



SOA & XML Workshop General Session

Ron Schmelzer & Jason Bloomberg
Senior Analysts
ZapThink, LLC

Copyright © 2004, ZapThink, LLC

zapthink



Introduction & Agenda

- XML Primer
 - Introduction to Service-Oriented Architecture
 - SOA Case Studies
 - XML & SOA Market Trends
- 

Copyright © 2004, ZapThink, LLC



zapthink

eXtensible Markup Language (XML)

- Based on W3C Standards
 - Based on SGML, a 20 yr. Old+ standard
- Text-based format
- Metadata-enhanced, Self-Defining Data Structure
- Rule Enforcement: Schema and Validity
- Technology heading to XML
 - Infrastructure can deal with it
 - Standard Parsers
 - Standard Training

Copyright © 2004, ZapThink, LLC



zapthink

So, why will XML succeed?

- **What is often said...**
 - "XML is easier to read and create"
 - "XML can use comparably cheap interoperability software"
 - "XML can be exchanged over the Internet"
 - "Agreement on a Schema can enforce interoperability"
 - "We can build processes based on the DOM"
 - "Remove Process from Content"
 - "No need for integration middleware?"
- **Even better arguments...**
 - Finally, for the small and medium sized guys
 - Lead by strong standards organizations
 - Training
 - Company attitudes
 - Security
 - Proliferation

Copyright © 2004, ZapThink, LLC



zapthink

Two Sides of the Data Coin



- Is Data Message or Document?
Hint: Shelf-life

Copyright © 2004, ZapThink, LLC



zapthink

The Changing World of Data



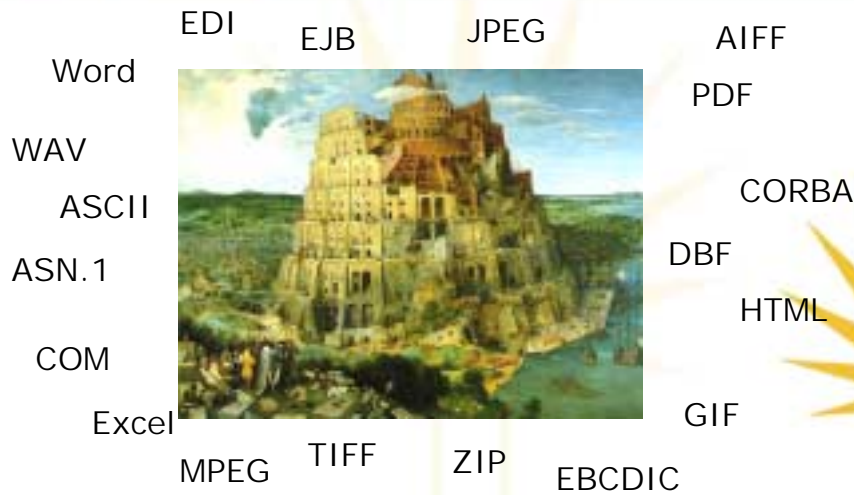
- Proliferation of Content
- Increased Integration of Applications
- Automation of Paper-based Systems
- Shift to Customer-Centric KM Model

**BIG CHANGES
AHEAD**

Copyright © 2004, ZapThink, LLC



The Data Tower of Babel



Copyright © 2004, ZapThink, LLC



XML to the Rescue?

- Markup languages are *tailor-made* for content:
 - SGML
 - HTML
 - XML... as an evolution of SGML
- XML can represent *documents* as well as *messages*
- XML separates content *representation* from content *presentation*
- But... you need metadata to make it all work!

