

AVIVA
March 2006

Service Oriented Architecture: A Case Study

Aviva Life Insurance

Brian A. Hebert
Director, Distributed Systems
Aviva Life Insurance

Overview

- Introduction
- About Aviva
- Needs Analysis
- Proposed Solution (SOA middleware)
- Proof of Concept
- Future Plan

Aviva Life Insurance Company of America

- Part of Aviva plc, one of the 10 largest insurance groups worldwide
- \$498 billion in assets under management
- One of the fastest growing US life insurers
- Focused on products such as:
 - **Universal Life**
 - **Term and Whole Life**
 - **Wealth Transfer Life Insurance**
 - **403(b) TSA Annuities**
 - **Structured Settlement Annuities**

The Business Driver

- All business lines needed to sustain large-scale growth
 - New Products
 - New Technology
 - New Partners
- Customer service had to be dramatically improved
- Data processing would have to have the highest degree of quality control possible

Analysis Phase: Understanding the current environment

- Individual systems performing specific tasks
 - Second Tier, Financial, Web, Mainframes, Workflow
- Different technology platforms aimed at servicing different groups
 - Agents, Admin, IT, Finance, etc.
- Multiple authoritative sources of data
- Overlapping system responsibilities

Analysis Conclusion: Complex environments promote inefficiencies

- Isolated development efforts
- Fracturing of business logic
- Manual processing
- Redundant data entry
- Data inconsistencies
- Complex training requirements
- Limited growth potential

Aviva's Solution: Establish a SOA middleware

- Simplified integration of existing systems
- Easy to introduce new applications
- Added functionality in one system becomes immediately available to the entire enterprise
- Able to blend vastly different technologies
- Enables all transactions to be monitored, secured, and audited

Why Service Oriented Architecture?

- Reusability
- Integration
- Consolidation
- Efficiency
- Flexibility

A highly streamlined enterprise that costs less to maintain and can rapidly adapt to an ever changing business environment.

Proof of Concept: Agent Licensing Problem

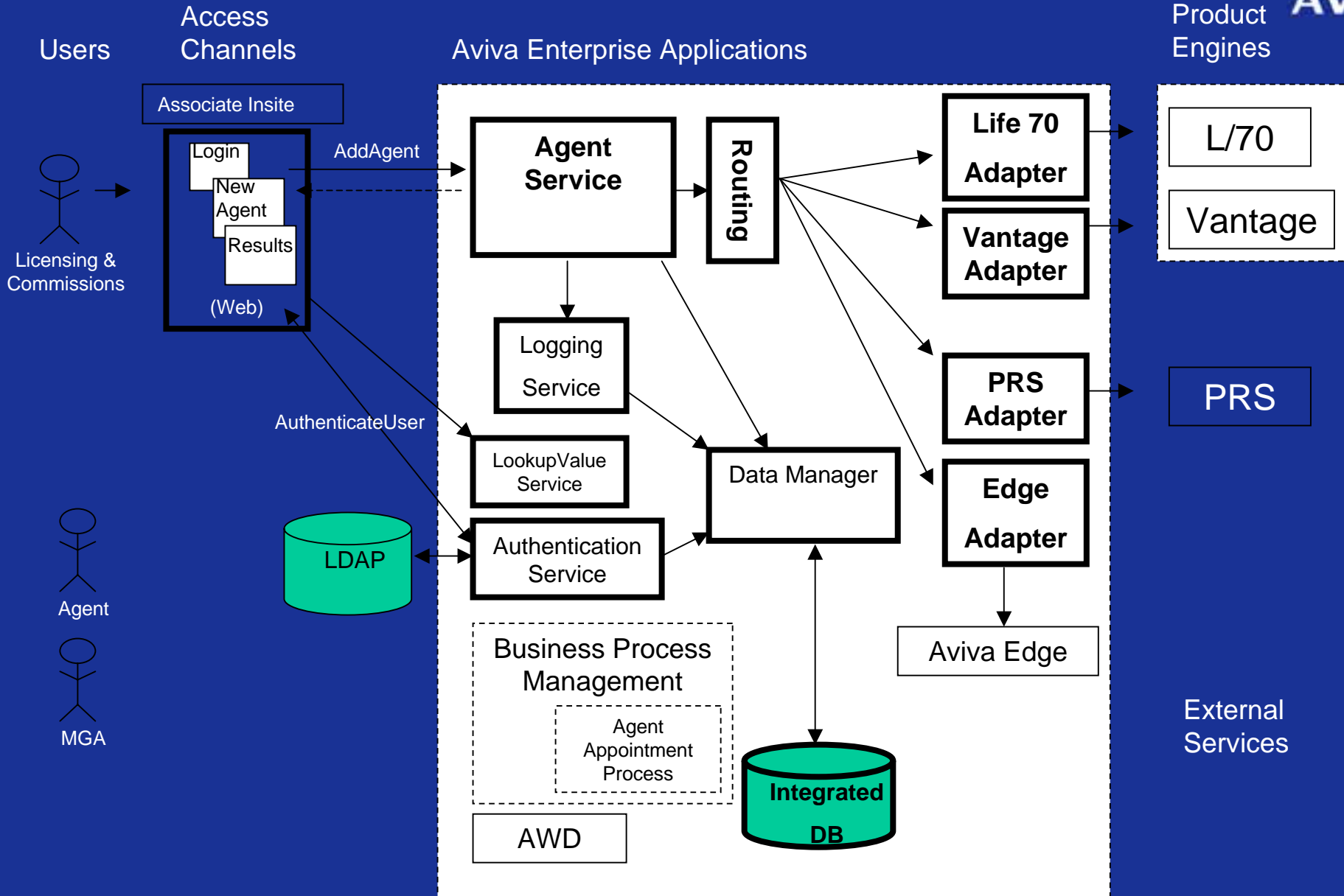
- Manual data entry
- Redundant data entry
 - Data was input into four separate systems
 - A single agent took two hours to input
 - Information accuracy was at 52%
- Training difficulties
- Anticipated increase in agent requests

