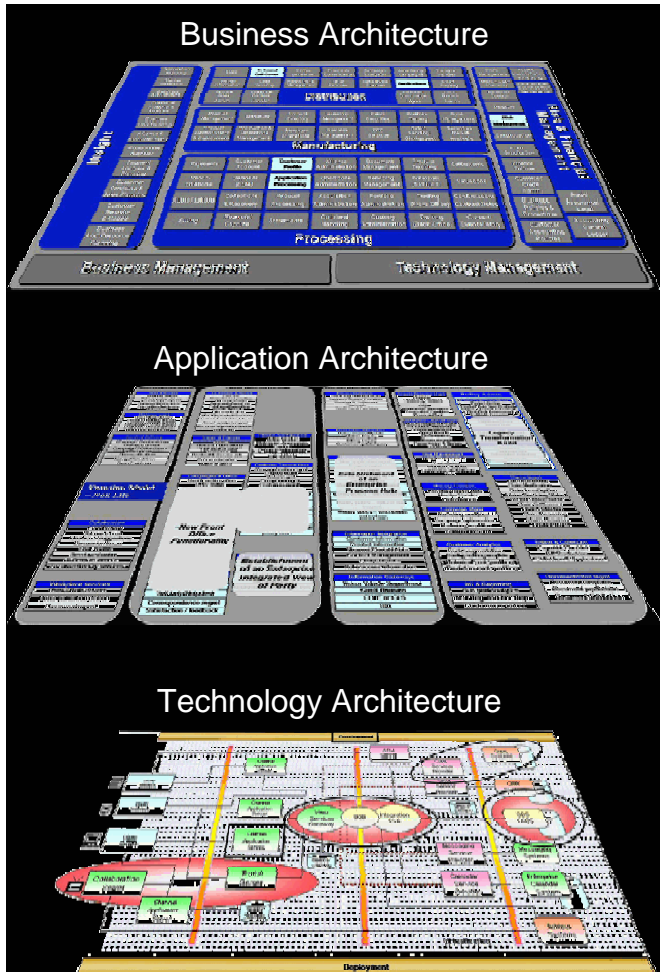
WebSphere Integration Developer

WebSphere Multi-channel Bank Transformation Toolkit

WebSphere Application Server & WebSphere Process Server

Tivoli Configuration Manager & Monitoring

The End-to-end IBM Approach: Models, Tools, Middleware



Component Business Models

Information Framework (IFW)

WebSphere Business Modeler

Rational Application Developer

WebSphere Integration Developer

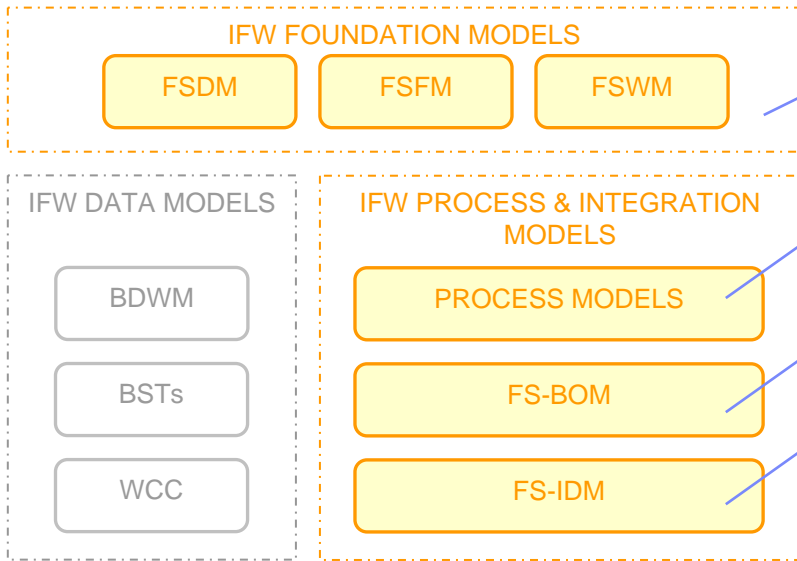
WebSphere Multi-channel Bank Transformation Toolkit

WebSphere Application Server & WebSphere Process Server

Tivoli Configuration Manager & Monitoring

The Information FrameWork (IFW)

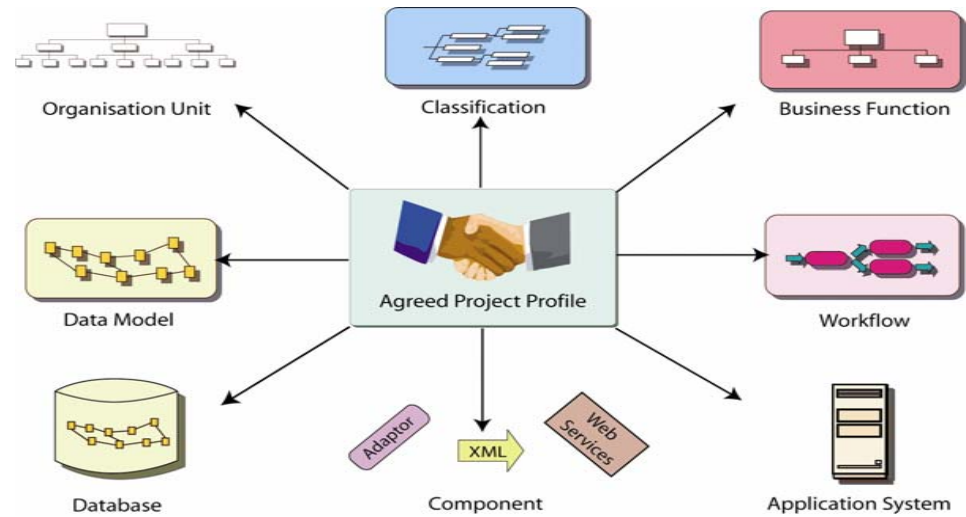
A framework of related business models, describing different aspects of the analysis and design required to support a financial institution



