Building a Future Proof SOA

Ross Altman
CTO – Software Infrastructure
Sun Microsystems

Why Composite Applications? Because our requirements have changed.

- 1 mainframe computer
- 1 user device
- 1 network connection
- 1 population of users
- 1 set of user requirements
- 1 user location
- 1 program
- **1 program “owner”**

- Many computers
- Many user devices
- Many types of network connections
- Many diverse populations of users
- Many different user requirements
- Many user locations
- Many programs
- **Many program “owners”**

What is a composite application?

- A composite application leverages existing and independently developed applications, information stores and new business logic
- A composite application combines these resources to create a brand new solution of a business problem that these current resources cannot solve on their own.
- In some cases it requires human involvement beyond what is afforded by the existing applications. New user views and actions may be needed.
- Throughout the enterprise, many composite applications may exist:
  - Some may be “stand alone” while others are loosely associated
  - Some may be recursively built on top of other composite apps

A Composite Application

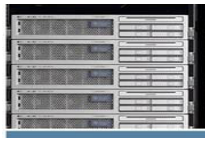
Shipment Tracking

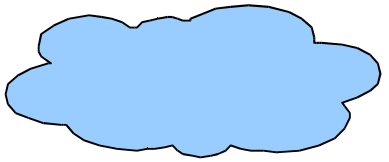
Olympic Mutual - Shipping

Ownership & Valuation

In-transit Location

Shipping Manifest





Enterprise Systems

Divisional Systems

Remote Feeds

Inventory

Logistics

Customer Management

Container Design

Freight Tracking

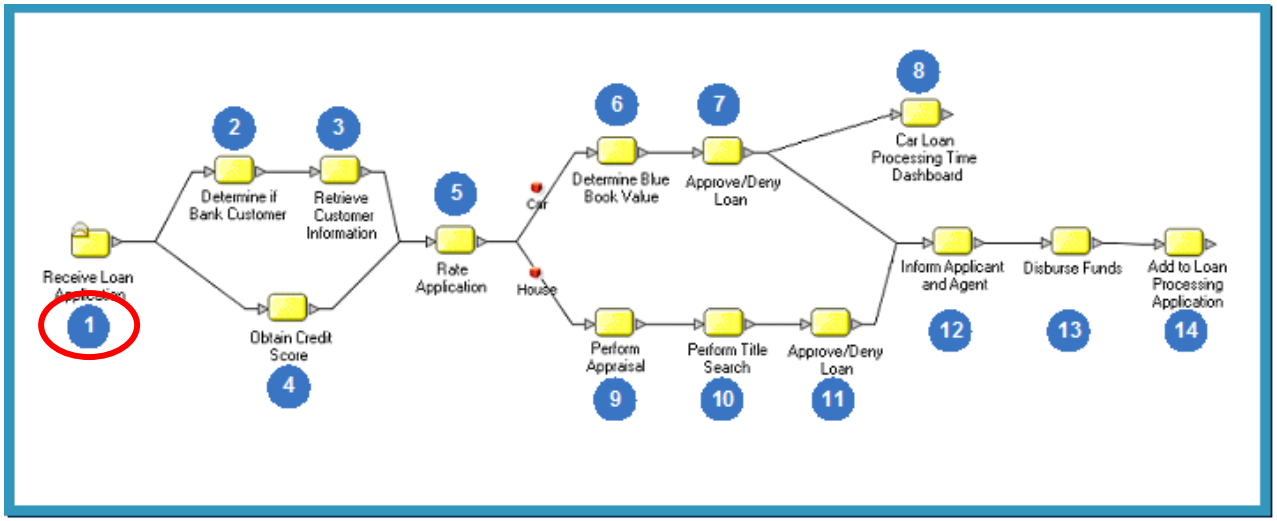
RealTime Locator

Mapping

A Composite Application

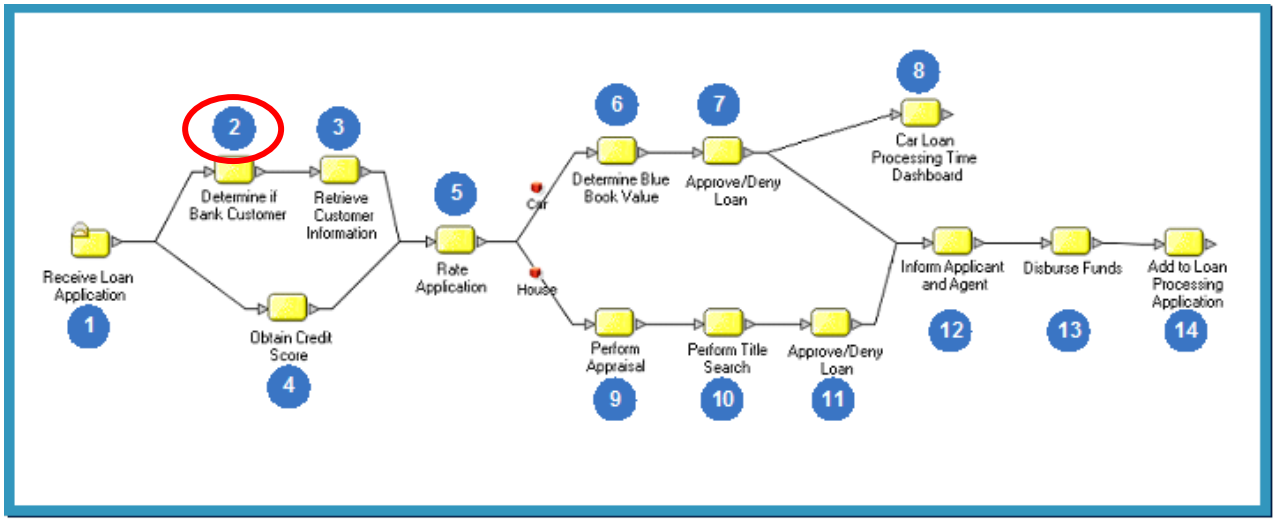
Loan Application Processing

Olympic Mutual's Loan Processing System



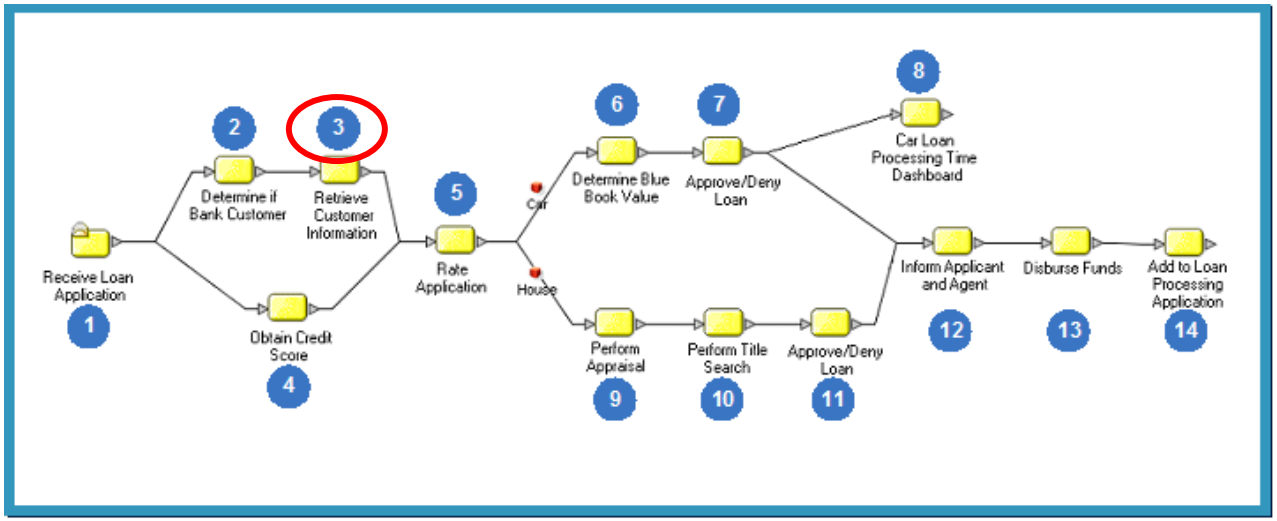
1. Tri-Valley set up a B2B gateway, which is used by large car dealerships and real estate offices to submit loan applications on a program-to-program basis over the Internet. But, since this B2B gateway is overkill for smaller car dealerships and real estate offices, the bank has also implemented a portal-based program where loan applications can be submitted manually.

