

Mobility and SOA: A Technology Perspective

Ronald Schmelzer
Senior Analyst
ZapThink, LLC

Take Credit Code: SCOMOBT
(1 credit)

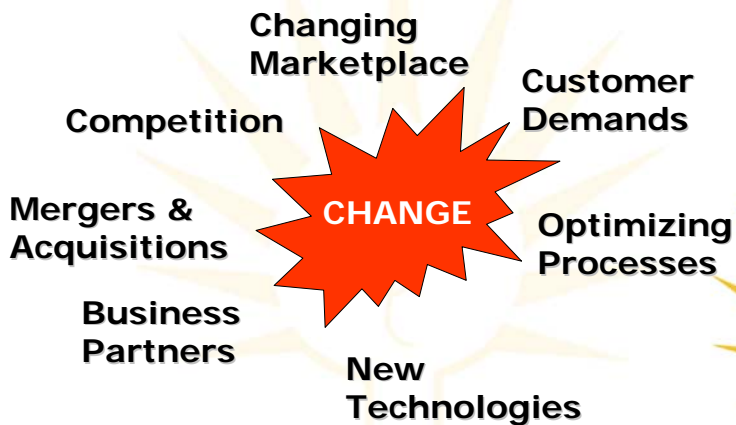
zapthink

Copyright © 2001, ZapThink, LLC



zapthink

Business Constant: Change



A Business is Never STATIC

Copyright © 2001, ZapThink, LLC



zapthink

Level Set – What is SOA?

- SOA is *architecture* – a set of best practices for the organization and use of IT
- Abstracts software functionality as loosely-coupled, business-oriented *Services*
- Services can be composed into *business processes* (which are also *Services*) in a declarative manner



Copyright © 2001, ZapThink, LLC



zapthink

The Benefits of SOA...

- ✓ Reduced cost of integration
- ✓ Improved value from legacy applications
- ✓ Reuse leading to reduced redundancy
- ✓ Greater visibility for governance & compliance
- ✓ Increased reuse of software assets
- ✓ Business agility...
 - Respond quickly & efficiently to business change
 - Leverage change for competitive advantage

Copyright © 2001, ZapThink, LLC



SOA != Web Services

Architecture



Implementation



The Secret Sauce: Loose Coupling

- Power of Abstraction
- Power of Open Standards
- How to Think Loosely Coupled



How loose is YOUR coupling?

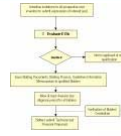


zapthink

Is Interoperability Important?



Service Contract



- Distributed Computing Yesterday: Portability
- Distributed Computing Today: Interoperability
 - Keep Investments in Legacy Systems
 - Allow for Best of Breed
 - Interop only important when “crossing boundaries”

Companies want choices, but resist making them

Copyright © 2001, ZapThink, LLC



zapthink

Service Orientation: Light at the End of the Tunnel



- Service Orientation: a *business* approach
- What's a Service anyways?
- Services + Loose Coupling = Agility
- ~~Integration~~ Process

Copyright © 2001, ZapThink, LLC



zapthink

SOA shifts the way we think

Traditional Distributed Computing	Service Oriented Architecture
Designed to last	Designed to change
Tightly Coupled	Loosely Coupled, Agile and Adaptive
Integrate Silos	Compose Services
Code Oriented	Metadata Oriented
Long development cycle	Interactive and iterative development
Cost centered	Business centered
Middleware makes it work	Architecture makes it work
Favors Homogeneous Technology	Favors Heterogeneous Technology

Copyright © 2001, ZapThink, LLC



zapthink

Characteristics of Services

- Services are *loosely coupled* – making a change to a service provider does not mandate changing any service consumers.
- Business processes are composed of services, and are in turn exposed as services.
- Services are *policy-driven* – business users can change how a service behaves.
- Systems are *inherently integrated* by virtue of composable services – not through layers of middleware.

Copyright © 2001, ZapThink, LLC



Characteristics of Services

- Services *leverage* legacy systems – SOA does not mandate replacement of runtime infrastructure.
- In SOA, *metadata* controls how the system behaves instead of code – business logic trumps application logic.
- In SOA, it's the *contracted interface* that matters, not the underlying runtime environment.