IFW FOUNDATION MODELS:
SCOPING AND DEFINITION OF BUSINESS TERMS

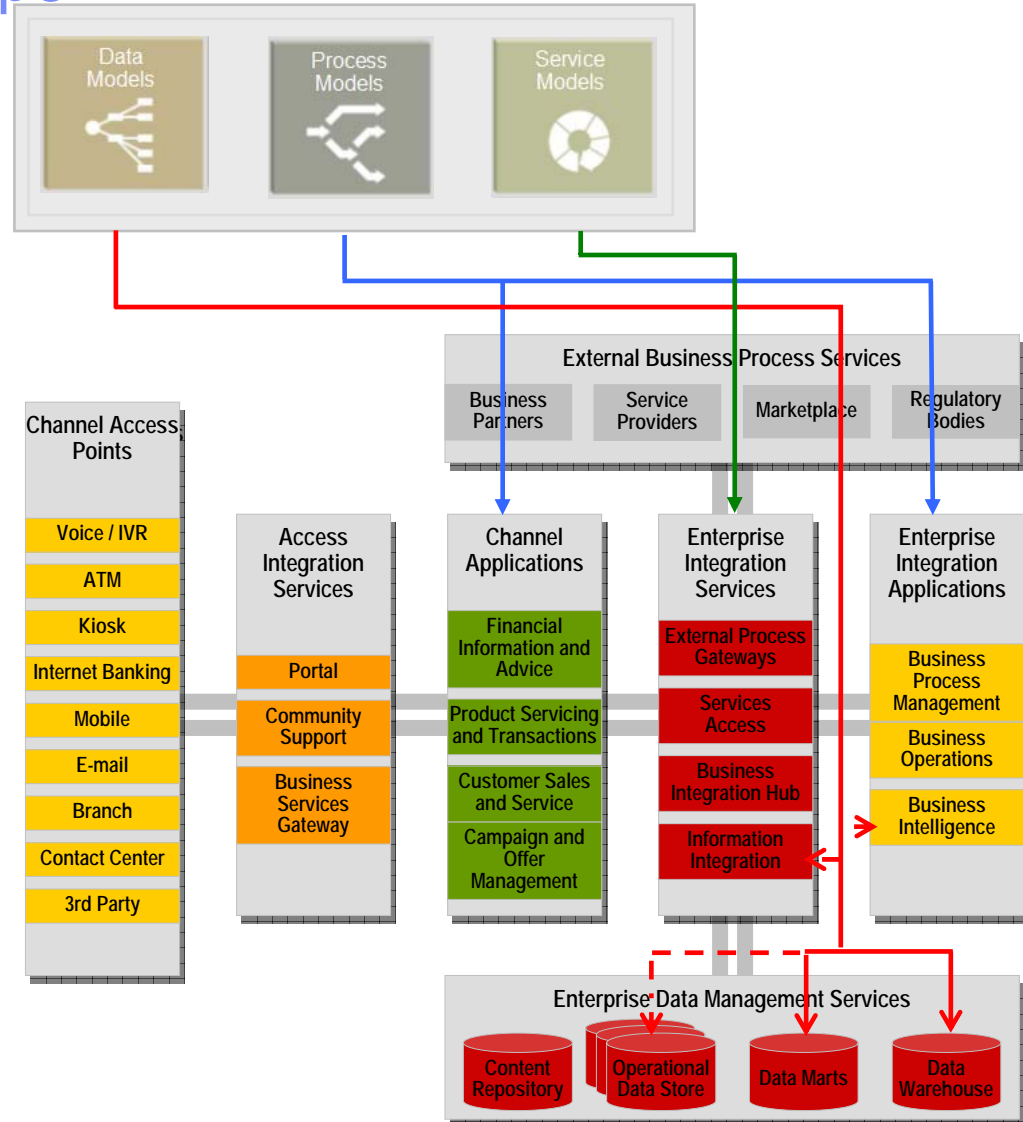
IFW PROCESS MODELS:
DETAILED ANALYSIS OF BUSINESS PROCESSES

IFW INTEGRATION MODELS:
ANALYSIS AND DESIGN OF REUSABLE SERVICES

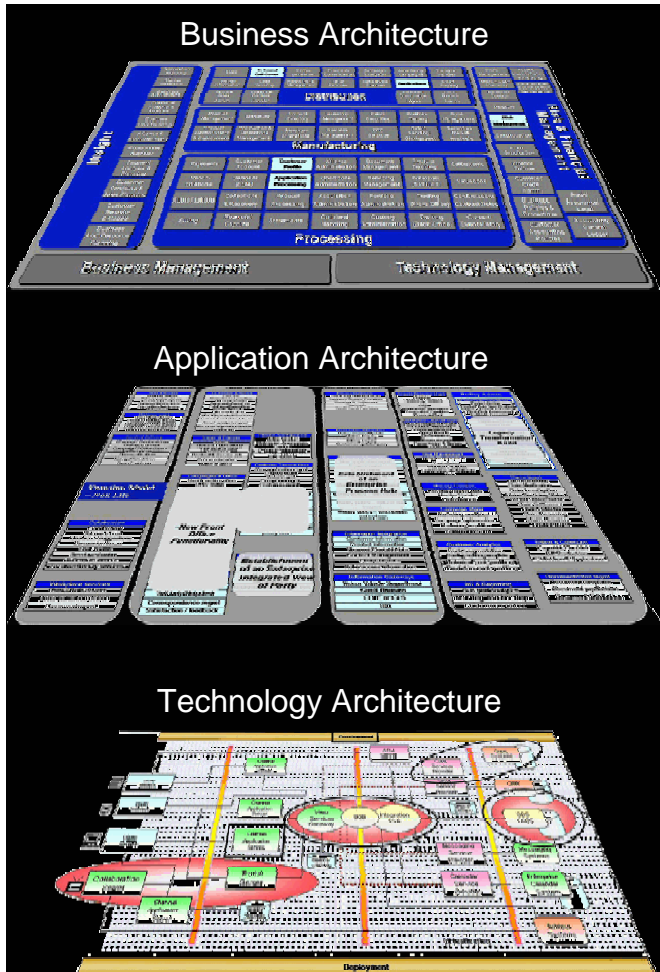


Using the IFW Banking Models across a typical Financial Services application landscape

- Achieve consistent data models across products, channels and external business partners
- Achieve streamlined, consistent business processes across products and channels
- Improve business processes internally and across external business partners
- Achieve straight-through end to end processing
- Facilitate consistent provision of products through multiple channels
- Define workflows and processes independent of line of business, product, channel, organization structure and technology



The End-to-end IBM Approach: Models, Tools, Middleware



Component Business Models

Information Framework (IFW)

WebSphere Business Modeler

Rational Application Developer

WebSphere Integration Developer

WebSphere Multi-channel Bank Transformation Toolkit (BTT)

WebSphere Application Server & WebSphere Process Server

Tivoli Configuration Manager & Monitoring

WebSphere Multi-channel Branch Transformation Toolkit: A Multi-Channel Framework for Reusing Business Components

BRANCH	CALL CENTER	SELF SERVICE	ELECTRONIC
Full Service Supermarket Mobile Bankers	Inbound Outbound	ATM Kiosk IVR VRU	Web Wireless

