



ZapThink Presentation for Energy Client: How do you Build a Service?

Jason Bloomberg and Ronald Schmelzer
Senior Analysts
ZapThink, LLC

Take Credit Code: ENBUILD

Copyright © 2005, ZapThink, LLC

zapthink



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

Copyright © 2005, ZapThink, LLC



Emerging Best Practices



- Architecture is a *discipline*
 - SOA mandates new practices and methods
- *Best Practice*: Iterative development
 - Start small, and let it grow
- *Best Practice*: Contract-first, top-down as well as bottom-up
- *Best Practice*: Implementing governance sooner than later

You can't get SOA from a software vendor!

Copyright © 2005, ZapThink, LLC



Top-Down & Bottom-Up Approaches

- Top-down only: have the plan, may not be able to execute
- Bottom-up only: build Services, may not be reusable
- SOA planning *must* be both
 - Develop the vision (but not the details) ahead of time
 - Service development should be iterative

Copyright © 2005, ZapThink, LLC



Process Mentality

- Developer Mind-set: "Bottom-Up"
 - Everything is a Service or an Interface
 - Goal: connect Services
 - Method: Use objects and App Servers
 - Problem: Too many things to connect!
- Business Mind-set: "Top-Down"
 - Everything is a Process
 - Goal: Run business efficiently: manage processes
 - Method: Use diagrams and flowcharts
 - Problem: How can you turn "shelf-ware" into software?

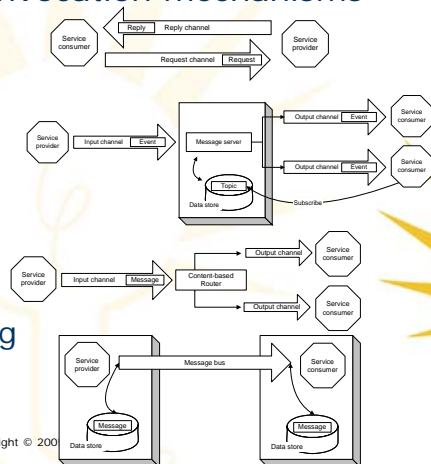
Copyright © 2005, ZapThink, LLC



Invocation Mechanisms in SOA

- SOA is more flexible than client/server – supports multiple invocation mechanisms

- Request/Reply
- Publish/Subscribe
- Routed Events
- Reliable Messaging

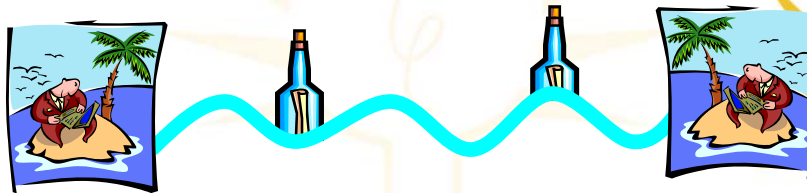


Copyright © 200



Event-Driven SOA

- Event-Driven SOA is an approach to distributed computing where events trigger asynchronous messages that are then sent between independent software components that need not have any information about each other.



Copyright © 2005, ZapThink, LLC



How to Identify Services

- Process decomposition
- Identification of redundancy
- Potential Service identification
- Service prioritization
- Iterative Service implementation

Copyright © 2005, ZapThink, LLC



Service Granularity

- The trick of building composable Services is building at the right level of granularity
- Challenges:
 - Engraining business logic into code
 - Decomposing legacy services that are not fine-grained enough
- Method
 - Top-down process decomposition, vs. bottom-up Service development

Copyright © 2005, ZapThink, LLC



SOA Tenet: Move from Code to Composition

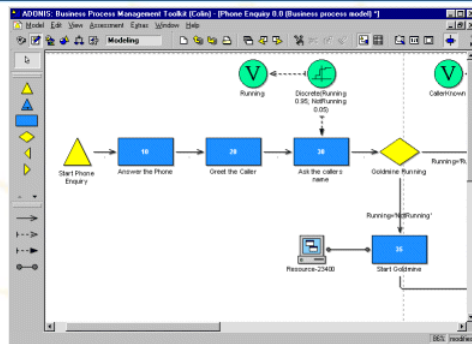
- Fundamental shift from code-centric business logic to metadata-centric
- It's not the Service that makes the application, but the composition and the policy
- Goal: users build applications out of Services



