

# Presentation-Layer Web Services

Ronald Schmelzer  
Senior Analyst  
ZapThink, LLC

zapthink

Copyright © 2001, ZapThink, LLC



## The Role of the User

- Human Interactivity
  - *Putting a face on system-to-system interactions*
  - *Making use of "wetware" (the brain)*
- Involving the Human in Business Processes
  - *Some processes cannot be fully automated*
  - *Human "checks and balances"*
- Presenting Consolidated Information and Access
  - *Providing a single view of multiple systems*
  - *Optimizing productivity and efficiency*
- Business Intelligence
  - *Giving the business the power to make decisions*
  - *The real face of "management"*

**Comes down to two parts: Interface and Workflow**



zapthink

## Interface Requirements

- A Single View of Data
  - Putting all the “right” information on the screen
  - Avoiding “Swivel-Chair” integration
- A Rich User Experience
  - Dynamic information
  - Interfaces built for productivity
  - Multi-Channel “Publishing”
    - The Connected Web
    - The Disconnected Web
    - Mobile Users
    - Rich Clients
- Making it all happen at “real-time”

Copyright © 2001, ZapThink, LLC



zapthink

## Workflow Requirements

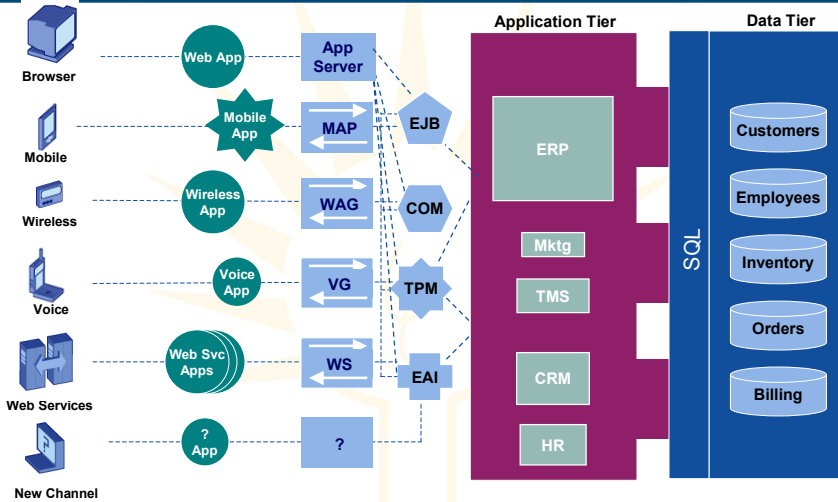
- Involving the human in business processes
  - Workflow: Approvals and Input
  - Management and Auditing
  - Introducing Process Change
- Cohesive identity management:
  - Single Sign-on
  - Enterprise Policies

Copyright © 2001, ZapThink, LLC



# The Interface Challenge

zapthink



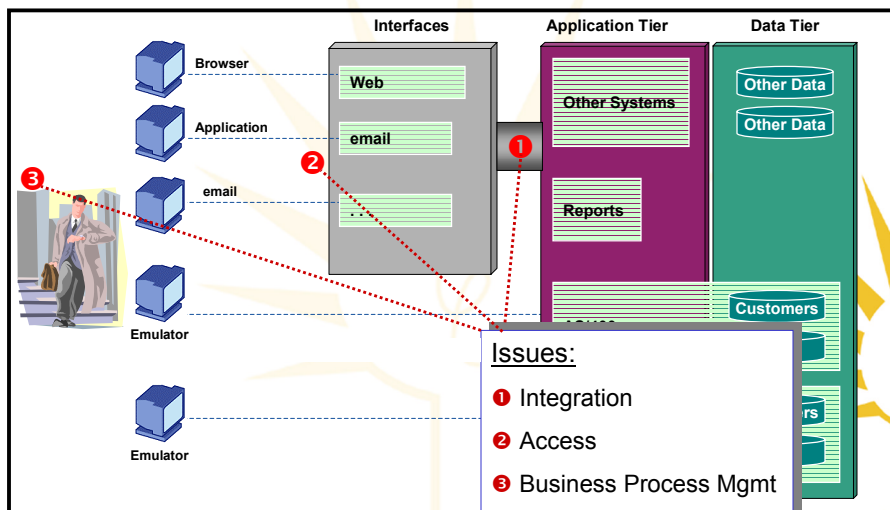
Copyright © 2001, ZapThink, LLC

Source: Cysive



# Current Interface Solutions

zapthink



Source: Cysive

Copyright © 2001, ZapThink, LLC



zapthink

## Separating the Presentation Tier

- Why?
  - Distribute user load
  - Allow for multi-channel publishing
  - Introduce new ways to access old systems
- Approaches:
  - Server-side Transformation
  - Portals
  - Generic Rich Clients
  - Embedding Web Services in Applications

Copyright © 2001, ZapThink, LLC



zapthink

## Portals

- The new Corporate Information “Desktop”
  - Remember the Common Desktop Environment (CDE)?
  - For distributed employees, customers, partners
  - Simplifying User Access
- Challenges
  - Personalization / Role-based Access
  - User Interactivity (interface + performance issues)
  - Data and Application Aggregation