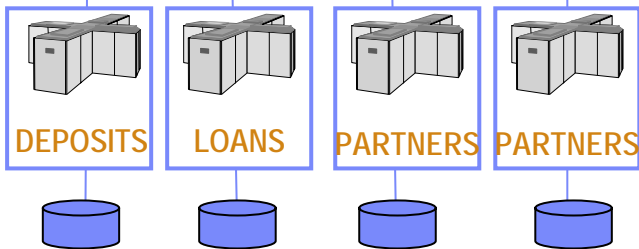
User Interface

Security & Access Services

Business Components

Business Process

Enterprise Services Bus



Information Integration

- Provides a proven pattern for front office application development
- Introduces a well-defined, repeatable end-to-end structure and nomenclature
- Integrates with open source and standards-based programming models (J2EE, Eclipse, BPEL, Struts/JSF, Web Services)
- Provides a graphical workbench and visual development tools for building multi-tiered process-oriented front office applications
- Establishes a productive environment with pre-selection of programmatic settings to enable quick, reliable, and consistent creation of new applications
- Addresses customers' needs for an efficient, flexible programming model that reduces the needs for a highly technical programming staff

BTT Information Technology Benefits

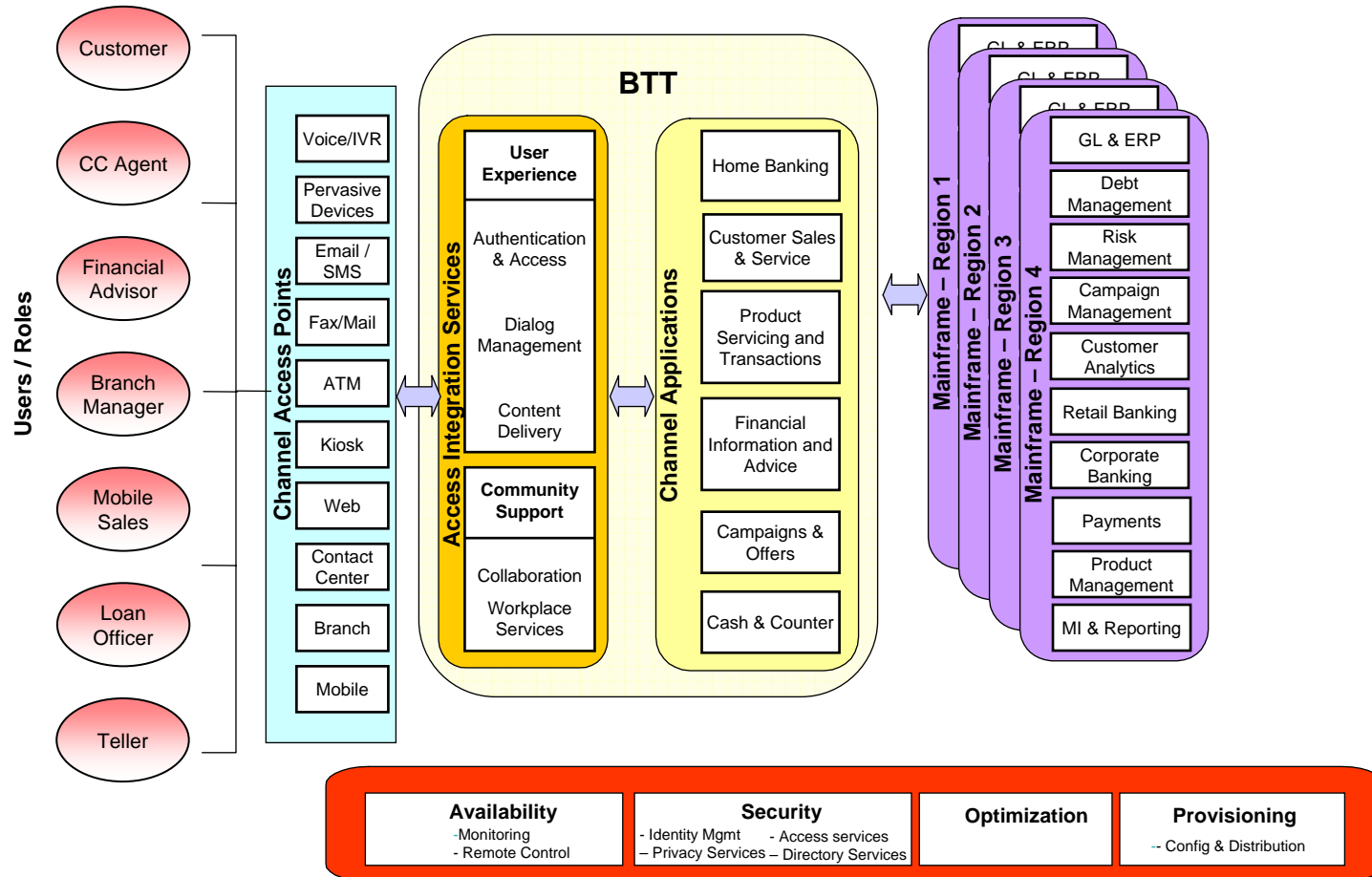
Information Technology Benefits

- **A proven pattern for front office application development with a well-defined, repeatable end-to-end structure and nomenclature**
- **A graphical workbench and visual development tools for building multi-tiered process-oriented front office applications**
- **Integrates with open source and standards-based programming models (J2EE, Eclipse, BPEL, Struts, Web Services) and innovative WebSphere component-based technology**
- **Provides access to and integrates with existing business function**
- **Protects investment in infrastructure by enabling extension of existing transactional systems to branches**
- **Addresses customers' needs for an efficient, flexible programming model that reduces the needs for a highly technical programming staff**
- **Integrates closely with SOA solution**

A Major European Bank: Benefits of BTT

- All of The Bank's Channel Applications are Built on BTT
 - Home Banking
 - Teller
 - ATM
 - Contact Center
- **Consistent and Seamless User Experience** Across Multiple Countries with Separate Core Systems
- **Simple, Agile and Predictable** Development Process Led to
 - Outstanding Performance of the Bank in Retail Channels
 - High Confidence that IT can Complete Projects On Schedule
- Standardization led to Proven and Stable Runtime Environment
 - **Less Risk** when Deploying New Releases to Production
- Ability to "Get a Mainframe Programmer Productive in 4 weeks"
 - High Level of **Flexibility** in Obtaining and Allocating Development Resources
- Ability to Consistently Leverage Investment in Channel Applications over Multiple Channels
 - **Faster Return** on Development Investments
- With a **Smooth Migration Process** for each Release of BTT
 - Existing Systems Developed over **Hundreds of Man Years** Can Benefit from New BTT Features
 - Ability to Maintain a **Single Technically Current Development Framework** Over Long Periods of Time.

A Major European Bank: A Simple and Efficient Multichannel Architecture Across Countries



Wing Lung Bank: A Multichannel Success

The Challenge

Lower operating costs as net interest margins squeezed due to greater competition

Increase non interest income thru cross selling on Internet and Branch

Change in function for the branch – more of retailing function

The Solution

Replaced 300+ terminals for 34+ branches – fully integrated system on Java technology for its Callcentre, Internet and Branch.

