

Building Sustainable Banking Architectures

Mike Walker

Microsoft® Corporation

February 2007

Revised March 2007

Applies to:

2007 Microsoft Office system

Microsoft BizTalk® Server 2006

Sustainable Banking Solutions

Summary: In this article, we will use a banking scenario to focus on some of the specific needs for financial services. This scenario will provide means for partners and customers to integrate banking applications and establish consistency in this integration. This article describes the Loan-Originiation Reference Architecture Pack. (36 printed pages)

Introduction.....	3
Enabling Technologies	3
Banking Business and Technical Challenges	4
Loan-Originiation Office Business Application	5
Technical Architecture	8
Presentation-Layer Architecture	10
Application-Services Layer	19
Services-Layer Architecture.....	23
Orchestration Architecture	27
Business-Rules Engine Architecture	28
Database Architecture	32
Enabling Scenarios.....	32
Products and Pricing.....	32
Loan Registration	34
Conclusion.....	35
References	35
About the Author	35

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in examples herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2008 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, BizTalk Server, Microsoft Office Access, Microsoft Office Excel, Microsoft Office InfoPath, Microsoft Office Outlook, Microsoft Office PowerPoint, Microsoft Office SharePoint Server, SQL Server, and Microsoft Windows Presentation Foundation are trademarks of the Microsoft group of companies.

All other trademarks are property of their respective owners.

Introduction

This article will show architects how to design sustainable banking solutions using the latest Microsoft® technologies. We combine a complex business scenario with many of the technology challenges in banking to make a real-world example of the relevant issues in the banking landscape. To make this more relevant, we introduce a consumer loan-origination scenario. The scenario addresses enterprise concerns around interoperability, workflow, and agility needs in this segment of the banking industry.

The article includes comprehensive guidance for making architecture decisions around real-world business and technology challenges. We will be breaking up the architecture into separate concerns that will help identify its various aspects. These different concerns will include reviewing the security, logical, physical, communication, and information architectures. It will also help you identify opportunities in your business for other areas in which these technologies can address your needs.

Building feature-rich, composite-style applications is the goal of Office Business Applications (OBAs). Using this technology, financial services organizations can migrate to a service-oriented architecture (SOA) with more ease than ever before. This article discusses both integration and composability issues between applications and several different ways to accomplish them.

The first and most mature way to integrate is by focusing on data integration. However, this approach leaves much to be desired because you lose the behavior that goes with the data. The introduction of service orientation and XML-based standards have created a second and much more useful method of integration that allows for applications to be integrated, while preserving behaviors and data to bring together disparate business logic through a standardized service facade. However, because there are many ways to provide a service that exposes application functionality—as well as many ways to implement the XML standards used—integration at the service level can become cumbersome because of a lack of consistency.

Enabling Technologies

This section describes the enabling technologies used in our scenario. The primary technologies used are Microsoft Office SharePoint® Server 2007, Microsoft BizTalk® Server 2006, and Microsoft SQL Server® 2005. The capabilities exposed here will help you to:

- Enable business processes through a standardized application services tier.
- Create a scalable messaging architecture to expose standardized Web services and maximize interoperability inside your organization.
- Expose Office Business Applications (OBAs) through these reusable services layers.

We will cover these technologies in depth as we describe the architecture. The following technologies are used in our reference architecture:

- **Microsoft BizTalk Server 2006.** By using BizTalk Server, banks can now orchestrate their business processes in a dynamic and fluid way.
- **Microsoft .NET Framework 3.0.** This includes Microsoft Windows® Presentation Foundation (WPF), Windows Communication Foundation (WCF), Windows Workflow Foundation (WF):