Copyright © 2004, ZapThink, LLC



Introduction to SOA



Copyright © 2004, ZapThink, LLC



Business Constant: Change



A Business is Never STATIC

Copyright © 2004, ZapThink, LLC



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

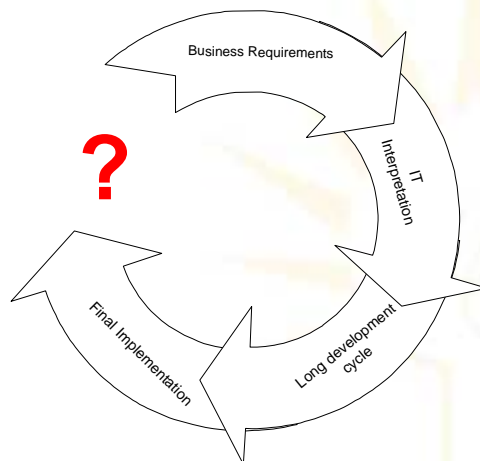
IT Capabilities

- 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



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

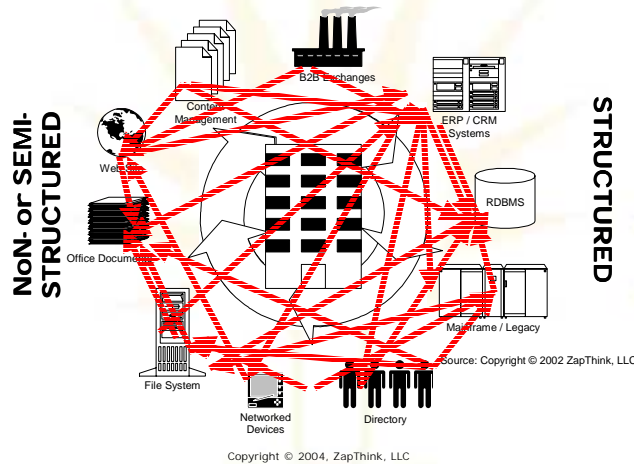
Copyright © 2004, ZapThink, LLC



zapthink

The Integration Challenge...

The N-squared Integration Problem:



zapthink

Integration Approaches of Yesterday

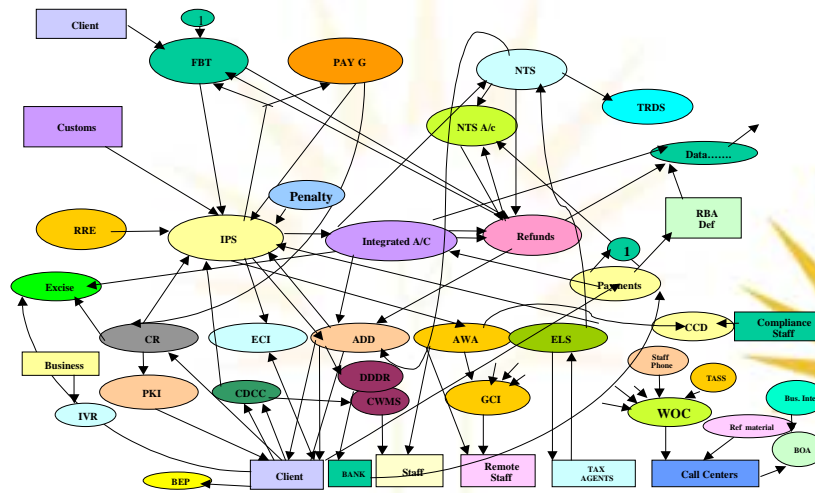
- Custom Integration: Coding to Interfaces
 - APIs: COM, Java, COBOL, Assembly?
 - Distributed Computing?: DCOM, CORBA
 - Screen-Scraping and Emulation (3270 and HTML)
 - Message-Queues
- EAI and B2Bi Middleware
 - Automating interface-level integration
 - Bus or hub-and-spoke architecture

Fundamentally *brittle* approaches to integration

Copyright © 2004, ZapThink, LLC



The "Rat's Nest" Architecture



Copyright © 2004, ZapThink, LLC



What is Service-Oriented Architecture?

- Access software via discoverable, *loosely coupled* Services
- Users can compose Services into business processes that are also Services



Copyright © 2004, ZapThink, LLC



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?



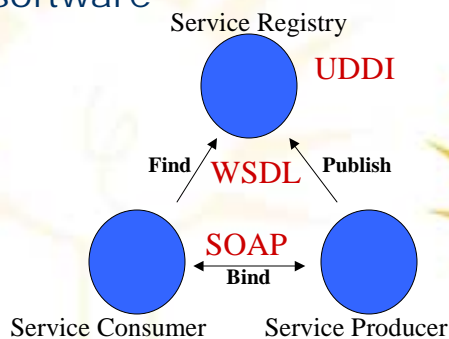
Copyright © 2004, ZapThink, LLC



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



zapthink

Web Services are the Trees....