Common foundation for all retail customer services channels – increase cross selling and customer satisfaction

Key Business Benefits

10 – 20% (savings) in Costs.

Quantum leap in terms of functionality Branch staff now have through a consistent and friendly interface

Less interruptions on customer services across Callcentre, Internet and Branch

Shorten time to market on new Products



“IBM’s consultants/engineers helped us put in place a retailed banking platform that streamlines internal processes and gives staff a complete picture of their customers which is key to x selling and increasing customer satisfaction”

-- Hong Kong 7April

Quote from Exec Director/GM Mr C S Chung

ANZ: Branch Banking in Australia and New Zealand

- Challenge
 - Renew critical branch applications but avoid single – channel obstacles
 - Create common customer experience across multiple channels
- Approach
 - Establish channel integration solution
 - Attack high volume channels and tie to integration framework
 - Replicate approach to renewal of additional applications and channels
- Benefits
 - Avoidance of duplicate channel infrastructure
 - Savings from web-based teller
 - Projected savings of 30% for additional development in other channels

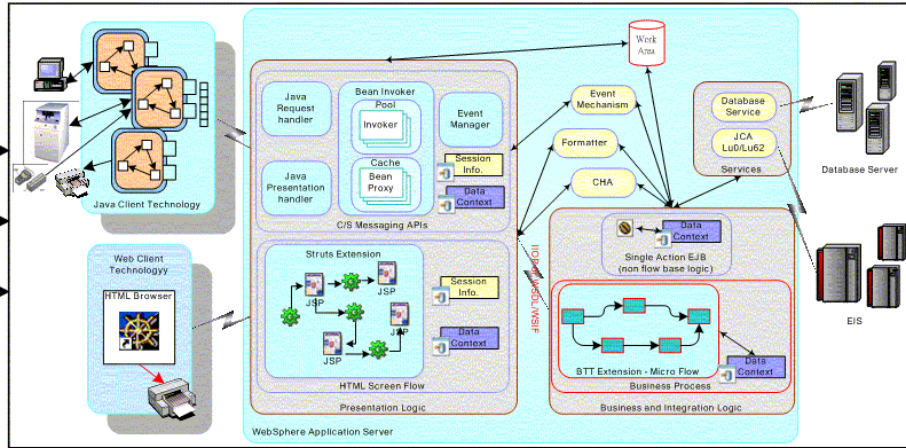
“ANZ, a large Australian based bank, selected IBM to work with them to develop a new teller application and infrastructure for their retail branch network using IBM WebSphere e-business software. The new application was deployed in 2004, replacing existing legacy technology and improve the level of customer service provided by the tellers. ANZ also intend in the future to leverage and reuse components of the teller platform for other multi-channel initiatives within ANZ such as Internet Channel.”

2007, ANZ is currently in a project to rollout the branch solution to New Zealand.



A Major Hong Kong Bank: A Complete Solution

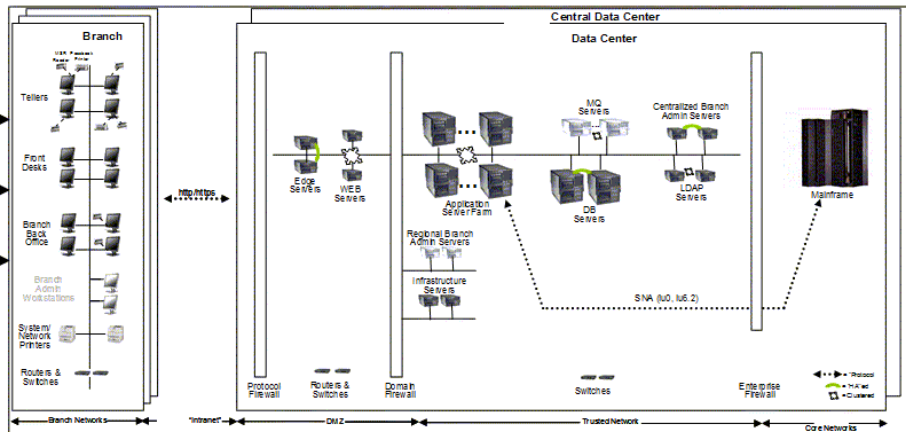
IBM Developed BTT Framework
Standard Based
End-to-end Framework



Branch Transformation Application Based on BTT Framework

- Proven
- High Performance
- Resilient & Highly Available
- Secure
- Scalable, Flexible Expandable

IBM Computer Hardware
IBM Computer Software
Synerav with existing Systems



Branch Transformation Infrastructure using IBM Hardware and Software

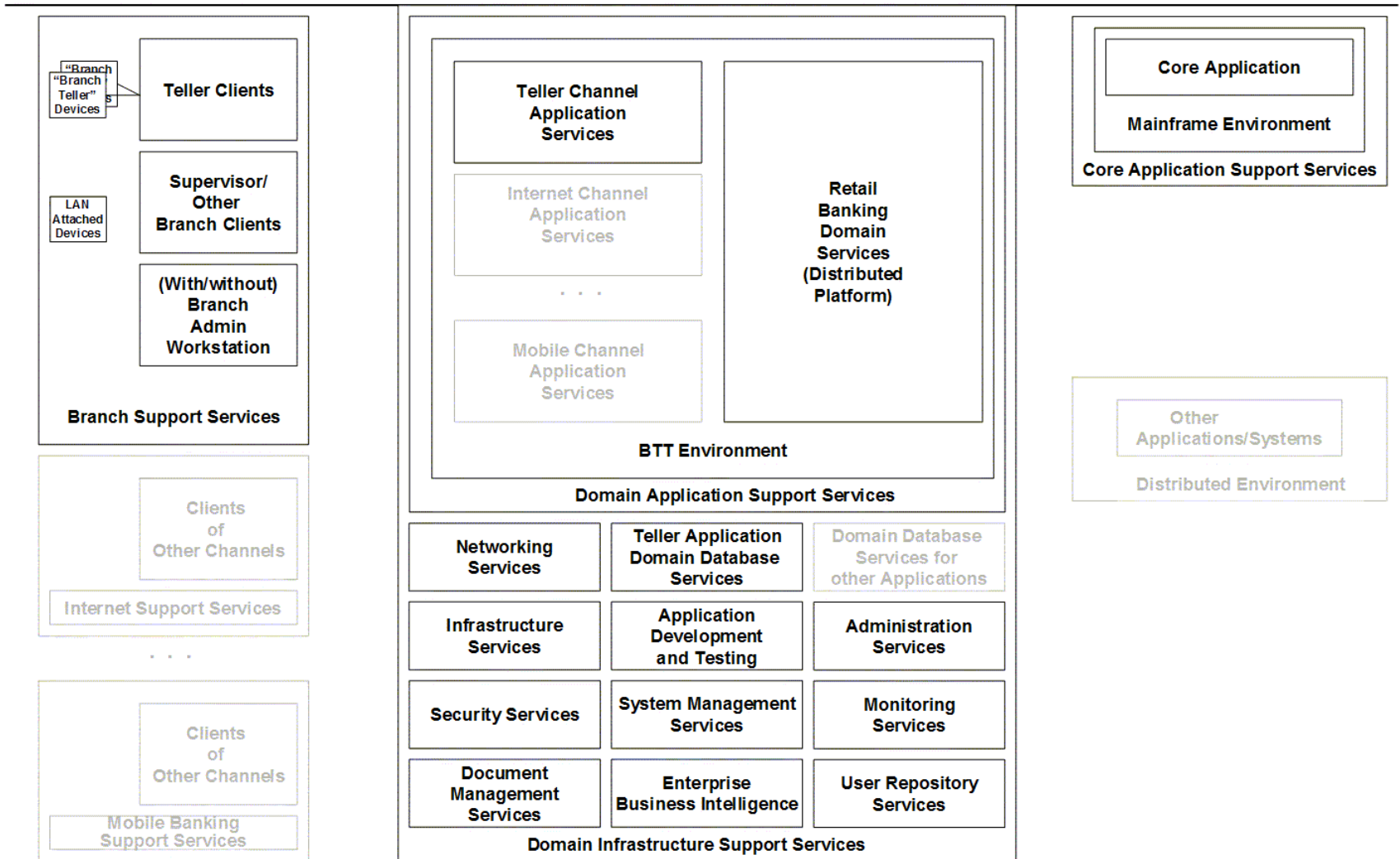
- Support Current Requirements
- Ready for Future Enhancements
- HW/SW/Framework Synergy
- Strong LAB BTT Support
- Proven Infrastructure
- Strong Partnership with