- Windows Presentation Foundation (WPF) can reduce the training time of loan executives, officers, underwriters, and secondary marketing personnel. This is increasingly important in high-turnover areas.
- Windows Communication Foundation (WCF) will reduce the increased complexities of the integration needs. For example, it will cover integrating broker, correspondent bank, flood, mortgage insurance (MI), appraisal, credit, verification of funds, and cross-sell services.
- Windows Workflow Foundation (WF) provides banks the flexibility to create workflows around document types such as Microsoft Office Excel® 2007, Microsoft Office Word 2007, and Microsoft Office PowerPoint® 2007.
- **Microsoft Office SharePoint Server 2007.** This is used for Microsoft Office SharePoint Server 2007, Office Excel 2007, and documents:
 - Office SharePoint Server 2007 can host your loan documents in a centralized, versioned, and secure environment, while also adding functionality, such as approval workflows and communications using Office Communicator 2005.
 - Office Excel 2007 can host your Office Excel data in a centralized SQL Server information store. This robust relational database provides data protection, centralized backup procedures, a single version of the truth, and security around sensitive data. This also provides the Office Excel user experience for multiple services without cross-training on other applications.
 - Document formats have been changed to Open XML, an open document format. This ensures cross-platform compatibility, robust integration, document manipulation opportunities, and controlled data integrity checks from within the documents themselves.
- **Smart Client.** Having disconnected systems for the mobile workforce is becoming increasingly important for banks.
- **Windows Mobile.** Mobile devices can be leveraged in the field for appraisal visits and zoning.
- **Microsoft SQL Server 2005.** This is the repository of choice for storing all the sensitive loan data needed for this solution. Additionally, SQL Reporting Services and Analytical Services will provide robust Web parts to plug into our Office SharePoint Server 2007 portal.

Banking Business and Technical Challenges

In the context of composite applications, there are quite a few challenges for financial services. Banks and insurance companies are starting to push more and more services to the edge. From a consumer perspective, this includes services such as ordering stamps at an automatic teller machine (ATM).

For this paper, the focus will be on a specific line of business (LOB): the consumer loan. Within the consumer loan, there are multiple channels in which business enters into the process and many stages in those channels. There are four major channels that are common in banks today:

- **Wholesale.** This is the channel in which brokers interact with the lender. Brokers have the option to use multiple lenders. It is critical for lenders to provide a differentiating experience and service to brokers for a competitive advantage.

- **Correspondent.** Smaller lenders can interact with larger lenders through this channel to book loans that they might not have the funds to book independently.
- **Telesales.** Sales of loans occur through the Telesales channel.
- **Retail.** Direct lender-to-consumer interaction occurs through this channel.

For simplicity, we will cover only the Wholesale channel in this article. There are some commonalities among the channels, but there is also exclusive behavior. We have built the reference architecture to support the primary capabilities of each channel.

Loan-Originiation Office Business Application

In the loan-origination business, there are many business drivers for banks: process consolidation, regulatory compliance, and faster product delivery (loan completion). Loan products change frequently and are usually dynamic, based on location (regional or state). Developing and modifying products in an agile manner enables banks to be highly competitive and adaptable in key markets. Also, compiling and staying on top of regulatory laws is always a challenge, given the turbulent changes happening in the industry today. Laws such as the Community Reinvestment Act (CRA), Home Mortgage Disclosure Act (HMDA), and anti-predatory lending often require cumbersome integration issues and data-mining exercises. The complexities of multiple processes for regions, tiered pricing, or special market demands complicate the overall process. Banks are now trying to consolidate processes to reduce this complexity and maximize value. Not only are banks trying to reduce the amount of processes, but they are also trying to reduce what is called the "application-to-delivery cycle." By reducing this window, banks are seeing savings in the range of three basis points and closing more loans per month.

The scenario application demonstrates alignment with key business processes within the loan origination. Because we are taking only a subset of loan origination and focusing on one channel, there will be some processes missing. It is also important to note that the reference architecture will stop at document preparation and closing. The channels highlighted are the following (see Figure 1):

- **Products and pricing.** The process in which personnel at the lender will analyze the secondary market and apply the proper pricing to the products (for example, 30-year fixed) or their portfolio products (products that the bank funds and usually does not sell to other banks or the secondary market).
- **Application or registration.** This is when the customer enters the proper qualification information for a loan product.
- **Processing or locking loan.** This is when a customer agrees to a specific product and rate and "locks into a commitment." This stage kicks off many subprocesses that gather extended information on the customer to prepare the underwriters to make a decision on the loan.
- **Underwriting.** The process in which an underwriter makes a decision on the loan.