Service Orientation is the Forest

Copyright © 2004, ZapThink, LLC

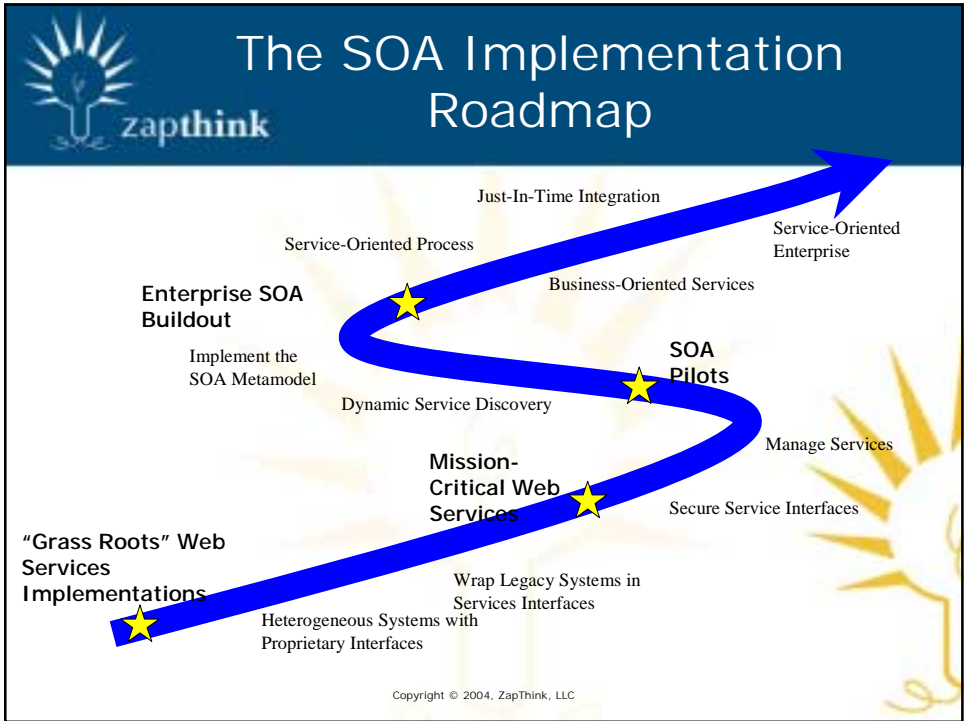
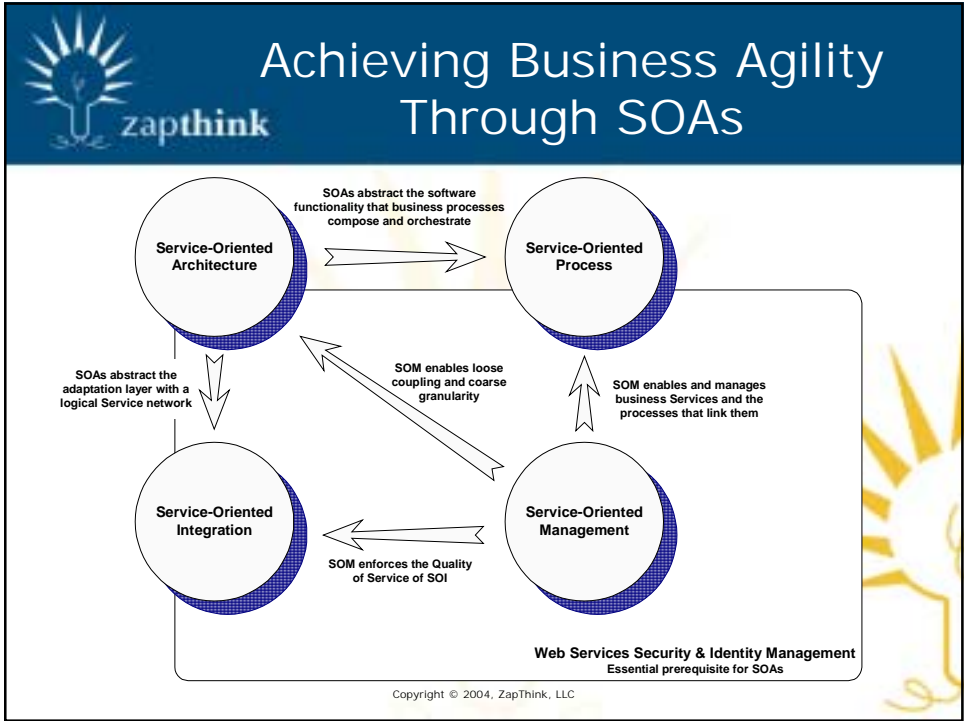


zapthink

Service orientation...the next big thing?