Customer Data Integration



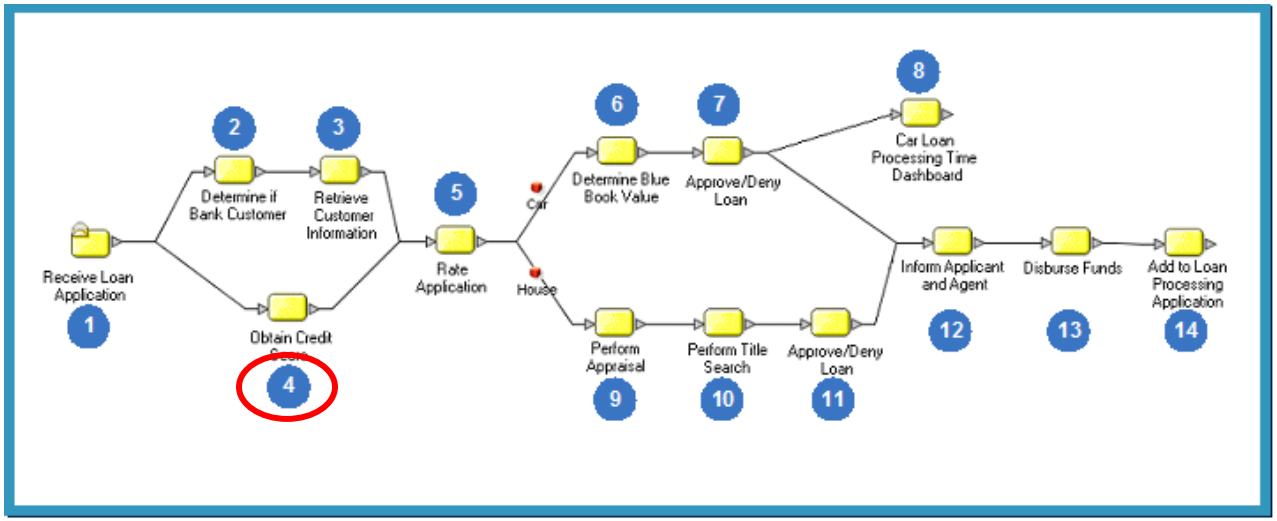
2. Once the information on the loan applicant arrives, the bank determines if the applicant is a past or current customer of the bank. Since the loan application doesn't always include Tri-Valley Bank account numbers, the bank uses the applicant's name, address and phone number to try to match the applicant to loan and account records. Using probabilistic matching algorithms in master data management software, the bank determines whether the submitted data matches any of the bank's savings accounts, checking accounts, certificates of deposit, car loans or mortgages.

Access to Multiple Internal Applications and Data



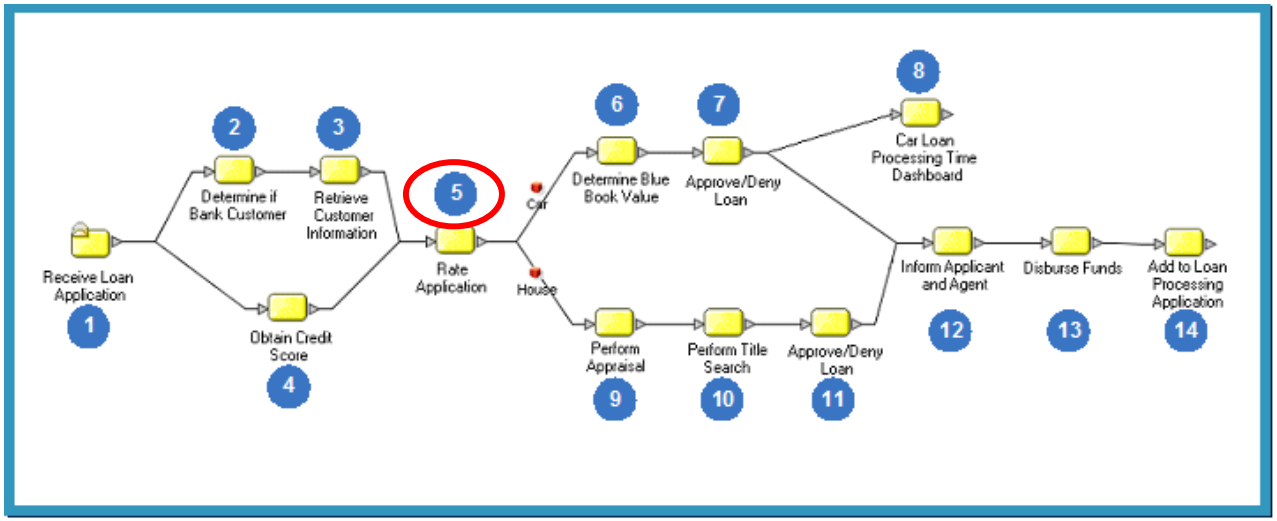
3. If the applicant has an account history at the bank, then this data is retrieved from the relevant databases and transformed into a common data format.

Access to External Credit Bureau(s)



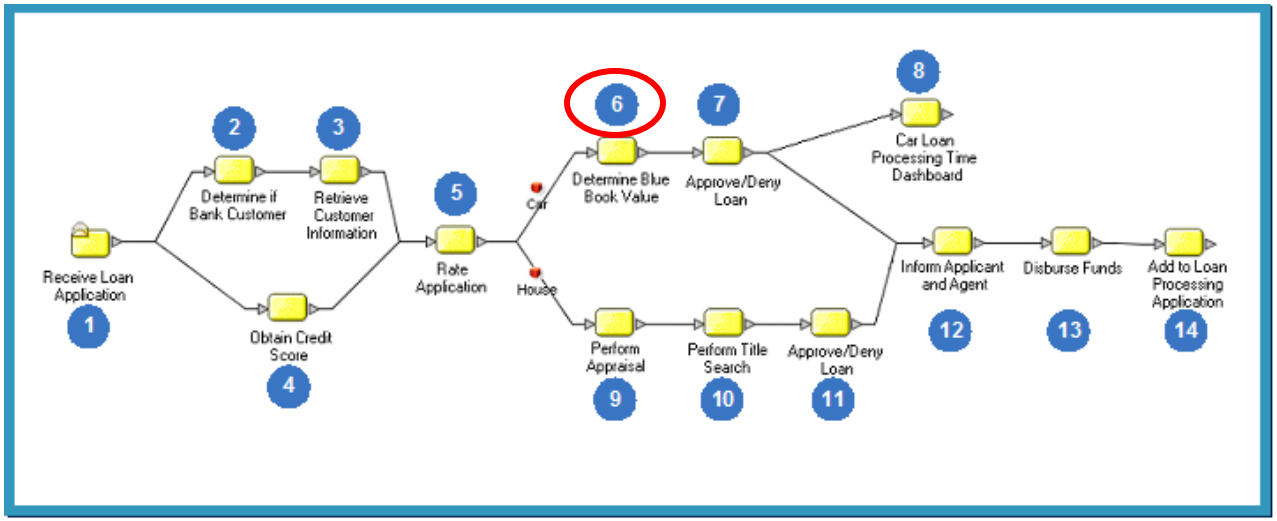
4. While this master data management and account data retrieval is taking place, the bank sends a B2B message to a credit bureau to secure a third-party credit rating for the applicant.

Access to External Credit Bureau(s)



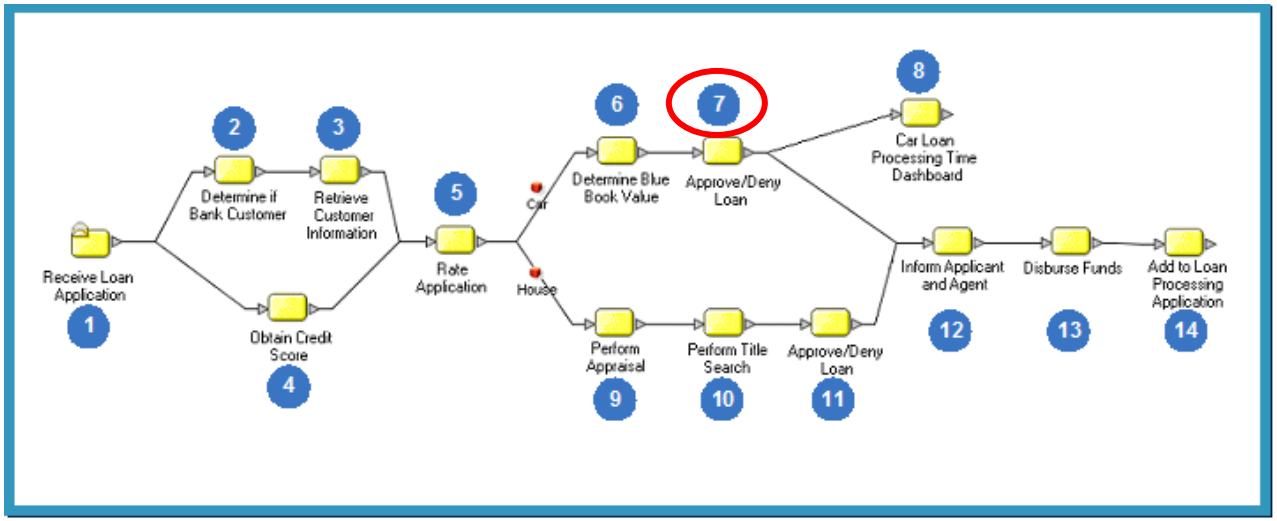
5. Once all the required data has been collected, a rules engine applies a set of rules that have been built by one of the bank's experts. This rules stack is used to rate the applicant.