Colleagues
IBM Local Support Team
IBM LAB Support



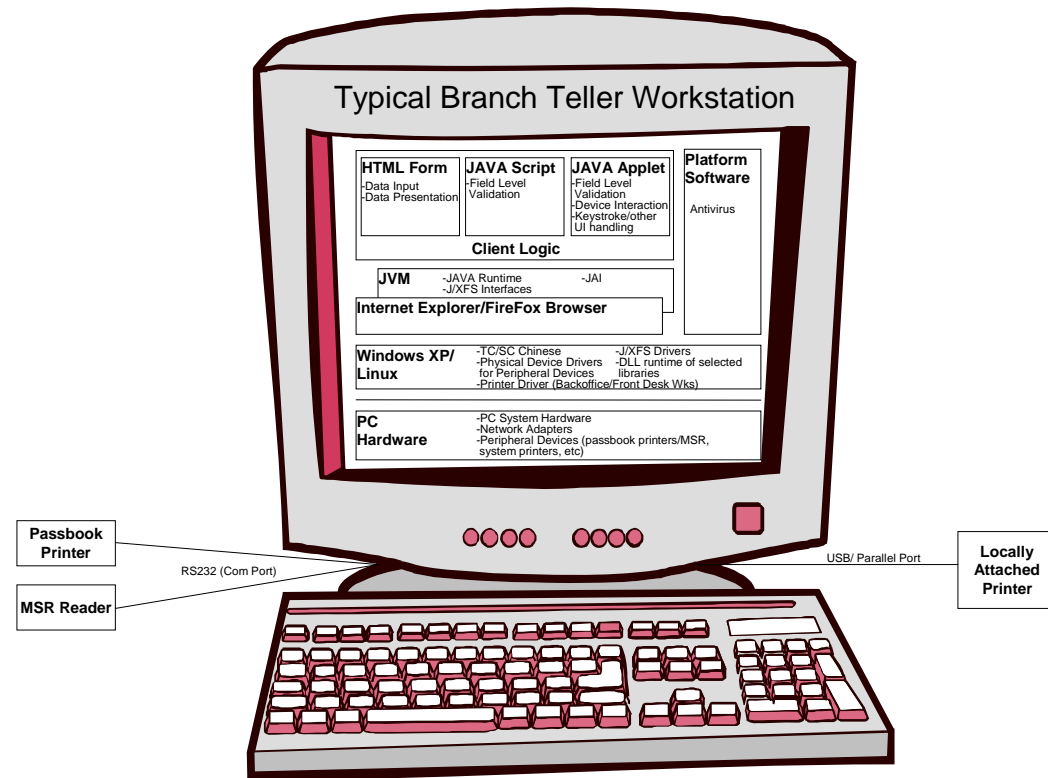
IBM Partnership

A Major Hong Kong Bank: A Multichannel Solution

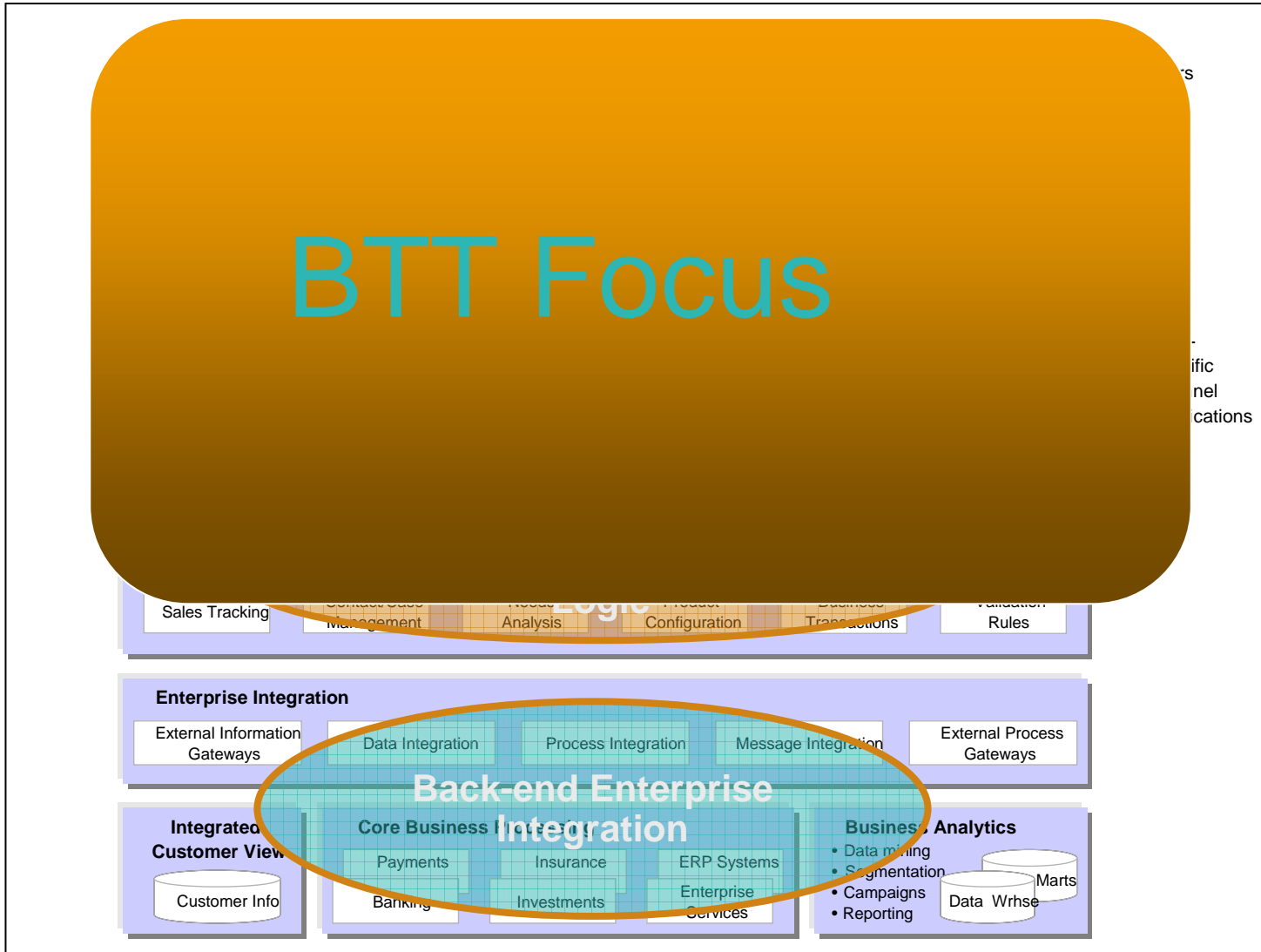


A Major Hong Kong Bank: Thin Client Teller

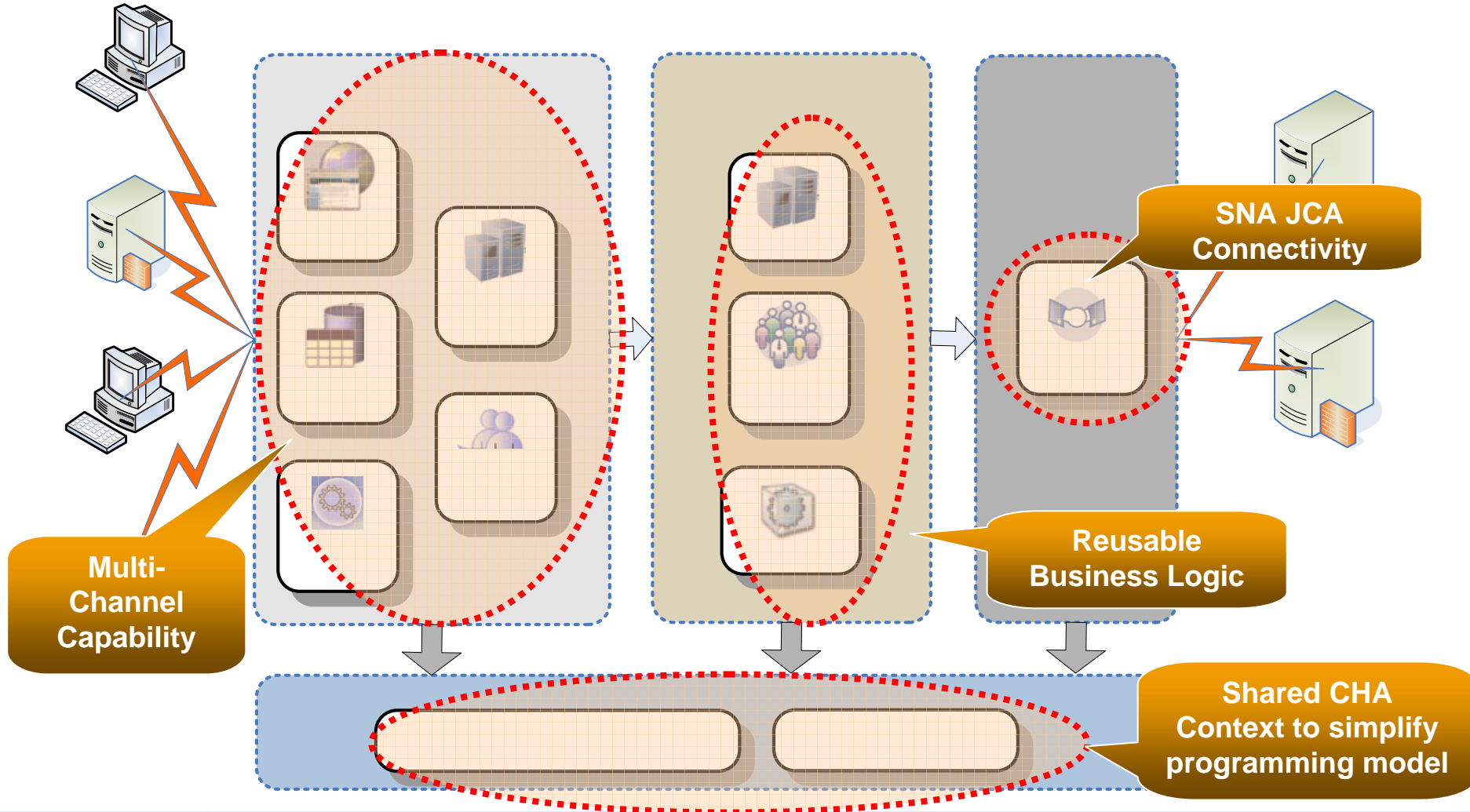
- Thin Client Architecture
- Built on BTT Framework
- Supports 300 branches and 6000 concurrent users
- 95% of transactions involving mainframe access take less than 3 seconds
- Most BTT Customers Implement Teller on a Rich Client Platform (Lotus Expeditor)