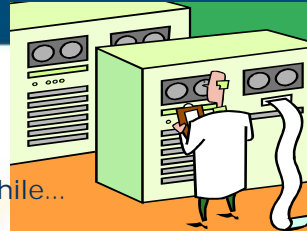
Approach	Timeframe	Programming Model	Business Motivations
Mainframe timesharing	1960s -1980s	Procedural (COBOL)	Automated business
Client/server	1980s-1990s	Database (SQL) and fat client (PowerBuilder, Visual Basic)	Computing power on the desktop
n-Tier/Web	1990s-2000s	Object-oriented (Java, COM)	Internet/eBusiness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

Copyright © 2004, ZapThink, LLC





Legacy Systems: The Brains Behind the Operation



- What is "legacy"?
 - Host-based systems...
 - Anything that's been around for a while...
 - Anything that's in use...
- Legacy systems enable a significant amount of mission-critical functionality
- Rip-and-Replace vs. Maintain-and-Extend

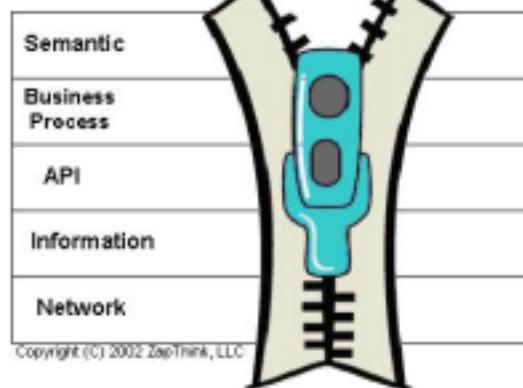
The key to extending functionality: abstracting the implementation – aka "Service Wrappers"

Copyright © 2004, ZapThink, LLC



The Integration "Zipper"

The Integration "Zipper"



Copyright © 2004, ZapThink, LLC



zapthink

“Enterprise Information Integration” vs. “Service-Oriented Information Integration”

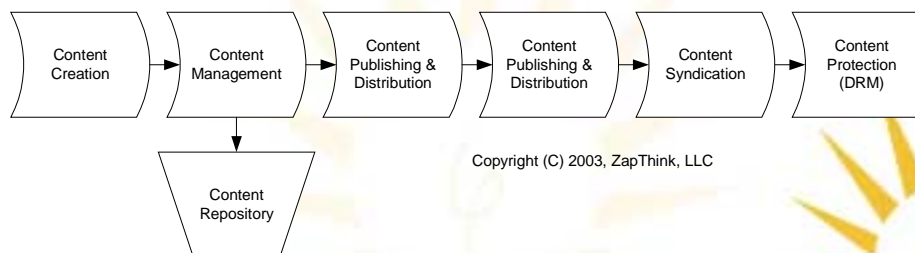
- “EII” – Enterprise Information Integration
 - The philosophy of a single view and access to distributed information.
 - Tightly-coupled integration → “So 1996.”
- “SOII” – Service-Oriented Information Integration
 - Loose Coupling
 - Information consumers and producers independently controlled
 - Changing one doesn’t break the other
 - Build one without being aware of the other – true distribution
 - Coarse Granularity
 - Business-oriented requests and responses
 - Blocks of information exchanged, rather than discrete API calls
 - Functionality and data composed into coarse-grained, business Services
 - Asynchrony
 - Coarse-grained information access MUST be asynchronous
 - Unfortunately most underlying systems are synchronous