Copyright © 2005, ZapThink, LLC



Business Process and Integration



- Executing a business process accomplishes the goals of integration...
- However, hasn't worked in the past

Copyright © 2005, ZapThink, LLC



Service-Oriented Process

- Processes that are *coarse-grained*: composed of Services and *exposed* as Services
- Processes that are *loosely coupled*: a change to a process flow, activity, subprocess doesn't effect other processes
- Processes that are *asynchronous*
- Processes that are dynamically discoverable

PROCESSES THAT CAN RESPOND TO CHANGE

Copyright © 2005, ZapThink, LLC



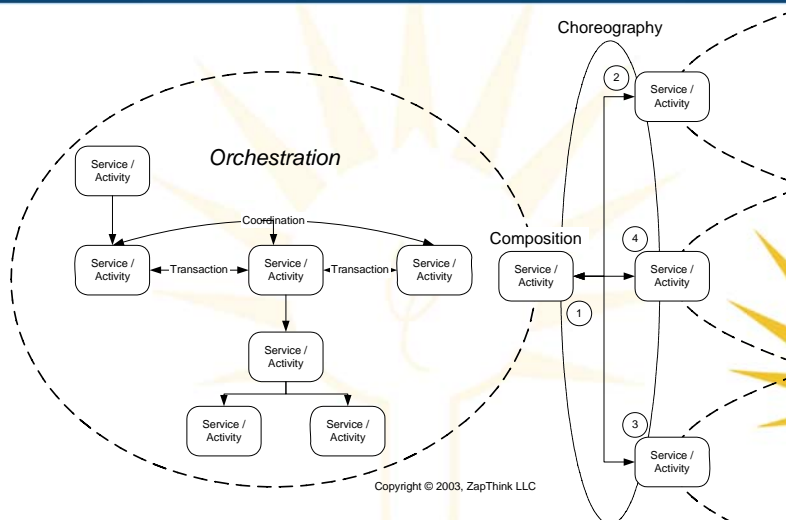
SOP Concepts

- *Composition*
 - Turning fine-grained atomic Services into coarse-grained business Services
- *Orchestration*
 - Composing Services using a logical Flow
- *Choreography*
 - Interacting between independent processes
- *Coordination*
 - Defining the relationship between two or more Services
- *Transaction*
 - A *reliable* relationship between two or more Services

Copyright © 2005, ZapThink, LLC



SOP Concepts





zapthink

What about Workflow?

- Involving the Human
- Is BPEL / BPML / WSCI sufficient?
- Other Specifications
 - Wf-XML
 - XPDL
- Concept: Portals as a gateway for workflow?



Copyright © 2005, ZapThink, LLC



zapthink

Processes: App Servers or Message Buses

- “Application Processes”
 - Connecting Services
 - Everything runs on the App Server
- “Business Processes”
 - Defining and Executing Processes
 - Activities can (and do) run everywhere
- Infrastructure choices
 - App Server: For Application Processes
 - Message Busses: For Business Processes

Copyright © 2005, ZapThink, LLC



zapthink

Combining with Service-Oriented Management

- Achieving Loose Coupling
- Managing Atomic Services
 - Making sure the activities work!
 - Processes *are* Services
- Managing End-to-end Processes
- Keeping the abstraction in place