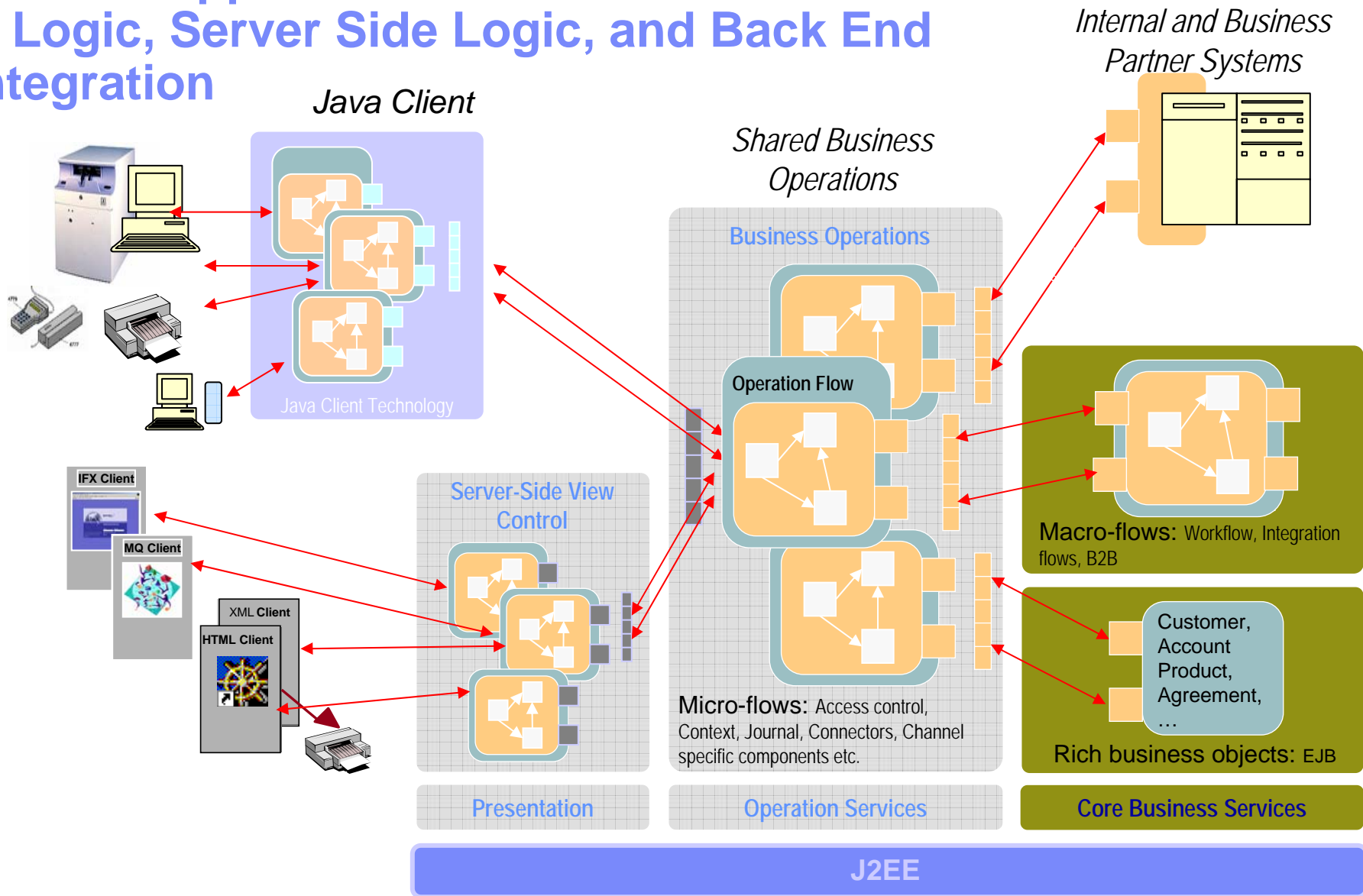
BTT Focus



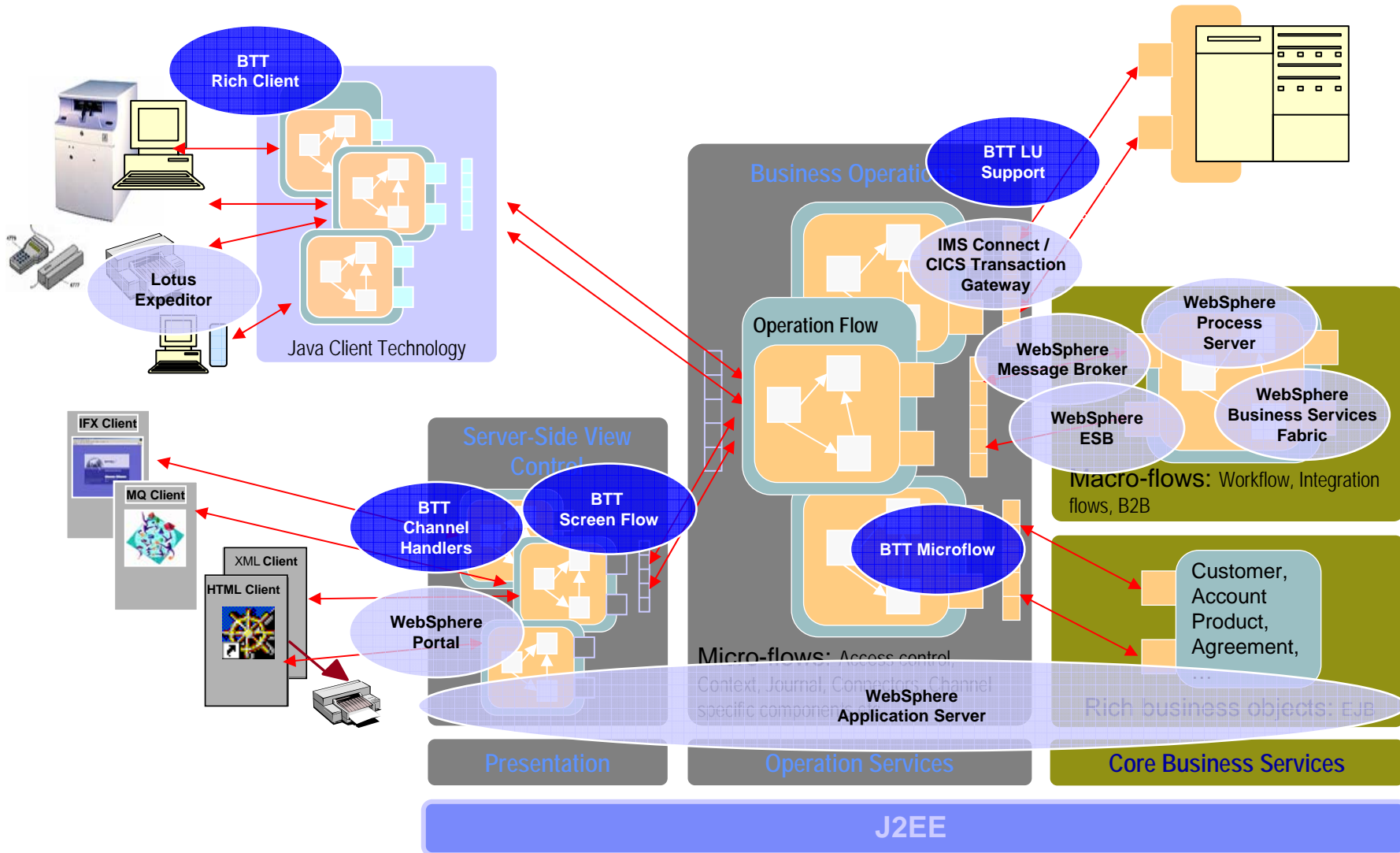
BTT 5.2 Runtime Architecture



Channel applications consist of Client Presentation & Logic, Server Side Logic, and Back End Integration



Flow and Control are key aspects of all three layers of a Channel Application.