Copyright © 2004, ZapThink, LLC



zapthink

The Content Lifecycle

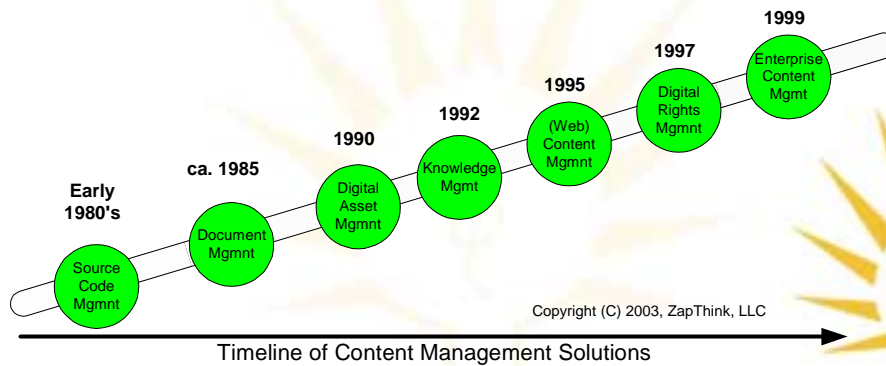


Copyright (C) 2003, ZapThink, LLC

Copyright © 2004, ZapThink, LLC



The Evolution of Content Management



Copyright © 2004, ZapThink, LLC

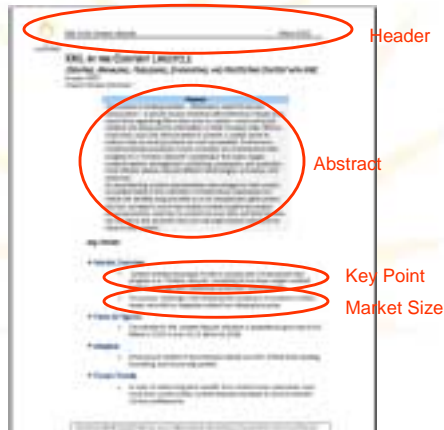


The XML-enabled Content Lifecycle

- Content Creation
 - Markup-based editors
 - Content conversion from unstructured to structured
- Content Management
 - XML-based, component-oriented content systems
 - WebDAV
- Content Storage
 - Native XML Data Stores
- Content Publishing
 - XML-based transformation
 - A myriad of digital formats
- Content Syndication
 - RSS and other news feeds
- Content Protection
 - Digital Rights Management

Copyright © 2004, ZapThink, LLC

An example...



Copyright © 2004, ZapThink, LLC

The ROI of Service-Oriented Content

- **Cost Savings:** Content reuse
- **Cost Savings:** Efficient content search
- **Cost Savings:** Integrating islands of content
- **Revenue Enhancing:** Enabling content syndication
- **Cost Savings:** Increased content process efficiencies
- **Cost Savings:** Reduced human errors
- **Security:** Improved content security
- **Cost Savings:** Improved content accuracy and relevancy through the use of line-of-business content authors
- **Flexibility:** Improved agility through speedy content development and deliver

Copyright © 2004, ZapThink, LLC



zapthink

Two Ways to Implement SOA with Content

- Content Services
 - Key: expose critical content lifecycle functionality as services on the Web
 - Advantage: no more monolithic functionality
 - Disadvantage: the content is still monolithic
- Content *as* Services
 - Key: componentize content and encode with discoverable metadata
 - Advantage: Content can be treated like any other digital asset (data, business logic, etc.)
 - Challenge: need new tools, approaches, ways to metadata-enable content

Copyright © 2004, ZapThink, LLC



zapthink

Case Studies



Providence | Health System



Copyright © 2004, ZapThink, LLC



Case Study: The Hartford

- SO Business Application for insurance agents
- Services handle multiple versions of insurance forms
- SOA handles multiple versions of Services



Case Study: Providence Health System



- SOA with 30 composite Services (each with ~10 atomic Services)
- Variety of uses, including patient portal, lab results, aggregation of medical records
- Supports HL7, X12 (payor interactions)
- Less duplication of effort, better patient care, faster & more complete patient information

Copyright © 2004, ZapThink, LLC



zapthink

Case Study: Merrill Lynch

- Have over 23,000 live CICS (Customer Information Control System) applications
- Have thousands of systems, other applications, and person-years of custom development in place
- Wanted to leverage legacy investments across enterprise



Copyright © 2004, ZapThink, LLC



zapthink

Solution Attempts

- Realized that 90% of the cost of IT was infrastructure and integration
- 10% of the cost of IT was business logic
- Other attempts to provide better access to legacy data were too slow and brittle



Copyright © 2004, ZapThink, LLC



Solution Framework

- Established “XML for Merrill Lynch” initiative they call X4ML
 - Program analyzer
 - Directory
 - Monitoring & logging
 - Testing
- Expose CICS transactions as Web Services
- Reduce software licensing costs
- Migrate to Web hosting on Linux on IBM mainframes



Copyright © 2004, ZapThink, LLC



Implementing the Solution

- Build UDDI-based registry first
 - Key to centralizing Services
 - Wanted “single root”
- Use both J2EE and .NET technology
- Dealt with semantic issues resulting from centralized Services – hundreds of “getCustomerInformation” Services example



Copyright © 2004, ZapThink, LLC



Results

- Increased response times by 10 to 20 times
- Increased throughput by 10 times
- Reduced the percentage of cost for infrastructure and integration from 90% to 65%
- Reduced the time to get data off of mainframe from 3-4 months to a matter of minutes



Copyright © 2004, ZapThink, LLC



Case Study: DFAS

- US Government office: "Defense Finance & Accounting Service" in the Department of Defense (DOD)
- Very large, complex organization that handles accounting for Army, Navy, Air Force & Marines
- Process \$1 Billion dollars of payments per day