Copyright © 2005, ZapThink, LLC



zapthink

Process Management

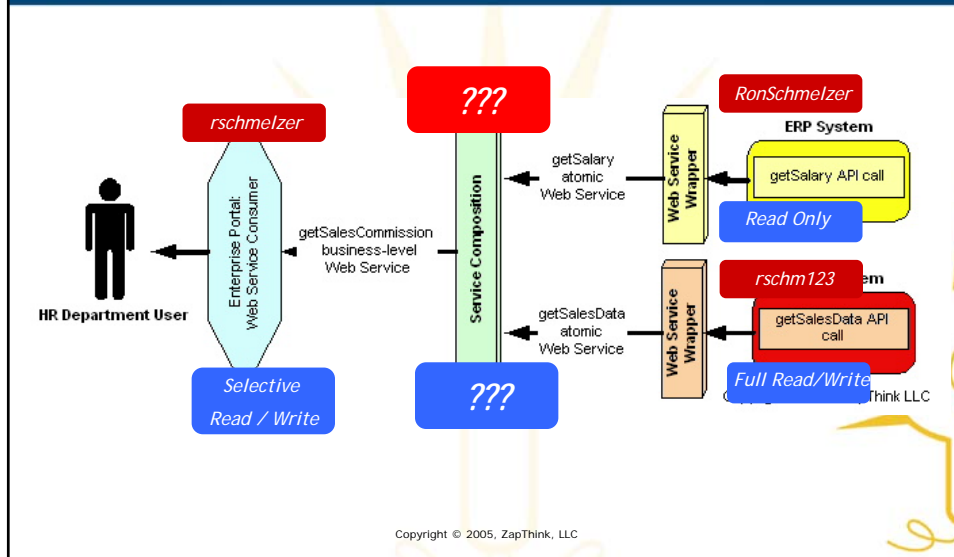
- Business side to Process Management
 - Did this process fulfill my business goals?
 - How did this process flow compare with a different process flow?
 - How do two process instances compare?
 - What happens if I change my process?
 - What happens if I change an activity?
- Process dashboards, monitors

Copyright © 2005, ZapThink, LLC



zapthink

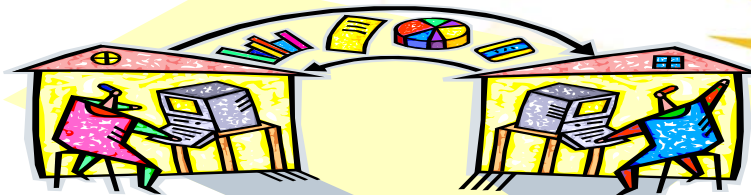
Context / Domain Dependencies

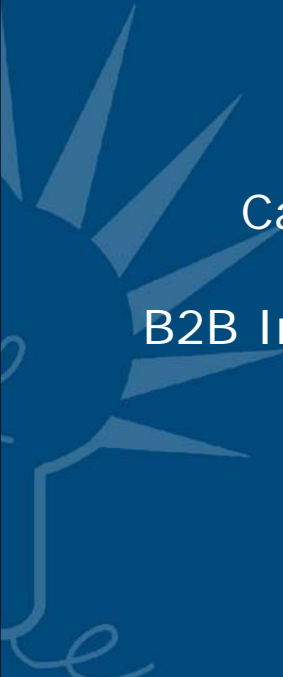


zapthink

Enterprise Applications and Process

- The problem with enterprise applications
 - Process bound to Functionality
 - High customization and integration cost
- New approach
 - Atomic Enterprise App Functionality
 - Separate Process Layer





Case Study: E2open

B2B Integration and Process

Copyright © 2005, ZapThink, LLC


zapthink



Introduction

- Who is E2open?
 - Worldwide B2B Supply Chain Network
 - Member companies include: Acer, Hitachi, IBM, LG Electronics, Lucent Technologies, Matsushita Electronic (Panasonic), Nortel Networks, Seagate Technology, Solectron, Toshiba, Mitsubishi Electronic, Omron, Ricoh, Sanyo, and Sharp
- What problem are they solving
 - How to efficiently connect companies in a supply chain and connect “cross-company industry process”