Copyright © 2001, ZapThink, LLC



Role of Composite Applications

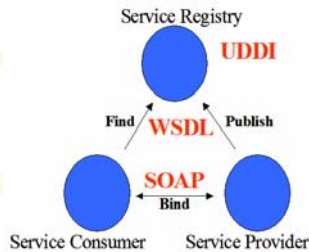
- *Architecture* underlies composition of Services
- Legacy assets key part of Service-oriented composite apps
- Benefit: combine old legacy processes with new composite business logic



Copyright © 2001, ZapThink, LLC



Ultimate Web Services Consumer?

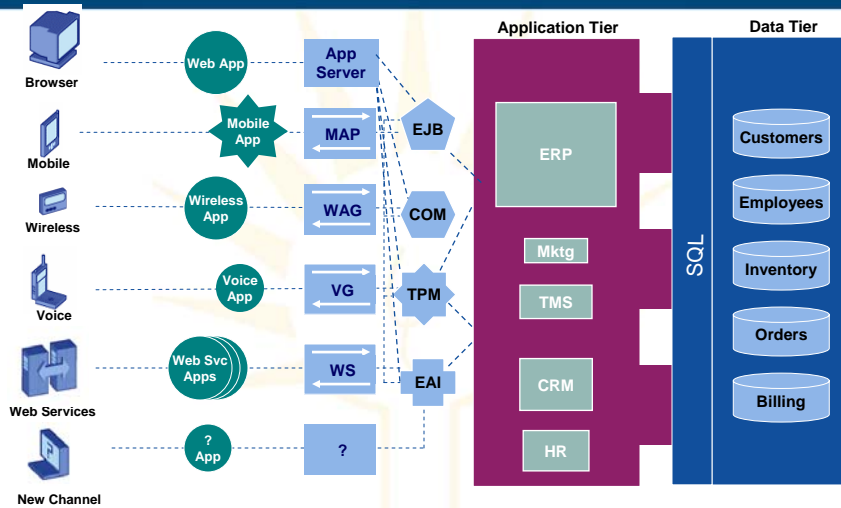


- Client-Server: The bottom of the SOA triangle
- Services are distributed, so why should they be consumed by web apps?
- Hello composition, bye-bye portals

Copyright © 2001, ZapThink, LLC



The Interface Challenge



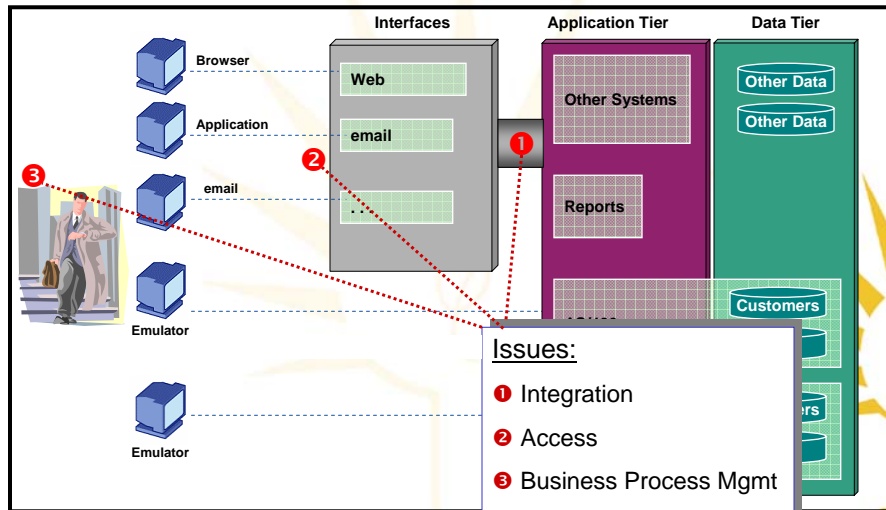
Copyright © 2001, ZapThink, LLC

Source: Cysive



zapthink

Current Interface Solutions



Source: Cysive

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



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



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



Mobility as a *Service*

- Information Anywhere
 - The workforce is mobile
 - Data is mobile
 - The problem is when you combine both...
 - Mobility for **consuming** Services
- Collaboration through Composition
 - Mobility enables dynamic composition
 - *The Mashup*
- Providing mobility Services to the rest of the enterprise
 - Mobility **providing** Services