Determining the Value of the Collateral



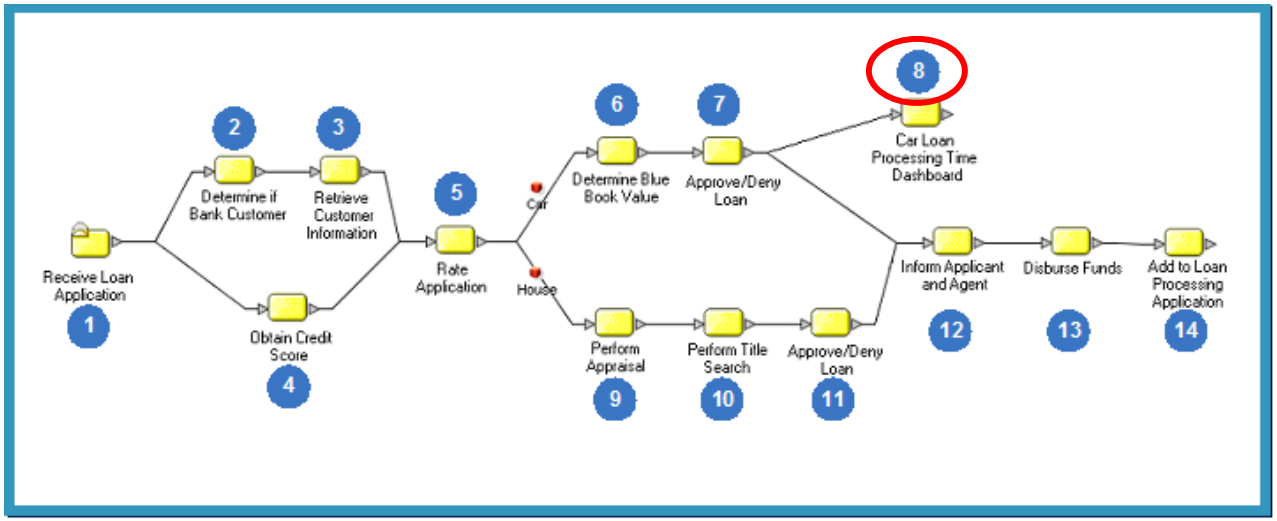
6. To this point, the processing for both car loans and mortgages has been completely automated. If the application is for a car loan, this automation continues as a B2B message is sent to a car rating service to determine the current value of the car that will serve as collateral for the loan.

Automatically Approving the Loan



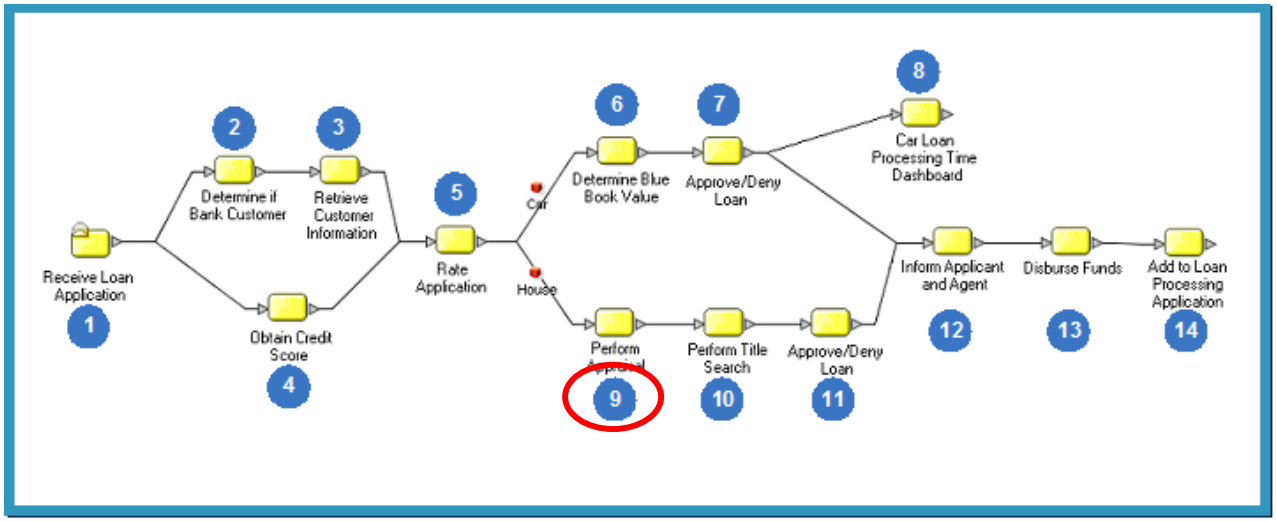
7. Once that information has been secured, a loan approval program automatically approves or denies the application based on the amount of the loan, the value of the car and the credit score that was assigned to the applicant.

Tracking Business Processes in Real Time



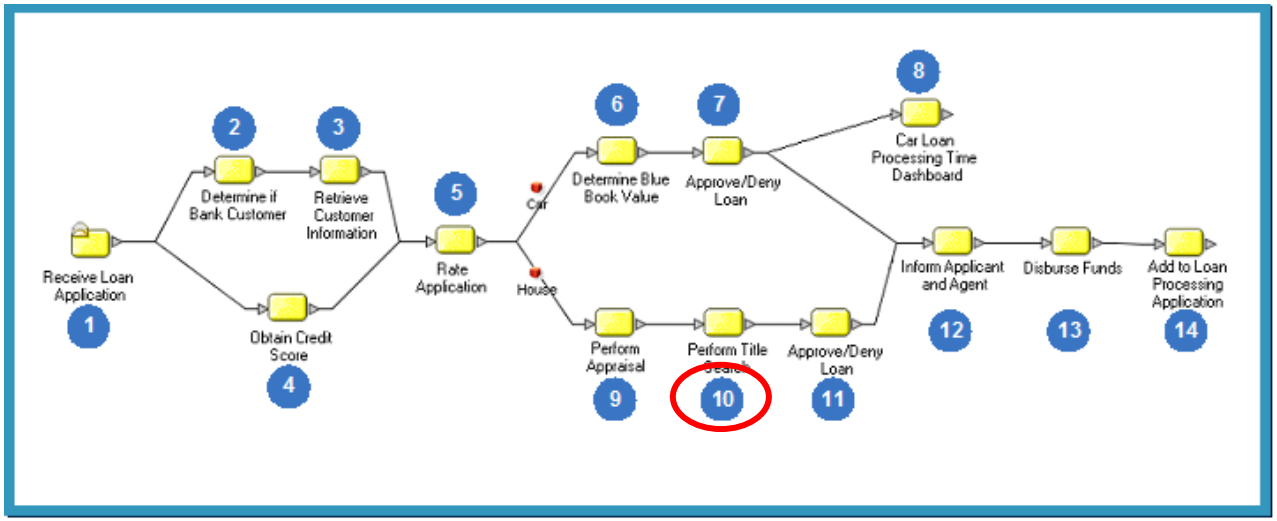
8. As soon as the loan is approved, a message is sent to a browser-based car loan processing dashboard, which is used to manage the response time of this loan processing system. To do this, a time stamp is attached to the loan application data when it arrives at the portal or the B2B gateway. A second time stamp is attached to the loan application data when the loan is approved or denied. These time stamps allow the dashboard application to display the average response time for loans over the last ten minutes. Also, a monitoring program tracks this response time and sends an alert (via email, pager, SMS messaging, fax . . .) whenever the average response time for the last ten minutes exceeds five seconds.

