



Fundamentals of XML, Web Services & SOAs for Financial Services

Ronald Schmelzer
Senior Analyst
ZapThink, LLC

Copyright © 2004, ZapThink, LLC

zapthink



zapthink

Introduction & Agenda

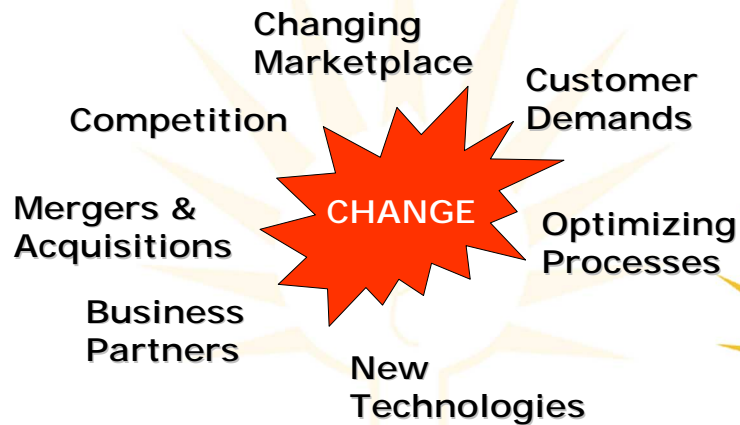
- 
- Implementing a Service-Oriented Architecture is a *journey*
 - Moving from proprietary interfaces to standards-based ones is just the first step...
 - SOAs require a combination of security, management, integration, process, and architecture tools
 - What are the right steps to guarantee overall success?
 - How can you guarantee an ROI while reducing risk?

Copyright © 2004, ZapThink, LLC



zapthink

Business Constant: Change



*A Business is Never **STATIC***

Copyright © 2004, ZapThink, LLC



zapthink

IT: Fulfilling Business Requirements

Business Requirements

- Service Customers
- Manage Operations
- Increase Worker Productivity
- Communicate with market
- Ensure reliable and secure operations
- Develop new products and services
- Respond to new business drivers

IT Capabilities

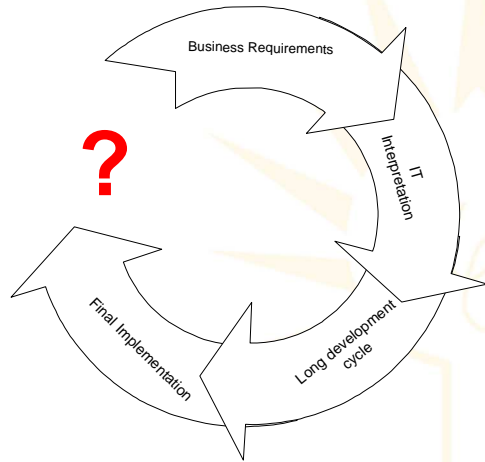
- Implement CRM Systems
- Implement ERP Systems
- Manage desktop environments
- Manage server environments
- Manage email systems and web sites
- Manage network and storage operations
- Develop applications

Copyright © 2004, ZapThink, LLC



zapthink

However, it rarely works that way...



- Requirements change
- Interpretations often inaccurate or limited
- Lengthy development cycles impervious to change
- Implementations "cast in concrete"

Result: IT that places limitations on Business

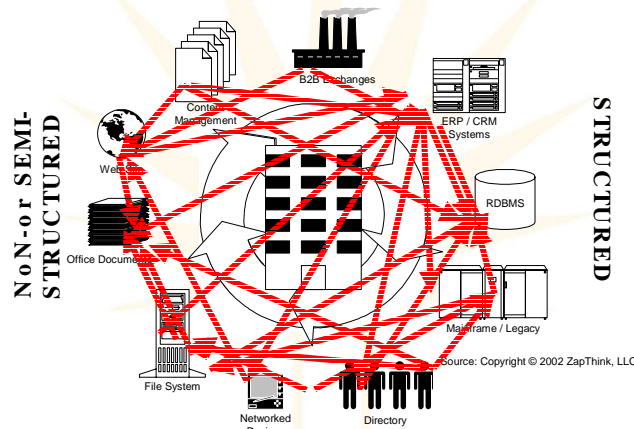
Copyright © 2004, ZapThink, LLC



zapthink

The Integration Challenge...