Flow and Control: Product Positioning of IBM Products and Assets

	BTT	ESB (WebSphere Enterprise Service Bus)	WebSphere Process Server (Product includes ESB capabilities)	WebSphere Business Services Fabric (Product includes WPS capabilities)	WebSphere Message Broker
Screenflow	★				
Channel Application Microflow	★				
Service Composition		★			
Services Orchestration			★		
Macroflow (BPEL & BSM)			★		
Dynamic Selection of Web Services				★	
Routing and Transformation		★			★

★ = Recommended SWG Product

Client requirements for:

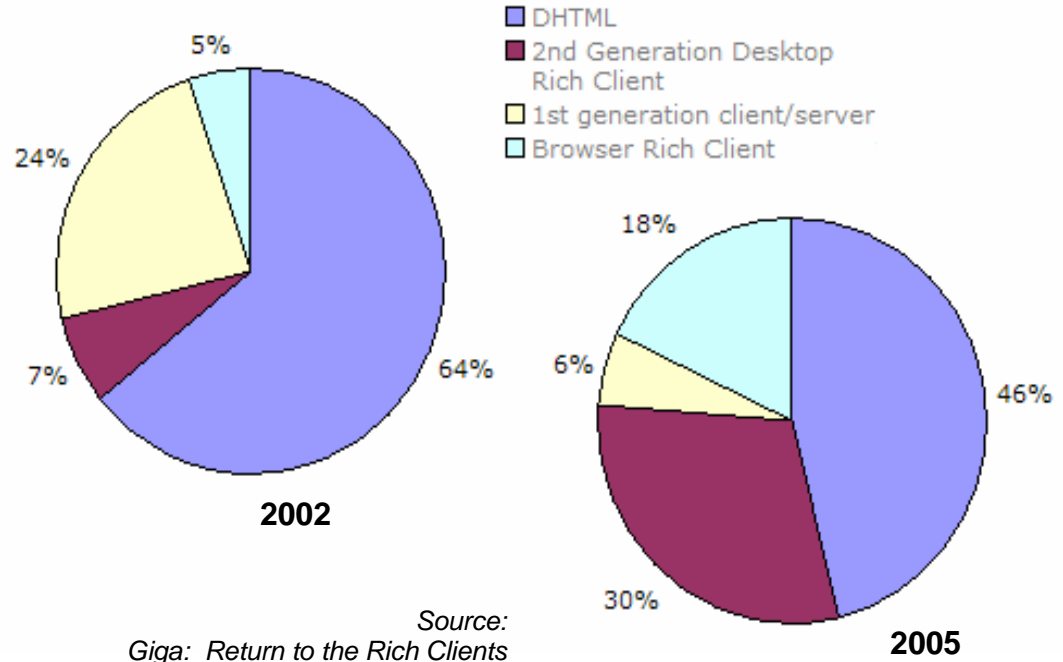
- Better (faster and consistent) response time on the Web based UI
- "Richer" UI than available in a Browser
- Ability to integrate existing "Client based" applications into their web applications
- Local business process execution (enabling mobility if needed)

Gartner

"...Growing trend back from browser-based thin clients to something in the middle – richer, smart clients..."



"48% of 2005 enterprise desktops deployed will be rich clients..."



Style 1: Self-built BTT-based Implementation

- Used by banks that either:
 - Need a low-cost entry point and simplified programming model
 - OR
 - Want to build a proprietary system for competitive advantage but do not wish to start from scratch.

- Suitable for either:
 - Enterprises with limited technical skills and do not wish to differentiate themselves in terms of multi-channel banking application design and capabilities
 - OR
 - Enterprises with extensive technical skills but need a framework to accelerate development and help insure a standardized development approach across channels and products.



Style 2: ISV Partner Lead Development

- Used by Banks that want a low-cost ISV solution
- Minimal development is required – just customization of vendor solution
- BTT provides a solid base on which the ISV can build a solution with minimal investment in development and ongoing support
- International Integrated System, Inc. (IISI)
 - IISI has a family of products called “iBranch” that integrates with BTT
 - IISI sold “iBranch” to banks in Taiwan and ASEAN like United Overseas Bank.

IISI 宏瞻資訊



Style 3: IBM-built Custom Implementation

- Used by banks that want a proprietary system for competitive advantage but do not have extensive development skills available to dedicate to the project
- IBM Service teams build a multichannel solution based upon BTT
- Once the system is built the customer can take over maintenance and incremental enhancements to the system
- Provides a solid technology base and helps insure a consistent approach across channels and products
- Infuses IBM technical skills into the enterprise.



BTT Customer Base

Over **50** BTT customers in EMEA, AP and Latin America



धन्यवाद

Hindi

多謝

Traditional Chinese

ขอบพระคุณ

Thai

Спасибо

Russian

Gracias

Spanish

شكراً

Arabic

Thank You

Obrigado

Brazilian Portuguese

Grazie

Italian

Danke

German

Merci

French

நன்றி

Tamil

多谢

Simplified Chinese

감사합니다

Korean

ありがとうございました

Japanese

Backup / Reference: Flow and Control Concepts

■ Channel Interaction Orchestration

- **Screenflow:** A lightweight Web / rich client tier control mechanism (usually a finite-state-machine) that guides the user from screen to screen. States and flows are encoded in XML.
- **Channel Application Microflow:** A lightweight Web / rich client tier control mechanism that provides a structured way to organize channel application operations such as screen flows, logging, reusable channel specific logic, invocation of business processes, and invoking back-end services. Tooling is specific to and integrated with the channel application platform. Flows are visually designed and encoded in XML.

■ Business Process Automation

- **Macroflow:** Long-running process or process involving human tasks to be performed by multiple people. Encoded in BPEL as a linear process or Business State Machines

■ Enterprise Application Integration

- **Service Composition:** The creation of a course-grained service from a number of finer-grained services and simple flow logic. Usually created via SCA components.
- **Service Orchestration:** Invocation of multiple services in the context of a microflow or macroflow execution. A flow or state machine may be used as the control construct to create a composite service from elemental services.
- **Routing and Transformation:** Routing of a service request to a service provider at runtime according to pre-determined rules and the transformation of the service name, number and type of parameters, and data structures as needed so as to insulate service consumers from service providers.
- **Dynamic Service Selection:** Determination of how to resolve a service binding at runtime.