Determining Value of Collateral – an Appraisal



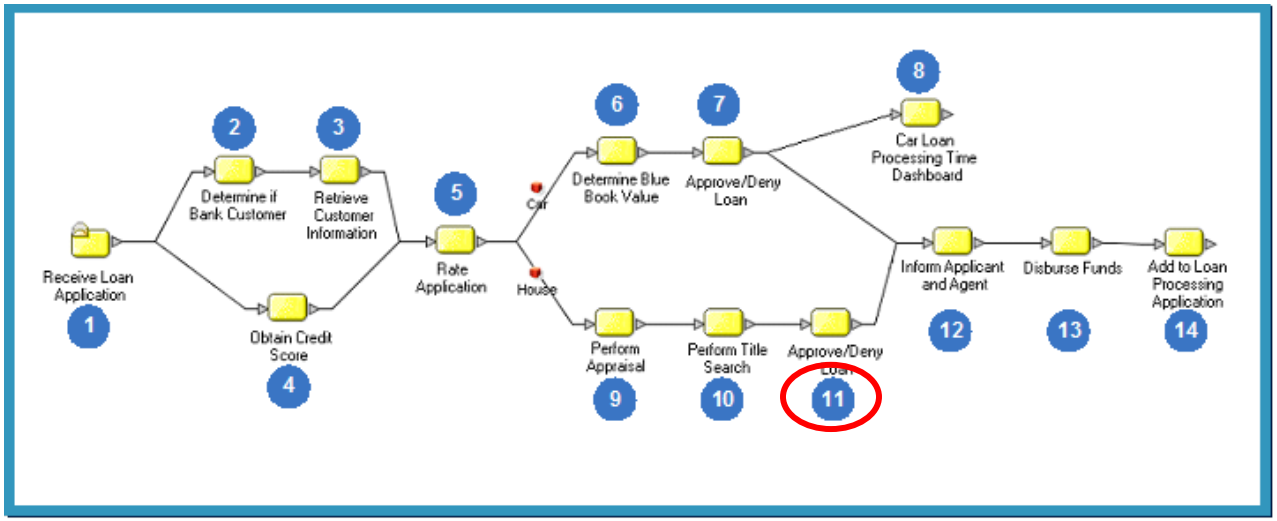
9. If the application is for a home mortgage, then the processing is substantially different. First, using workflow functionality, a message is placed in a work queue for a real estate appraiser, who must determine the market value for the property.

Performing a Title Search



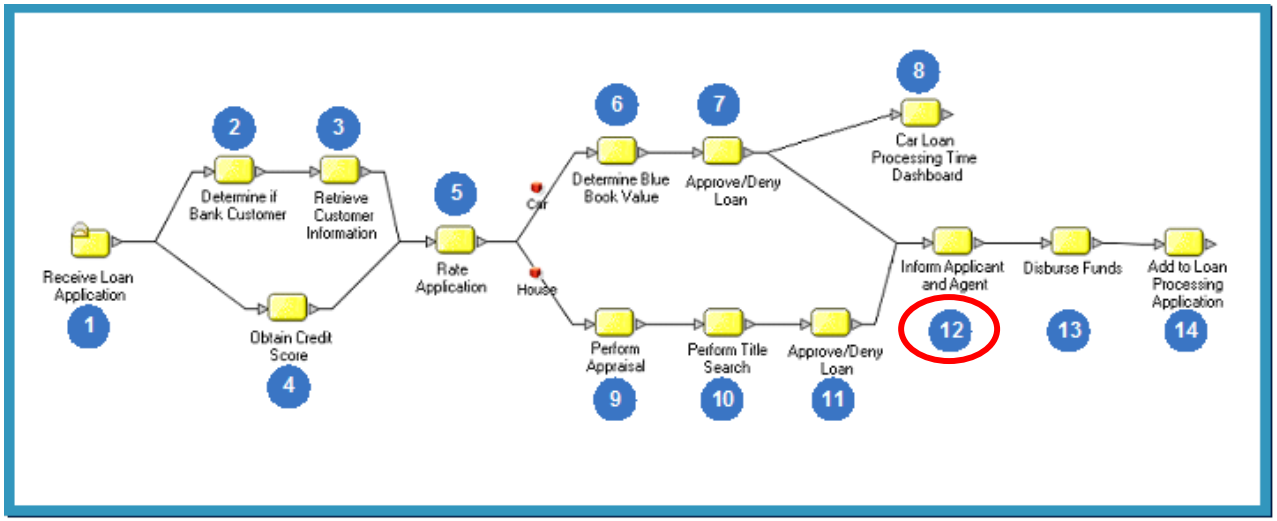
10. Then, more workflow is involved as a message is placed in a work queue for person to perform a title search on the property, to determine if there are any liens or other encumbrances on the title.

Approving the Loan – the Role of the Loan Officer



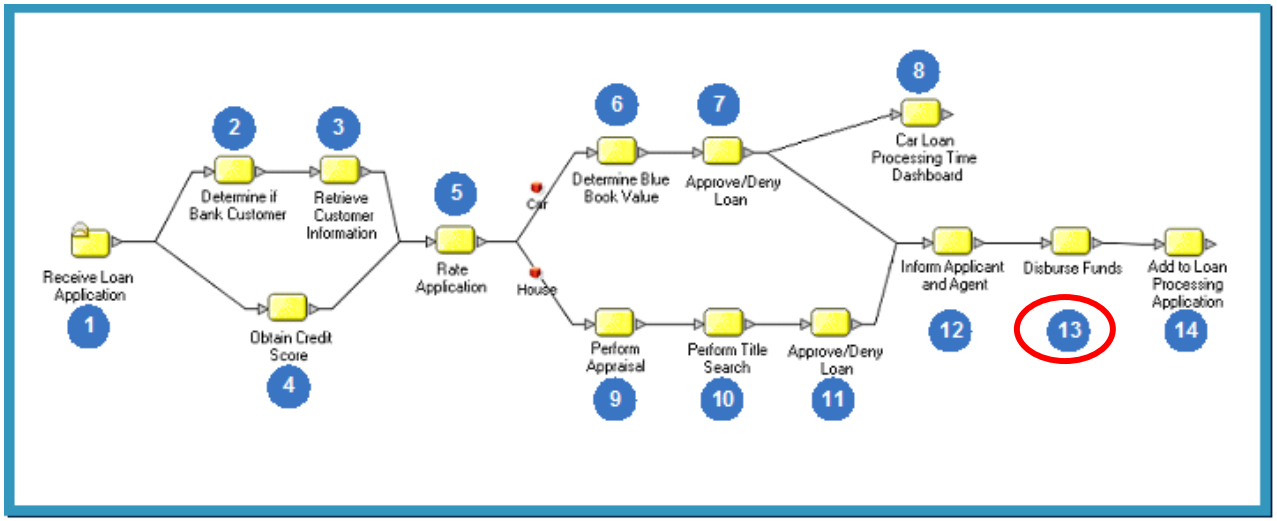
11. Once the appraisal and title search are completed, a loan officer reviews the applicant's credit score and the appraised value of the property to determine if the loan should be approved.

Informing the Applicant and the Applicant's Agent



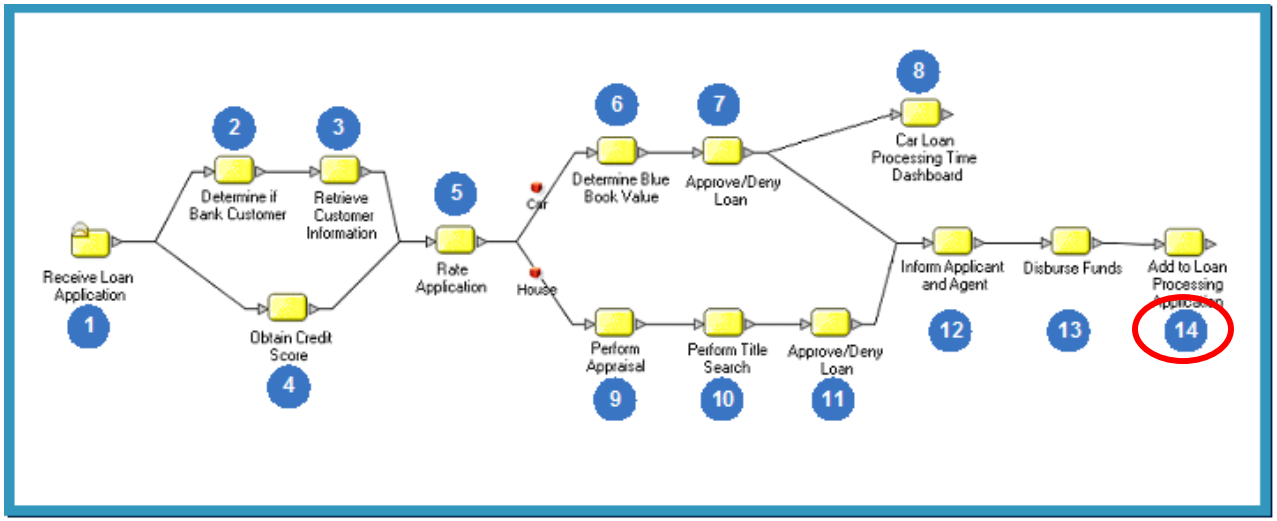
12. For both car loans and mortgages, once a decision has been made on the loan application, three steps are completed automatically. First, a message is sent to the application and his or her agent to indicate the decision.