Copyright © 2005, ZapThink, LLC





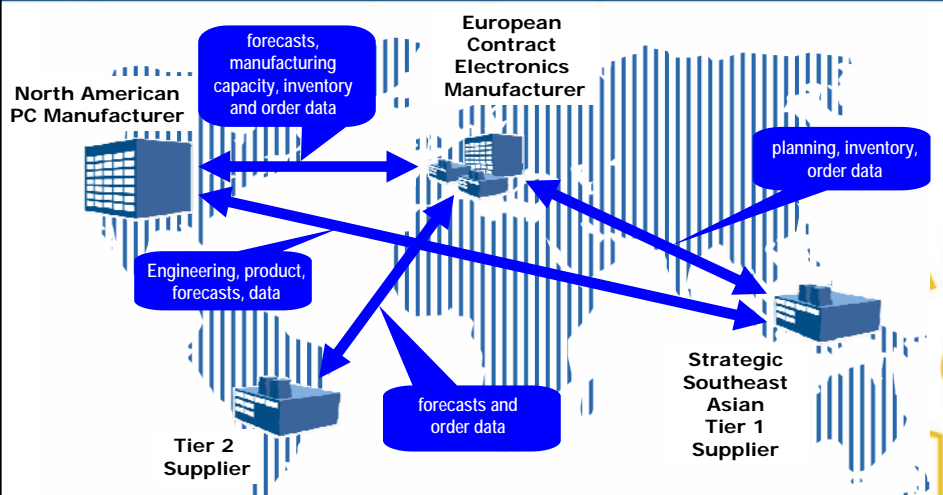
Framing the Problem

- Pressures on high-tech manufacturing companies:
 - Outsourcing (reduce cost)
 - Globalization (increase supply)
 - Time-to-market (reduce time)
 - Just-in-time manufacturing (reduce inventory)
 - Distributed value chain (satisfy customer)
- Resulting in IT Challenges
 - Integrating across multiple companies
 - Long-lived, multi-step, complex processes
 - Automating manual processes
 - Business semantic issues

Copyright © 2005, ZapThink, LLC



Typical Supply Chain...



Integrated and synchronized inter-company business processes



zapthink

Solution Framework

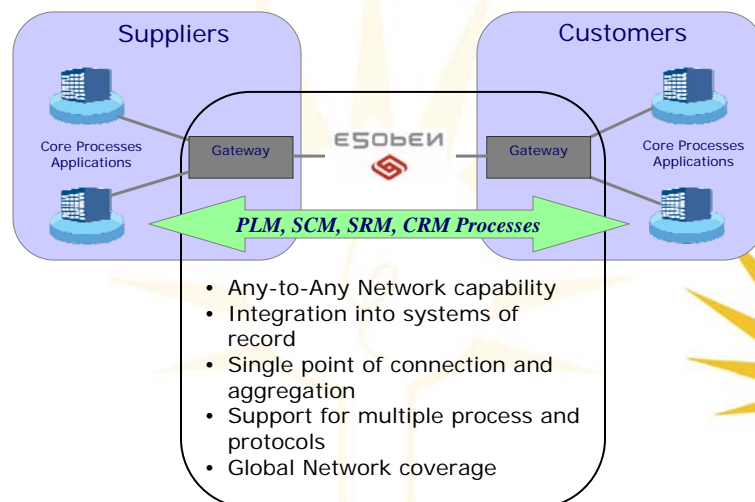
- Create standards-based collaboration framework leveraging process:
 - B2B process transformation
 - Create external process interfaces at supplier and manufacturer
 - Create a framework by which these processes can be synchronized
 - Standards-based technology infrastructure
 - Use industry standard vocabularies (RosettaNet, etc.)
 - Use standard transport protocols (HTTP, SMTP)
 - Use Registry (UDDI) to decouple the end points