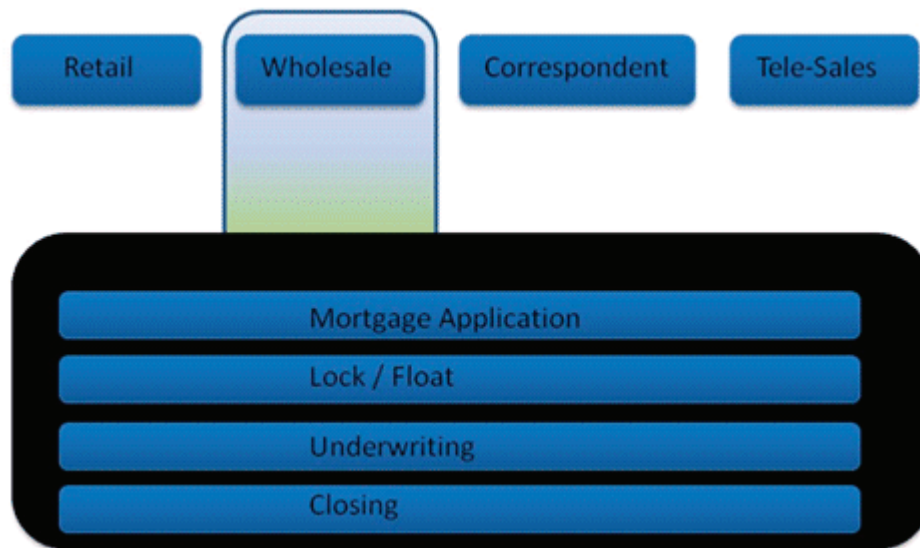


Figure 1. Business channels

With this solution, it is important to understand the amount of complexity both in the business processes and the current-state technological landscape. The technological landscape will be defined in the Technical Architecture section that follows.

Defining the business architecture will aid in the value proposition to banks and other enterprises that deploy similar solutions. This business architecture will define the core process areas and model them to derive the proper technical architecture.

When architecting solutions, keep in mind that the technology components are just one of the pieces. There are also human elements when architecting solutions. So, not only do we consider the system-level processes, but we also consider the human or manual processes, in order to give the architect the complete perspective. This is one of the major benefits of composite applications.

Understanding loan origination is not easy because of its complexities. However, comprehending some core concepts of loan origination will make things easier. These include:

- Definition of products and corresponding pricing.
- Long-running workflow.
- Approval processes.
- Calculations.

Figure 2 is a representation of the business capabilities. These are not independent functions. There are interwoven dependencies between each of the business functions.