Copyright © 2004, ZapThink, LLC



Framing the Problem

- Bring together finances of DOD
- Reduce hardware costs by streamlining operations
- Achieve "semantic alignment" across organizations
- Collating and rolling up information in the face of enormous complexity

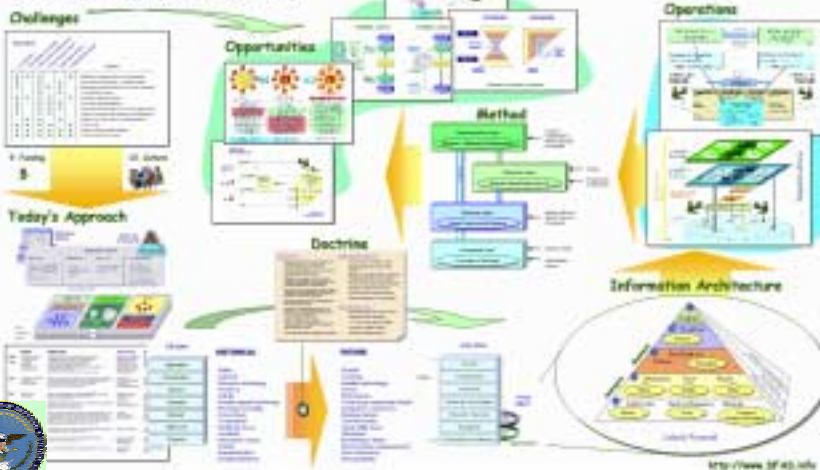


Copyright © 2004, ZapThink, LLC



Solution Attempts

Business-Centric Methodology For Enterprise Agility & Interoperability



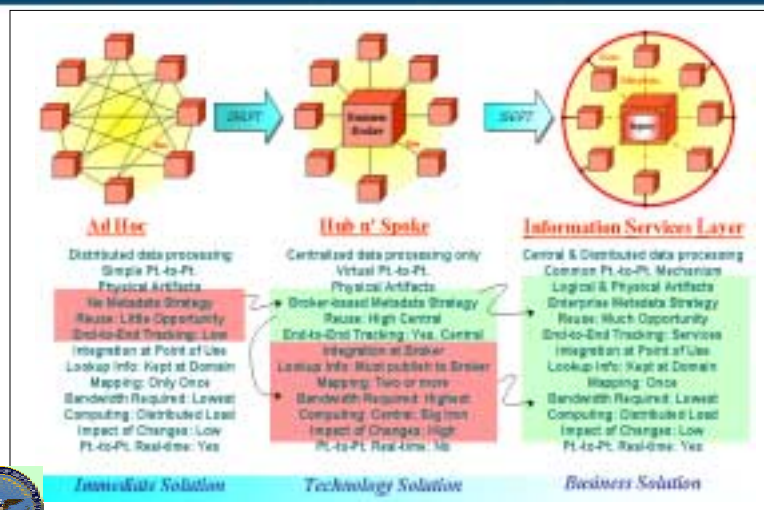
Solution Attempts

<u>Root Causes</u>						<u>Symptoms</u>
1,3. Semantics	4. Frameworks not complete	1. Take too long the steering wheel	4. One Size Doesn't Fit all	7. Incompatibility to Power	4. Both Parties participate	
●	●	●	●	●	●	Ineffective communication of requirements
●		●	●	●	●	Non-reliable information - Integrity/Quality
●		●	●	●	●	Extending individual efforts to common a painful
●	●	●	●	●	●	Coordinated processes
●		●	●	●	●	Inability to upgrade system
●		●	●	●	●	Don't have the information
●		●	●	●	●	Customer dissatisfaction due to not meeting needs
●		●	●	●	●	Unable to measure effectiveness of the Enterprise
●		●	●	●	●	Unable to go from vision to implementation
●		●	●	●	●	Scope-creep
●		●	●	●	●	Delay in system implementation
●		●	●	●	●	Cost overruns for a project.