Disbursing Funds



13. Then, if the loan has been approved, a message is sent to the bank's funds disbursement application so that the amount of the loan can be transferred to the seller.

Adding the Loan to the Loan Processing App



14. Finally, information about the approved loan is sent to the application used by the bank to collect loan payments from the borrower.

The Business Value of Composite Applications

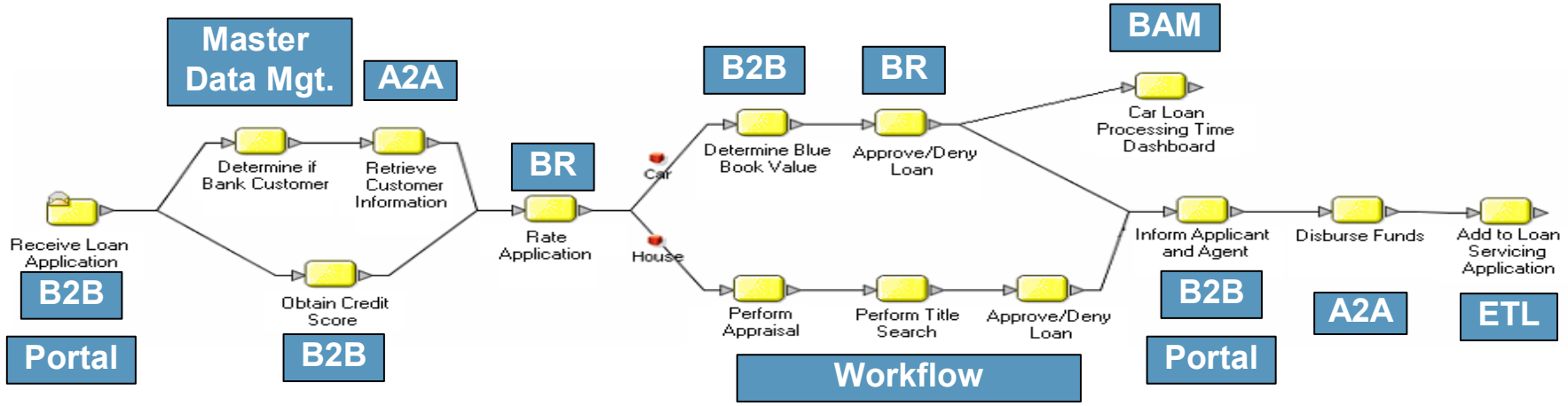
How Composite Applications Deliver Business Value

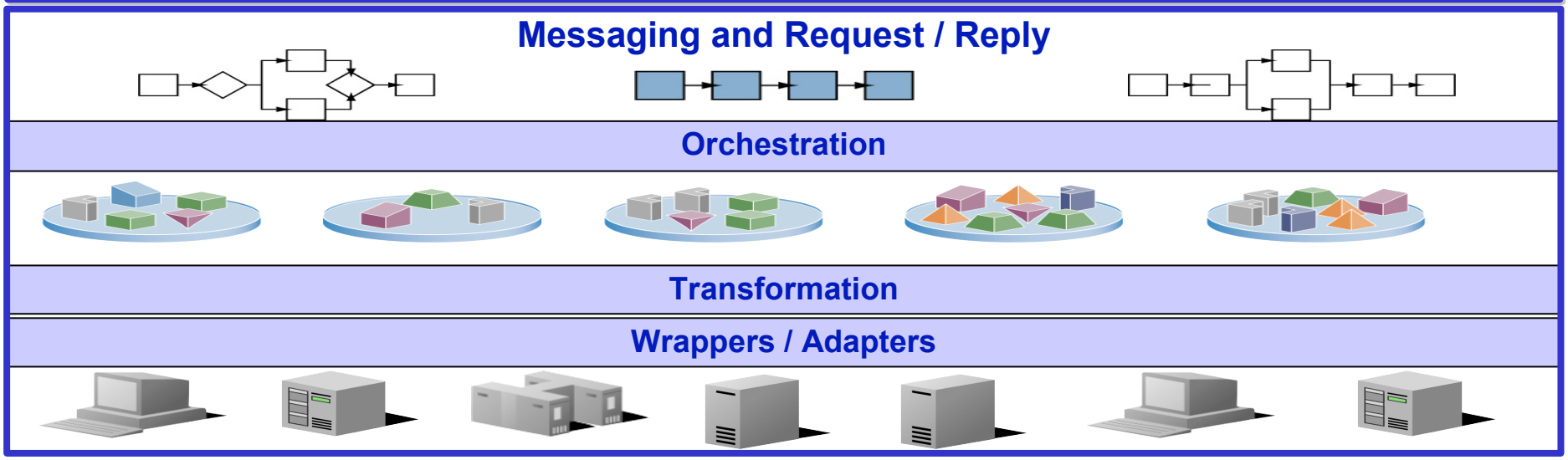
- Reduce personnel costs by automating process steps
- Reduce requirements for highly skilled decision-makers
- Decrease the frequency of “wrong decisions”
- Reduce process latency
- Track execution of multi-step processes to ensure process efficiency
- Track results of processes (through Business Activity Monitoring) to enable realtime process enhancement
- Automate some exception handling
- Thoroughly monitor and manage processes for compliance purposes

Future Proof SOA

- ✓ **Completeness of the Suite**
- Integration of the Suite
- Secure SOA
- Standards support for both Interoperability and Portability
- Near Zero Coding™
- Aspect-Oriented Implementation

There are a lot of “moving parts” in composite applications development on an SOA.





Future Proof SOA

- Completeness of the Suite
- ✓ **Integration of the Suite**
- Secure SOA
- Standards support for both Interoperability and Portability
- Near Zero Coding™
- Aspect-Oriented Implementation

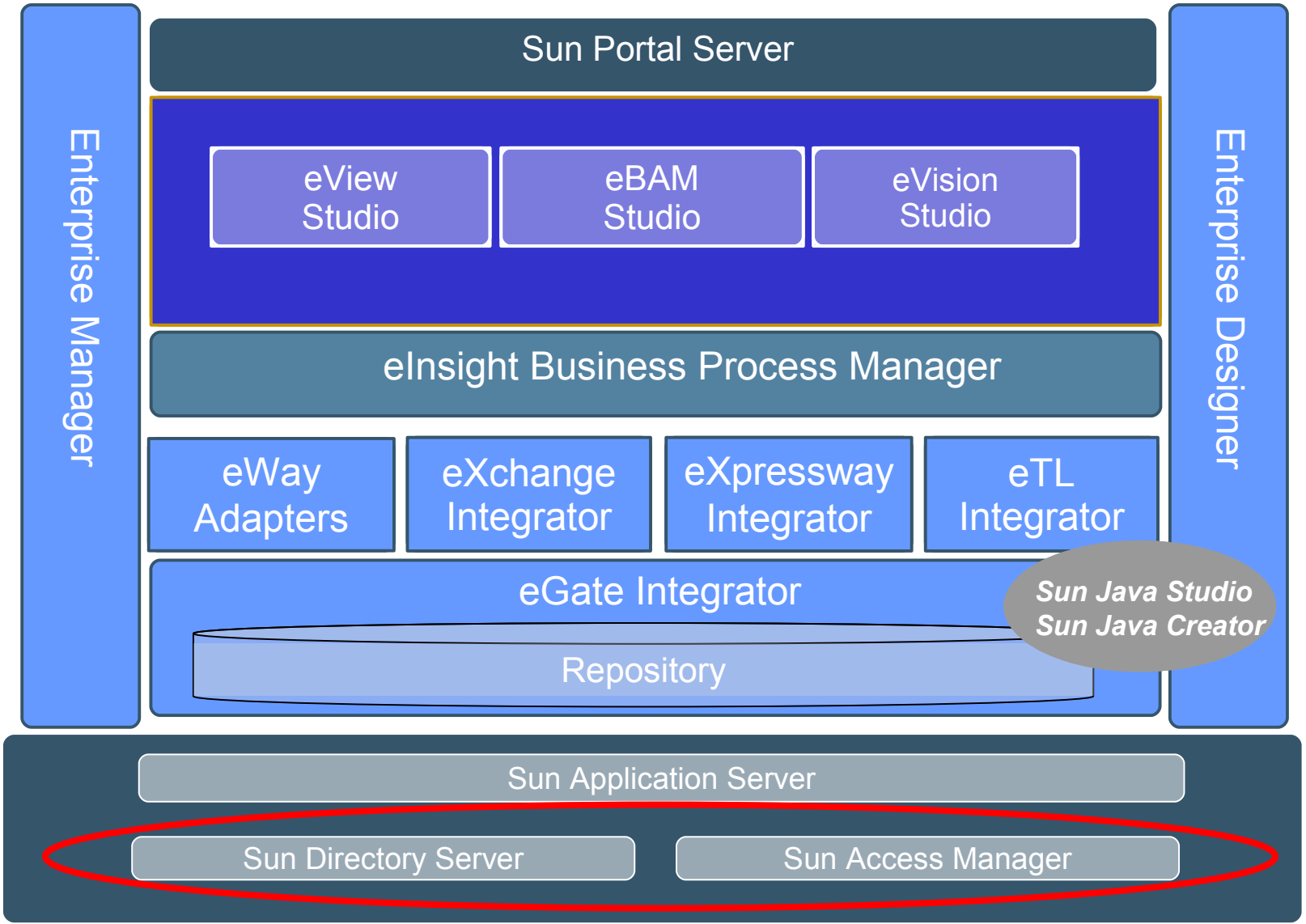
Integrated integration tools reduce costs throughout the SDLC.