The N-squared Integration Problem:

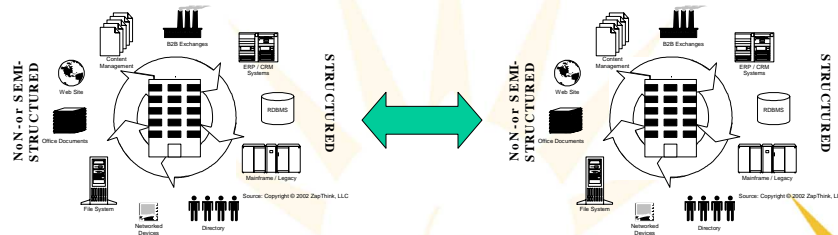


Copyright © 2004, ZapThink, LLC



zapthink

...Made Worse by External Integration



- *70% of IT development budgets are spent on integrating different systems – ZapThink*

Copyright © 2004, ZapThink, LLC



zapthink

Key to meeting Business Requirements: Effective Integration

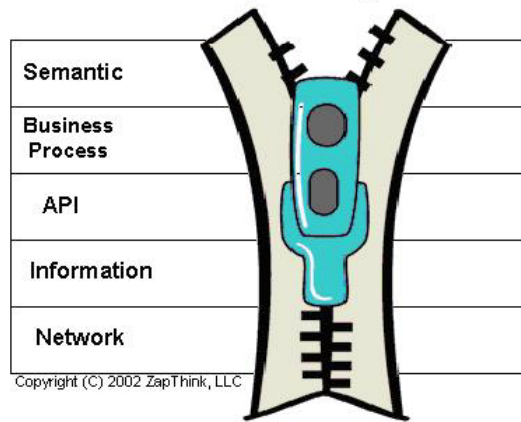
- ✓ Respond to Business Drivers
- ✓ Get a better understanding of customers and business operations
- ✓ Connect divisions, suppliers, partners, customers
- ✓ Lower cost of ownership for legacy applications
- ✓ Value-add existing applications
- ✓ Eliminate “islands of information”

Copyright © 2004, ZapThink, LLC



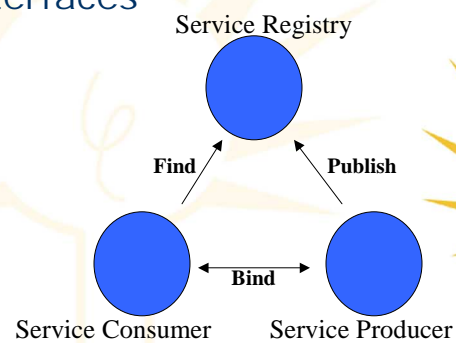
The Integration "Zipper"

The Integration "Zipper"



What is a Service-Oriented Architecture?

- Software functionality available via discoverable interfaces on the network





zapthink

Have We Been Here Before?

- Service-Oriented Architectures have been around a while
- CORBA (Common Object Request Broker Architecture) and DCOM (Microsoft Distributed Component Object Model) two familiar examples
- What's new this time?



CORBA®

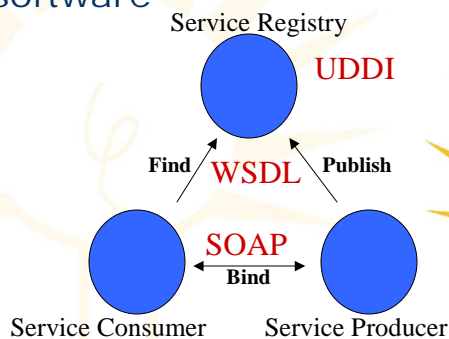
Copyright © 2004, ZapThink, LLC



zapthink

The Difference is Web Services

- *Standards-based* interfaces to software functionality



Copyright © 2004, ZapThink, LLC



zapthink

Web Services in the Present...

Web Services are in the *horseless carriage* phase

- Where new technology is applied in the patterns of the earlier technology
- Web Services are used to simplify integration



Copyright © 2004, ZapThink, LLC



zapthink

Web Services in the Future...

New approaches to software development, engineering, architecture, and management



Copyright © 2004, ZapThink, LLC



Service orientation...the next big thing? zapthink

Approach	Timeframe	Programming Model	Business Motivations
Mainframe timesharing	1960s -1980s	Procedural (COBOL)	Automated business
Client/server	1980s-1990s	Database (SQL) and fat client (PowerBuilder, Visual Basic)	Computing power on the desktop
n-Tier/Web	1990s-2000s	Object-oriented (Java, COM)	Internet/eBusiness
Service orientation	2000s	Service-oriented (SOAP, WSDL, UDDI)	Business agility

Source: Copyright © 2003 ZapThink, LLC

Copyright © 2004, ZapThink, LLC



Web Services are the Trees....