Copyright © 2001, ZapThink, LLC



Mobility Services: Capabilities

- Rich User Interface
 - Capable of GUI, window-based interaction
- Two-way, synchronous and asynchronous messaging
 - Request/Response, state-less modes aren't enough
- Integrate local and remote sources of data and logic
 - Give user control over data
- Disconnected and connected modes
 - Provide for long-running transactions
- Loosely coupled presentation and business logic
 - Decouple the UI from the Service
- Client access without install
 - Realize low TCO
- Multiple web browsers and OS support

Copyright © 2001, ZapThink, LLC



Mobility Approach: Flash/Plug-in Clients

- Can support wide range of user interaction requirements
- Not limited by browser sandbox and technology issues
- Easy to guarantee consistent, predictable user interaction
- End users must either have plug-in already installed or the ability to install client
- Must maintain end-user plug-in versioning
- Development is specific to plug-in language
- Not easy to support in all devices

Flash: Not Just for Designers Anymore!

Copyright © 2001, ZapThink, LLC



Mobility Approach: Java / VM

- Low TCO
- Wide range of user interaction capabilities
- Supports existing development skills
- Runtime present on most machines
- Might be limited by browser sandbox and security issues.
- Java on the client fading?
- ActiveX on the client already dead?

Java / ActiveX on Client: Dead or Dying?

Copyright © 2001, ZapThink, LLC



Mobility Approach: Ajax

- Ajax: Once was an acronym...
 - Asynchronous Javascript and XML... not just a pattern
- Limited by capabilities of widely accepted browser standards
- Requires thin technology layer at both server and client-side
- Can leverage existing development knowledge
- In theory, can support widest range of devices and systems
- Requires adherence to Browser technologies, policies

Ajax, Ajax, Everywhere!

Copyright © 2001, ZapThink, LLC



zapthink

Interoperability vs. Portability

- Traditional RPC-based distributed computing based upon *portability*
 - Object serialization
 - Remote method invocation
 - Simulate one local computer
- SOA based upon *interoperability*
 - Code remains in place, known only by contracts
 - Computers interact via messages

**“Write once, run anywhere” gives way to
“write once, access anywhere”**

Copyright © 2001, ZapThink, LLC



zapthink

Messages Correspond to Events

- *Ordinary event* – something that happens in the real world
- *Ordinary business event* – a meaningful change in the state of the enterprise or of something relevant to the enterprise
 - Customer order, the arrival of a shipment, etc.
- *Software event* – a record of an ordinary event in software.
 - Data that describe the ordinary event in the form of a message



Copyright © 2001, ZapThink, LLC



Request/Reply

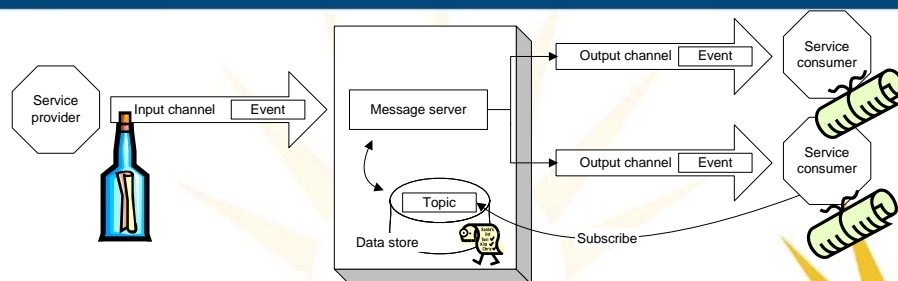


- Synchronous interaction made up of two asynchronous interactions
- This is the client/server invocation style – only one of several possible