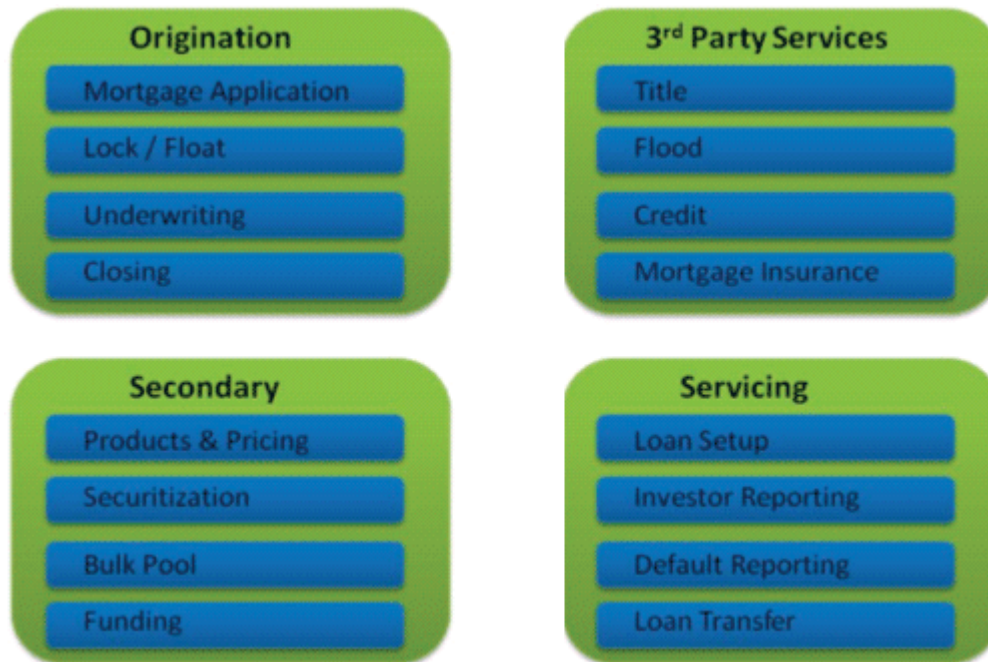


Figure 2. Business capabilities

Briefly, these capabilities are described as follows:

- **Origination.** The process of acquiring the customer and the corresponding data, making decisions based on processing, and requesting data from third-party services.
- **Third-party services.** Used to pull data on the customer and the property being purchased.
- **Secondary processes.** Processes that happen at varying times in the process. For the scope of this solution, we will focus on pricing for the various mortgage products (such as 30-year fixed).
- **Servicing.** After origination, loans are either booked on the bank or sold on the secondary market (other banks, Fannie Mae, Freddie Mac, and so forth).

These might seem like simple processes on the surface; however, there are several business challenges here. Earlier core concepts were described (long-running workflow, approval, and so forth). These core concepts are very complex. The goal is for this architecture to aid in the simplification or elimination of these issues. The following are some of the challenges that the loan industry faces:

- Workflows are complex and difficult to manage. These are often based on exception processes, particularly portfolio loans, and region. Typically, there will be business logic embedded in each of the processes.
- Fragmented processes and little automation.
- Redundant systems supporting business process.

Technical Architecture

Now that we understand the business architecture, we will describe the technical architecture in each logical tier of the solution (this is shown in Figure 3):

- **Presentation layer.** Serves as the user interface. We built this using ASP.NET 3.0 Web Forms hosted on the Office SharePoint Server portal. Office SharePoint will provide the underpinnings for the application. There will be several services that can be inherited from this environment—specifically, the portal architecture that will be required to deploy Web Parts for this solution.
- **Application Services layer.** A reusable layer in the architecture. This will allow applications to use functionality such as Digital Rights Management, Document libraries, workflow, and so forth.
- **Services layer.** Provides an infrastructure to communicate messages. The Office SharePoint Server 2007 layer will use WCF. The integration layer will use BizTalk Server.
- **Business Rules layer.** Centralized business rules to build consistency, reliability, efficiency, and cost reduction of system architectures. BizTalk Server has a built-in Business Rules Engine (BRE) that we will use.
- **Orchestration layer.** Process development and management occurs in this layer. BizTalk Server also has a robust orchestration engine for Business Activity Monitoring (BAM) types of activities. The architecture uses Business Process Execution Language (BPEL) extensively throughout to ensure interoperability between other orchestration systems.
- **Data Services layer.** Relational-database services and management occur in this layer.