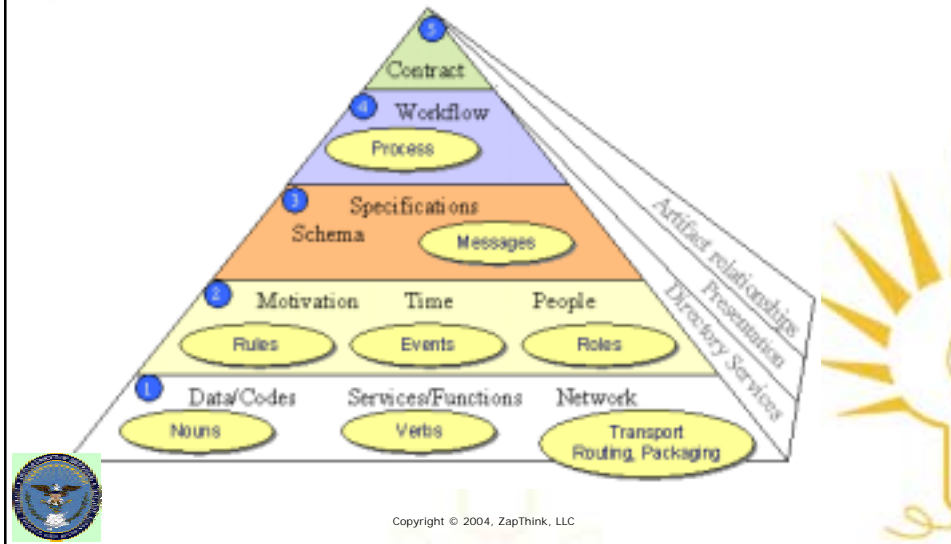
Copyright © 2004, ZapThink, LLC

Solution Attempts



Copyright © 2004, ZapThink, LLC

Solution Attempts



Solution Framework

- Develop a "Business-Centric Methodology" (BCM)
- Realize that infrastructure and culture each represent 50% of problem
- Move from OSI (network) model to a different concept of interoperability





zapthink

Implementing the Solution

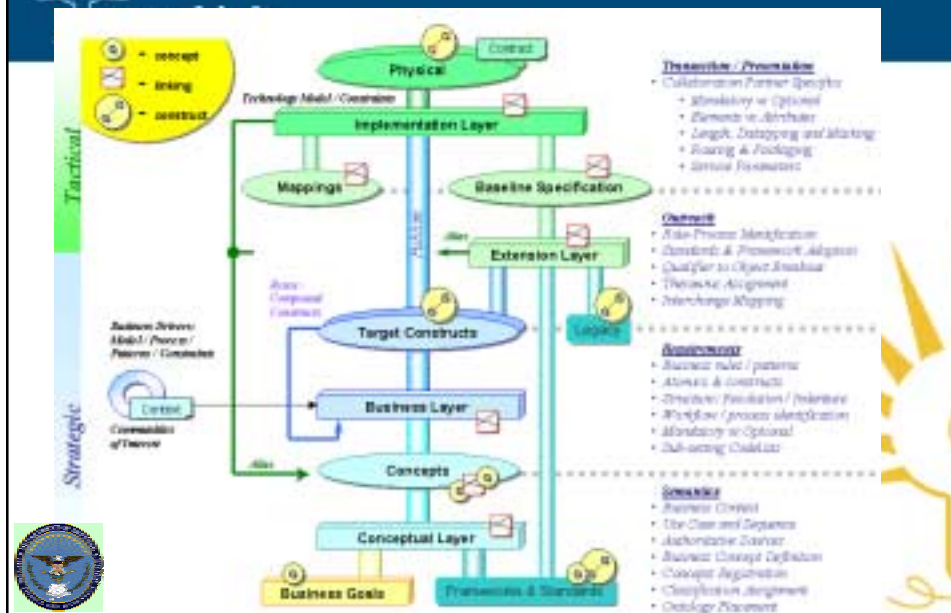
- Focus on information architecture as the key to managing slower-changing artifacts
- “Pragmatic interoperability” with the information architecture pyramid
- “Communities of interest” to filter correct information to the right people



Copyright © 2004, ZapThink, LLC



Results





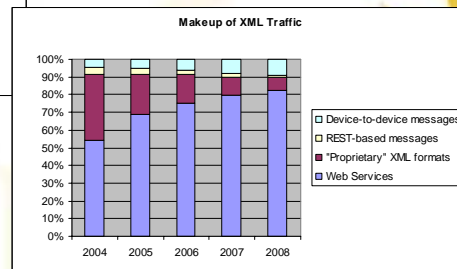
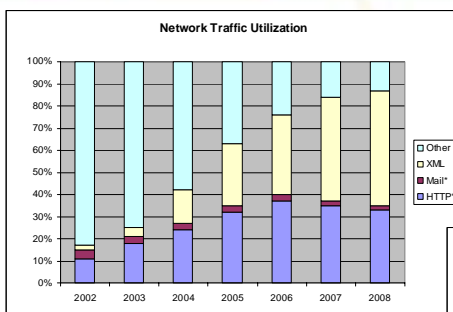
Market Trends