zapthink



Service Orientation is the Forest

Copyright © 2004, ZapThink, LLC



Web Services *Idées Fortes*: Loose Coupling

- Consumer and Producer controlled by different people
- Changing one doesn't break the other
- Build one without being aware of the other

Copyright © 2004, ZapThink, LLC



Web Services *Idées Fortes*: Asynchrony

- The Web is synchronous: click a button and wait for a response
- Web Services can also be asynchronous: allow for long-running processes

Copyright © 2004, ZapThink, LLC



Web Services *Idées Fortes*: Coarse Granularity

- Business-oriented requests and responses
- Blocks of information exchanged
- Encapsulate APIs into fine-grained, atomic Services and compose them into coarse-grained, business Services

Copyright © 2004, ZapThink, LLC



Financial Services = Information Industry

The Universe of Financial Services:

Capital Markets				Banking		Insurance		Real Estate		Accounting
Equities Trading	Fixed Income Trading	Currencies & Commodities	Investment Banking	Retail Banking	Commercial Banking	P & C	Life	Mortgage	Commercial Lending	Accounting

Source: Copyright © 2002, ZapThink, LLC

So, Just How Important are Financial Services?

"The Financial Services sector spent over \$195 Billion (US) in IT in 2001, with \$985 Million invested on XML technologies in 2002."

"Expenditures on XML technologies in the Financial Services sector will grow to over \$8.3 Billion by 2005."

Copyright © 2004, ZapThink, LLC



FSTC Example



The Financial Services Technology Consortium (FSTC) is a consortium of leading North American-based financial institutions, technology vendors, independent research organizations, and government agencies.

FSTC sponsors collaborative technology development-pilots, proofs-of-concept, tests, and demonstrations-supported by member financial institutions and technology companies. Its aim is to bring forward interoperable, open-standard technologies that provide critical infrastructures for the financial services industry.

Copyright © 2004, ZapThink, LLC



The FSTC Test...

- Goal: Provide guidelines on how web services can -- or cannot -- be used for complex and sensitive B2B cash management system integration
 - 5 month test. Participants included: Bank of America, JP Morgan Chase, Wachovia, NEC, and Sun
- Test comprised of several subtasks applying *today's* Web Services technology to multi-bank reporting services
- Produced a set of Web services, reference implementation, and next-generation "treasury portal" implementations
- Tackled the interoperability challenges of bank-to-bank cash management
- Web Services ability to provide interoperability, security, and reliability were evaluated

Copyright © 2004, ZapThink, LLC



FSTC Test Results...

- The Good
 - It mostly worked as hoped. End results: (1) a reusable architecture (2) Baseline Web Services (3) technical and business documentation for planning .NET and Java Implementations
- The Bad
 - Insufficient Standards development for:
 - "End-to-end security and reliability" to meet SLAs
 - SOAP attachments
 - Interface-level interoperability
- The Ugly
 - "Having multiple open source and commercial technologies use a common Web services description language (WSDL) file is problematic as the automatic code generators carry numerous bugs."

Copyright © 2004, ZapThink, LLC



FSTC Conclusions

- The Conclusion:
 - "Our members are pleased with insights from the initial phase of the project and plan to further explore the use of Web services in financial services applications"
- Issues for Vendors to solve:
 - Implementations were made to conform with some re-programming of the generated code
 - Hand tuning auto-generated code is not considered to be a default Best Practice
 - Security interoperability is very immature in open source code and commercial offerings
- ***"Financial institutions should remain optimistic about the benefits pledged by these technologies for partner, customer, and enterprise integration, and that the transition to Web services technologies requires a careful examination of the underlying service architecture required to provide manageable, secure, and reliable financial applications."***

For more info: www.fstc.org

Copyright © 2004, ZapThink, LLC



zapthink

Avoiding the Pitfalls of Web Services & SOAs

- Standards are still in flux, which means products are immature
- Large vendors have less flexibility, while small vendors have less stability
- “Web Services” does not equal “SOA”
- Only architects should do architecture (developers aren’t qualified)
- SOAs are NOT easy!

Copyright © 2004, ZapThink, LLC



zapthink

The Service-Oriented Enterprise

- IT resources are available on demand to businesses as Services
- The Service-oriented abstraction layer enables companies to run their operations and conduct business with each other in a dynamic and automated fashion
- Business drives IT, and agile IT enables agile businesses

Copyright © 2004, ZapThink, LLC

Take Credit for attending this presentation!



- Go to www.zapthink.com/credit and enter the code **XFFSINTR**.
- Download a digital copy of the presentation
- Sign up for our ZapFlash newsletter
- Get a ZapCredit good toward free research and ZapGear!

Copyright © 2004, ZapThink, LLC

zapthink