- Advantages of a single development environment
  - A consistent look and feel
  - Reuse of development tools
- Advantages of a single repository
  - Better support for reuse.
  - Better support for impact analysis.
  - Better support for version control and configuration management.
- Advantages of a single runtime
  - A single management framework.
  - A single security framework
  - A single transaction management framework
  - A single debugging environment

Future Proof SOA

- Completeness of the Suite
- Integration of the Suite
- ✓ **Secure SOA**
- Standards support for both Interoperability and Portability
- Near Zero Coding™
- Aspect-Oriented Implementation

Access Management is critical to Message-Level Security.



Future Proof SOA

- Completeness of the Suite
- Integration of the Suite
- Secure SOA
- ✓ **Standards support for both Interoperability and Portability**
- Near Zero Coding™
- Aspect-Oriented Implementation

An SOA suite should be standards-based for interoperability and portability

There are two types of standards:

Interoperability standards: "Write Once / Run Anywhere"

Interoperability standards: "Plug-and-Play"

A mature standard is functionally complete and ubiquitous.

What's required for a standard to reach maturity?

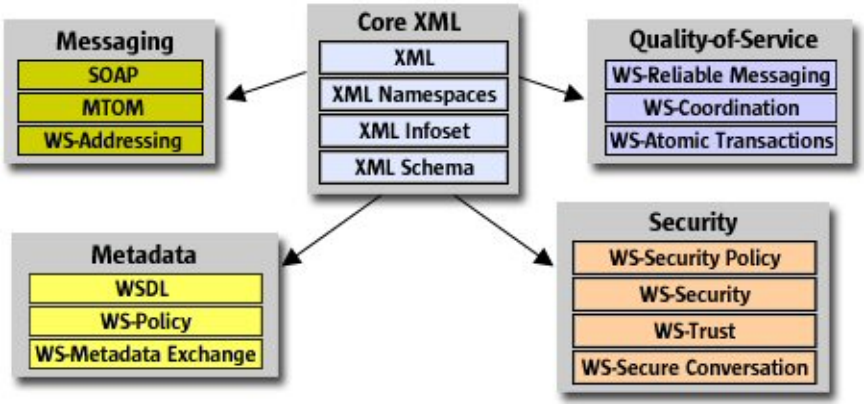
- ✓ Complete Specification
- ✓ Reference Implementation
- ✓ Certification
- ✓ Coercion

Project Tango provides Web services with Plug-and-Play Interoperability.

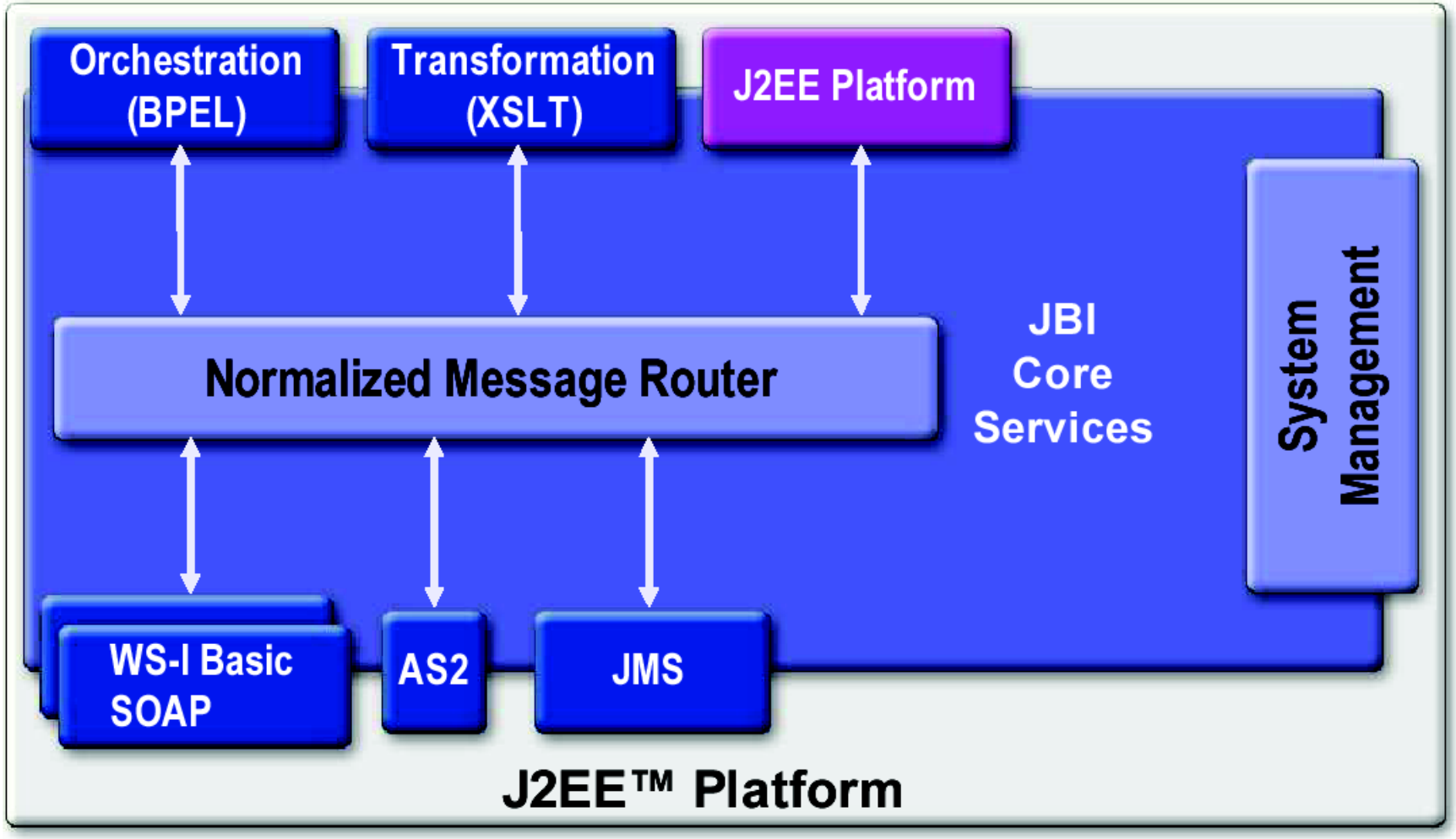
- Sun & Microsoft have collaborated on interoperable Web services stacks:
  - Microsoft's is included in **Windows Communications Foundation**.
  - Sun's is provided by **Web Services Interoperability Technology** (WSIT).
- There will be millions servers and PCs running copies of WSIT and WCF in 2-3 years.
- This will form the ecosystem that will enable plug-and-play Web services interoperability.

To ensure interoperability:

- WCF will be the de facto reference implementation
- Middleware and application vendors will "certify" interoperability with WCF
- 100's of millions of copies of WCF/WSIT will provide the "coercion"
- Plugfests will be the "certification mechanism" to ensure compatibility



Java Business Integration allows components to plug into an ESB “backbone”.