This was a real business problem that middleware had a real solution for.

Proof of Concept: Agent Licensing Solution

- Condense data entry to a single form
- Automatically distribute the information
- Consolidate business rules within enterprise services
- Use this opportunity to launch the middleware
- Larger initial investment
 - Foundational services
 - Enterprise standards (Coding, XML, Architecture, etc.)
 - Learning process for development team

Agent Licensing Project



Proof of Concept: Agent Licensing Accomplishments

- Dramatic Improvements
- Number of data fields: 30 (from 400)
- Average processing time: 5 minutes (from 2 hours)
- Average data accuracy: 98% (from 52%)
- Average training time: 1 week (from 6 months)

Special thanks to Scott Koehler from Koehler Consulting for helping make this project a success.

Proof of Concept: Agent Licensing Deliverables

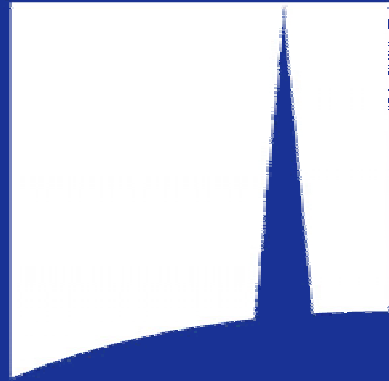
- New technology stack
 - Oracle 10g Application Sever & Database
 - Oracle Application Interconnect
- New enterprise standards
 - XML & Java standards
 - Enterprise Architecture Roadmap
- Web service foundation
 - User provisioning services
 - Error and logging services
 - Data access services
 - Mainframe integration services

The Future: Interconnecting Web Services through Orchestration

- The logical evolution of SOA
 - As services increase in number and complexity, it becomes increasingly cumbersome to embed business logic within them.
 - Web services excel at performing well defined tasks, but tend to lack an effective means of centrally controlling business processes.
- Serves a unique and specific role
 - Process automation takes advantage of the reusability inherent in SOA to interconnect web services.
 - An orchestration layer hides the complexities of process management from the Web Services, which in turn increases reusability.
- Essential for continued growth
 - The enterprise architecture must be continually streamlined to combat the natural complexities that come with expansion.

Lessons Learned

- Have a clearly stated business driver
- Set realistic and measurable goals
- Establish enterprise standards and enforce them
- Realize that you may need external help
- Apply SOA principles to EVERYTHING you build
- Don't be afraid to challenge your long-term plans
- Realize that packaged software can only take you so far



AVIVA