Copyright © 2001, ZapThink, LLC



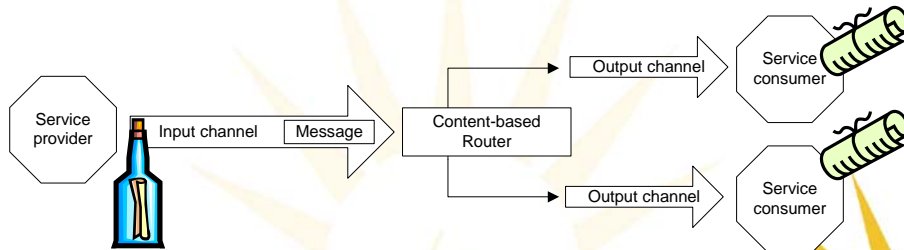
Publish/Subscribe



- Consumers subscribe to topic
- Provider creates event
- Message server duplicates and routes events depending on subscription info

Copyright © 2001, ZapThink, LLC

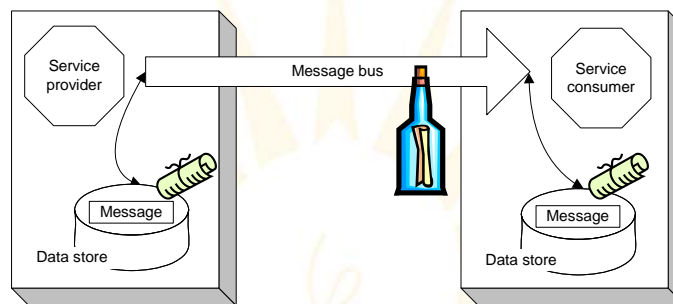
Routed Events



- Role of integration broker – transforms, routes messages
- Looks inside message
- ESBs offer this functionality

Copyright © 2001, ZapThink, LLC

Reliable Messaging

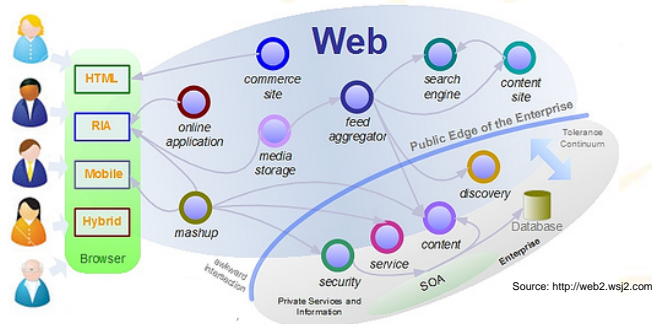


- Guarantees message delivery
- Can also guarantee messages sent only once, delivered in correct order

Copyright © 2001, ZapThink, LLC



The Rise of the Mashup



- Mashup = a flexible composition of Services within a rich user interface environment
- In essence, a Mashup is a SOBA interface

Copyright © 2001, ZapThink, LLC



Empower Business Users?



- The *mashup*: leveraging the Web to compose Services into *ad hoc* apps
- Without management and governance, will never be appropriate in an enterprise environment
- How to empower users in the spirit of the mashup, but maintain necessary control?

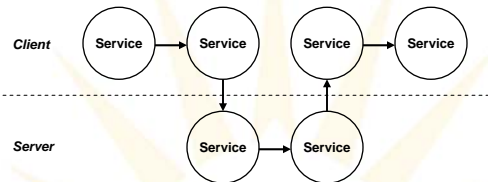
Governance the key to the "Enterprise Mashup"

Copyright © 2001, ZapThink, LLC



zapthink

Mashups Abstract "Client" & "Server"



- Clients are distributed, diverse, occasionally disconnected, hard to share
- Servers are centralized, easily governed, easy to share
- Mashups combine *location-independent* Services, abstracting role of client and server

Copyright © 2001, ZapThink, LLC



zapthink

The Key is the Service

- Standards for integrating the interface are one thing...
- Standards for defining Mobility Services are another
- An exercise: What does the *business* want?

Mobility and SOA: Taking an architectural approach to the needs of the mobile worker

Copyright © 2001, ZapThink, LLC

Thanks!



Ronald Schmelzer
rschmelzer@zapthink.com

Copyright © 2001, ZapThink, LLC

zapthink