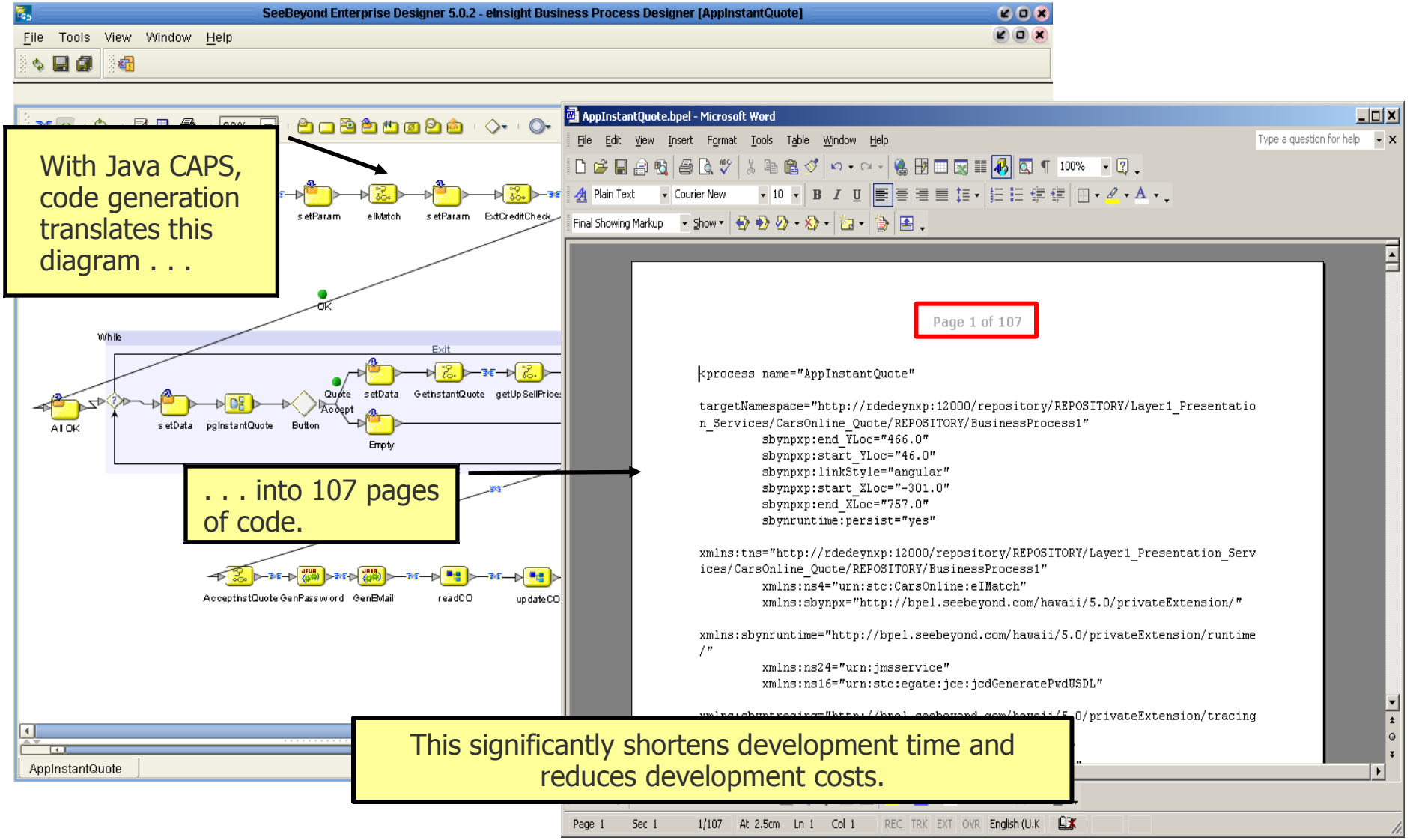
Non-Sun Open ESB Contributors

- Adnet – Workflow Service Engine
- Bostech – TCP/IP Binding Component, Scheduler Service Engine
- Eviware – Soap UI Netbeans Plugin
- Gestalt-LLC – XMPP, SIP, UDDI & RSS Binding Components
- Imola – CICS & CORBA Binding Components, Semantic Web Service Engine
- Neociclo – Odette FTP, Odette-J Binding Components
- Vikram India – DCOM Binding Component
- Yasu – Business Rules Service Engine
- Art Gillespie – 3270 Binding Component
- Jon Strayer & Art Gillespie – Screen-scraping Binding Component
- Gary Zheng – LDAP Binding Component
- Oscar Montoya – SWIFT Binding Component
- Stelian Markov – eMail Binding Component

Future Proof SOA

- Completeness of the Suite
- Integration of the Suite
- Secure SOA
- Standards support for both Interoperability and Portability
- ✓ **Near Zero Coding™**
- Aspect-Oriented Implementation

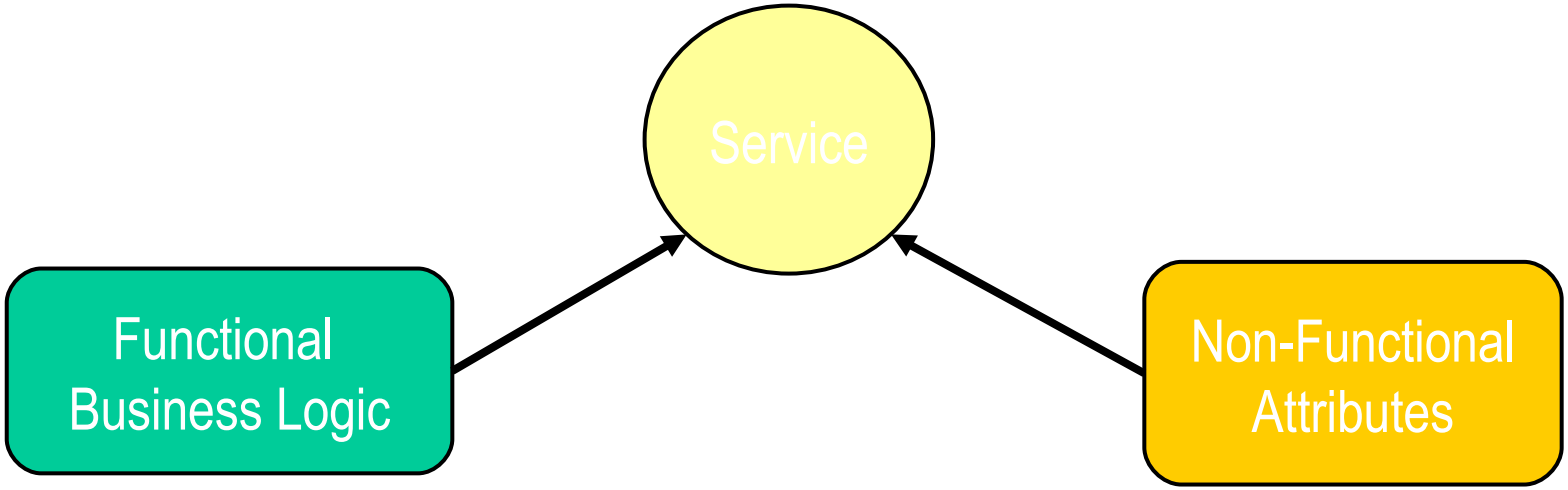
Near Zero Coding™ and "Round-Tripping"



Future Proof SOA

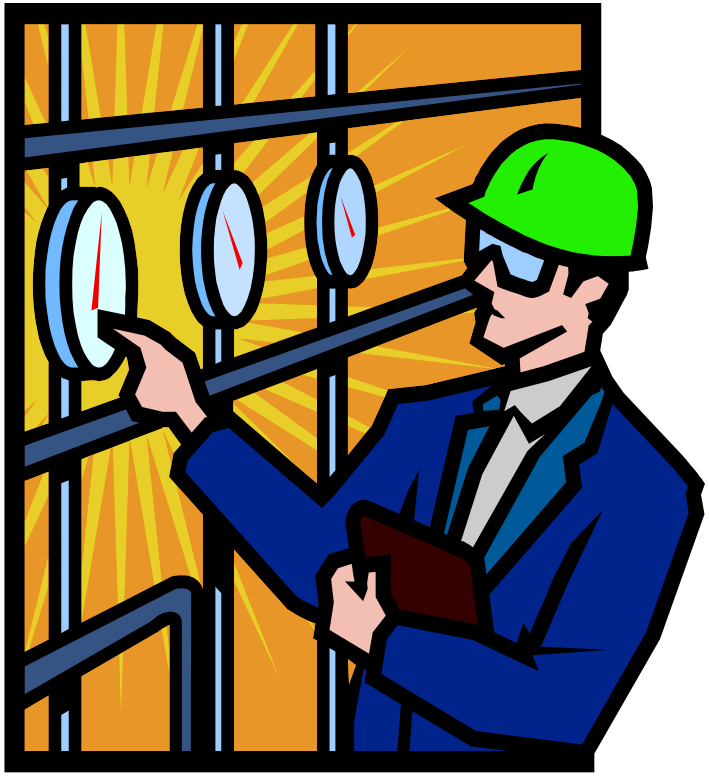
- Completeness of the Suite
- Integration of the Suite
- Secure SOA
- Standards support for both Interoperability and Portability
- Near Zero Coding™
- ✓ **Aspect-Oriented Implementation**

SOA Governance – What is a service?