Copyright © 2004, ZapThink, LLC



Growth of XML on the Network

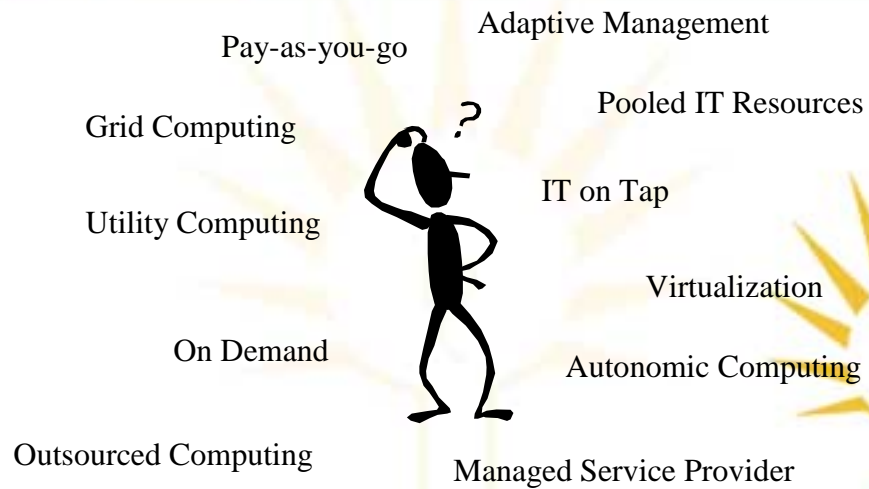


Copyright © 2004, ZapThink, LLC



zapthink

On Demand Computing



Copyright © 2004, ZapThink, LLC



zapthink

On Demand Computing

- Many different approaches to making IT a resource that is available as needed
- What's lacking from all of today's buzzwords is an organizing concept
- That concept is *Service Orientation*

Copyright © 2004, ZapThink, LLC



Unifying idea: *Virtualization*

- SOA → virtualization of application functionality
- Grid → virtualization of processing capability
- EII → virtualization of data
- SAN → virtualization of storage

- Service Orientation → virtualization of IT

Copyright © 2004, ZapThink, LLC

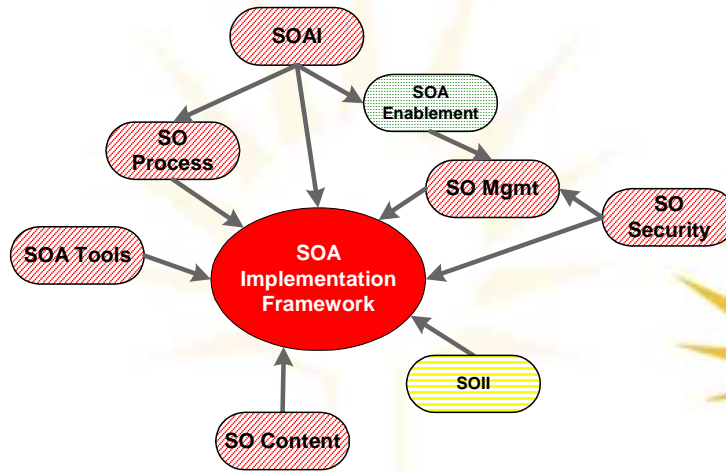


Introducing the SOAIF...

- The SOA implementation framework is a product or set of products that offer everything companies need to build, run & manage their SOA
- No vendor offers a complete SOAIF (yet)
- Leverages Service orientation for modularity
- Today's markets are converging on the SOAIF

Copyright © 2004, ZapThink, LLC

SOAIF Futures...



Copyright © 2004, ZapThink, LLC

The Service-Oriented Enterprise

- IT resources are available on demand to businesses as Services
- The SOA provides an *abstraction layer* that enables companies to conduct business with each other in a dynamic and automated fashion
- Business drives IT, and agile IT enables agile businesses



Copyright © 2004, ZapThink, LLC



ZapThink is an industry analysis firm focused exclusively on XML, Web Services, and Service-Oriented Architectures.



Ronald Schmelzer
rschmelzer@zapthink.com

Take Credit for attending this presentation!



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



Jason Bloomberg
jbloomberg@zapthink.com

Copyright © 2004, ZapThink, LLC

zapthink