Copyright © 2001, ZapThink, LLC



## Portals and Processes

- Where's the Service-Orientation?
  - Portals are interfaces to processes
  - Processes are recursive compositions of services
  - Portals composed of processes can be *loosely coupled, asynchronous, and coarse-grained*
- A Portal is an Interface Applied to a Process
  - Portals can be composed of Portals!
  - Portals can have multiple interfaces
  - Portals can involve automated and human-involved workflow

Copyright © 2001, ZapThink, LLC



## Portlets

- A portlet is a reusable piece of a Portal
- In many ways, a Portlet is really a user view of a subprocess
- However, portlets are typically *proprietary* per each portal platform
- Efforts to standardize portlet interaction:
  - Web Services for Remote Portals (Portlets) (WSRP) – Exchanging Portlets
  - Web Services for Interactive Applications (WSIA)- How to present UI for a Web Service

Copyright © 2001, ZapThink, LLC



zapthink

## Enterprise Applications and Portals

- If...
  - Enterprise Applications are increasingly exposing functionality as Services
  - Process is used to tie together enterprise application functionality to accomplish business requirements...
  - Portals are the interface to processes...
- Then...
  - Why should Enterprise Applications have their own interface?

Copyright © 2001, ZapThink, LLC



zapthink

## Server-side Transformation

- When portals aren't the answer, need a way to "transform" applications into interfaces
- Transformation technologies
  - XSLT
  - XQuery
  - Service-Oriented Process tools?
  - Service-Oriented Integration tools?
- Commercial Products
  - Cysive
  - Sarvega
  - DataPower

Copyright © 2001, ZapThink, LLC



## The Problem with Web Applications

- Web is a request/response model
  - Can't handle large amounts of data
  - Slow for dynamically updating information
- Does not meet user interactivity expectations
- Does not allow information to be merged with desktop information
- Asynchronous interactions not handled

... **Server-based presentation layers are not meant for robust user interaction**

Copyright © 2001, ZapThink, LLC



## Rich Clients

- Interact with users how they interact with desktop applications
  - Almost 20 years of GUI experience!
  - Just 6 years of Web interface experience
- Core capabilities
  - Windowing
  - Drag and Drop
  - Dynamic and instant response
  - Local capabilities (import, export, save, print, etc.)
  - Offline capabilities

Copyright © 2001, ZapThink, LLC



## Java-based Clients

- Java as a ubiquitous client VM
  - Applets in Portals
  - Client-side Applets
  - Java-based devices
- Commercial Java-based Rich Clients
  - Altio
- Java-like:
  - Curl

Copyright © 2001, ZapThink, LLC



## XML-based Clients

- Idea: Use XML processing power on servers and clients to handle presentation layer
  - XSLT → HTML
  - XML + Javascript + DHTML
- Commercial XML-based solutions
  - NexaWeb
  - Versalent
  - XMLSP

Copyright © 2001, ZapThink, LLC





## Build your own Client!

- Visual Basic .NET
  - VB has been used for years... successfully
  - VB.NET is a first-class Web Services consumer *and* producer
  - C# isn't that bad!
- PowerBuilder
- Borland Delphi

Copyright © 2001, ZapThink, LLC



## Microsoft Office as Client

- There's more Office in use than Internet Explorer
- The Web Services-enabled Office Suite:
  - Word: Content creation, aggregation, and publishing
  - Excel: Data aggregation and analysis
  - InfoPath: Intelligent, dynamic forms
- For asynchronous, coarse-grained interaction with Web Services by office users, Office might be the best interface!

Copyright © 2001, ZapThink, LLC



zapthink

## Requirements of Disconnected Clients

- Many business processes are not synchronous in nature... and neither are users
- Allow for mobile users
- Allow for service interruptions
- Allow for continuation of a business process on a different client / device / interface
- Solution
  - IONA Mobile Orchestrator?

**Must be asynchronous, process-driven, and presentation independent!**

Copyright © 2001, ZapThink, LLC



zapthink

## Mobile Device Interfaces

- Characteristics of a mobile device:
  - Limited bandwidth
  - Intermittent connectivity
  - Limited user interface and display
  - Different usage model: quick, short task-based interactions vs. long, business-oriented interactions
- Idea:
  - Use the mobile interface for executing asynchronous tasks
  - Don't think of a mobile interface as a small Web browser! It's its own rich client

Copyright © 2001, ZapThink, LLC



## Embedded Devices

- Devices that do a specialized task, but may participate in an overall process
  - Factory floor automation
  - Medical devices
  - Washing machines
- The interface is hidden
- The interface may not be electronic
  - Factory assembly devices
  - Heart monitor display
  - Status of Wash
- But, needs to be integrated with the rest of the system

Copyright © 2001, ZapThink, LLC



## Conclusion: Adding the Human

- The interface is required to make services useful
- Leverage process as the way to expose interfaces
- Don't let the interface drive the service definition
- Make interfaces appropriate to the device, but decouple it from the service

Copyright © 2001, ZapThink, LLC



# Questions & Discussion

Copyright © 2001, ZapThink, LLC