Copyright © 2005, ZapThink, LLC

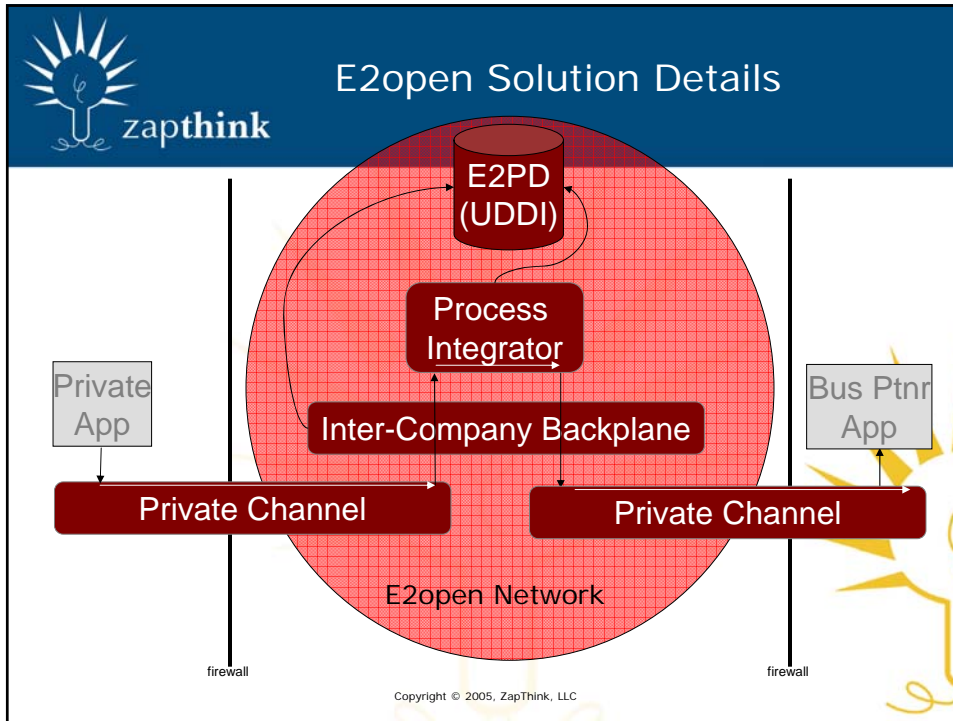


zapthink

E2open Solution



Copyright © 2005, ZapThink, LLC



Customer Example: Acer

- Company
 - In the top 10 of Personal Computer manufacturers
 - Annual chip purchases > \$700 million
- Challenge
 - Submitting supply forecasts 4-5 months in advance
 - Suppliers slow to respond
 - Result: forecasts that didn't match sales, inventory problems

The slide includes the Zapthink logo in the top left corner and the text "Copyright © 2005, ZapThink, LLC" at the bottom center.



zapthink

Results: Acer

- Forecast-to-order in real-time
- Reduced inventory dramatically
- Reduced cost of adding new suppliers
- Increased ability to negotiate new supplier relationships

Copyright © 2005, ZapThink, LLC



zapthink

Customer Example: LG Electronics

- Company
 - One of the largest electronics companies in Korea
- Challenge
 - Wanted faster, more efficient way for collaborative product development
 - Wanted real-time visibility into project status
 - Communication and planning problems lead to product release delays, impacting revenue

Copyright © 2005, ZapThink, LLC



Results: LG Electronics

- Solution: collaborative process design using integrated B2B processes
- Results
 - Development project that usually took 2 weeks took 4 hours
 - One project showed design time reduced from 53 days to 20 days
 - Meetings in real-time
 - Travel costs reduced by tens of thousands of dollars
 - Now capable of extending product design process to third-parties

Copyright © 2005, ZapThink, LLC



Lessons Learned

- Web Services over the Internet is Viable -- Today
- B2B Integration is about:
 - Choreographing processes
 - Agreeing on business semantics
 - Handling security and reliability issues
 - Making deployment easy
- We don't have to repeat the mistakes of the EDI past

Copyright © 2005, ZapThink, LLC

Thank You!



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



Jason Bloomberg
jbloomberg@zapthink.com



Ronald Schmelzer
rschmelzer@zapthink.com