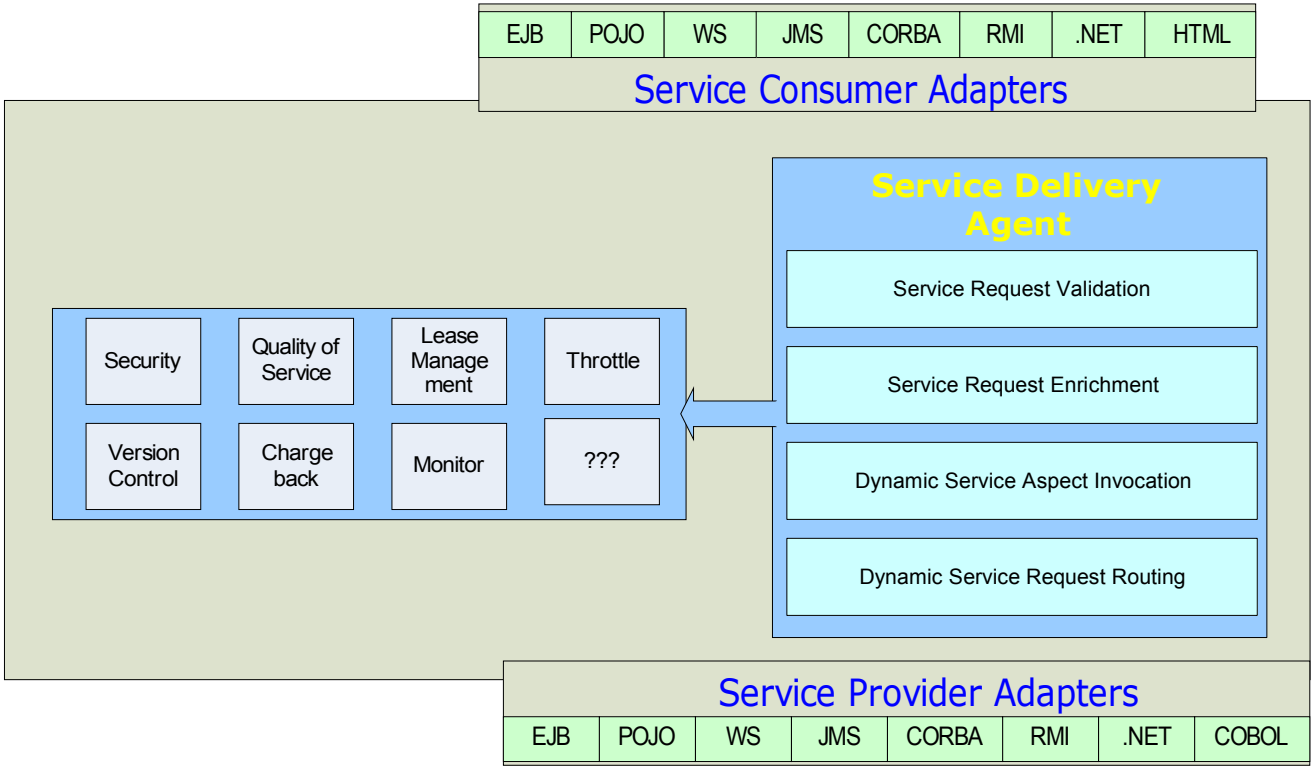
A **Service** is a self-contained component *delivering business functionality* combined with an *extensible set of non-functional, policy-driven qualities* (such as security, industry/customer defined service policies, management, monitoring, and lifecycle management) which responds to requests through a well-defined, standard, published interface.

Aspect Examples



- Security
  - SAML, X.509 certificates, username/password
  - Authentication and Authorization
  - Confidentiality and Integrity
- Monitoring and Metering
  - Minimal call path presence
  - Log service metrics
  - Create billing records
- Throttling
  - Impose throughput limits
  - Warning threshold and Enforcement threshold
- Lease and Version Enforcement
  - Enforce service expiration / sun setting
  - Route and translate based on version
  - Staging from stubs to production services

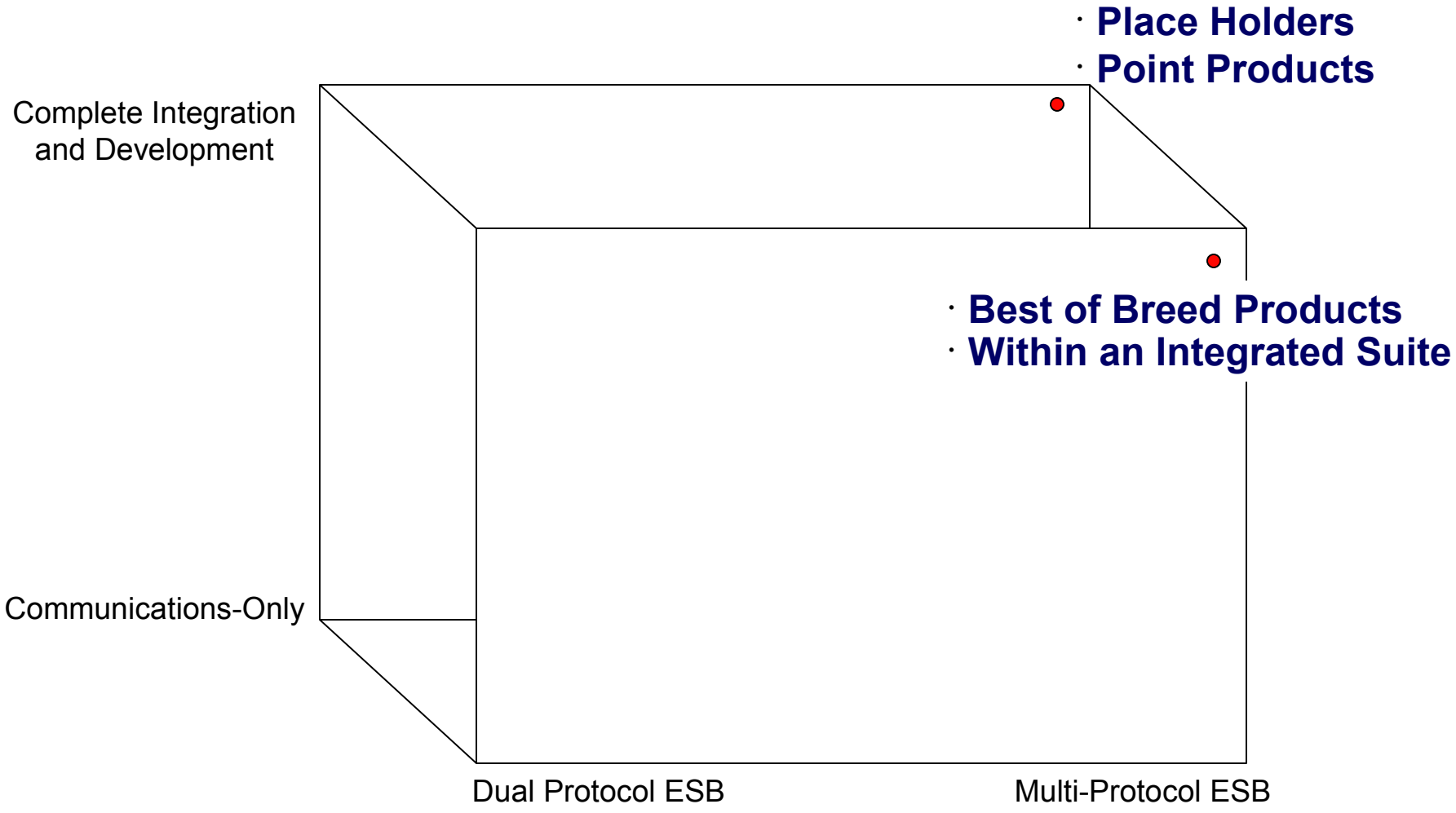
An Aspect-Oriented Governance Architecture



Future Proof SOA

- ✓ Completeness of the Suite
- ✓ Integration of the Suite
- ✓ Secure SOA
- ✓ Standards support for both Interoperability and Portability
- ✓ Near Zero Coding™
- ✓ Aspect-Oriented Implementation

Selecting technology in the ESB market?



Thank You

Ross Altman
Ross.Altman@Sun.com