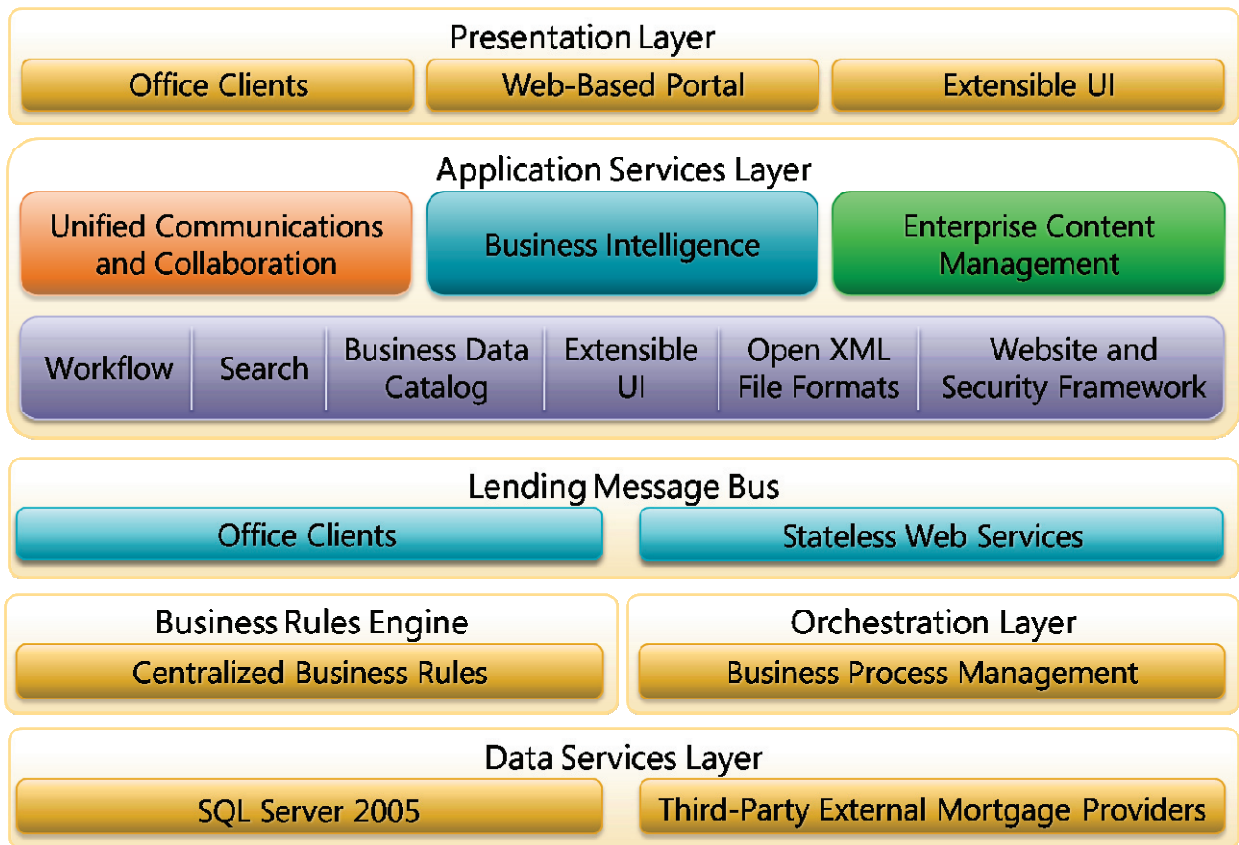


Figure 3. Logical architecture

There are many layers to this architecture. One might think that it is overly complex. However, it is really quite simple, compared to how processing is done today. The same system would be multiple systems dollied together using complex point-to-point integrations with even more complex and fragile monitoring systems.

A common question is, "Why is there a separation between the business-rules and the orchestration layers?" This separation is not always necessary, but in complex situations such as this scenario, in which business processes change frequently and rules are somewhat static, this scenario is ideal. By separating out these two layers, you achieve:

- **Agility.** The ability to have a base set of rules that can be used in different ways (that is, orchestrations) without development.
- **Reduction of code complexity.** Now, business analysts or developers can model the process based on rules developed and versioned in BizTalk Server.
- **Reusability.** In loan origination, there often are multiple loan-origination systems (LOS) in a lender's IT enterprise. Lenders are looking to consolidate and leverage a central system.

Presentation-Layer Architecture

With today's technologies, there are many options for user interfaces: Web 2.0, Smart Clients, Office applications, and Web portals. In this architecture, we will use the underlining Office SharePoint Server 2007 portal technologies. Office SharePoint Server 2007 portal will provide a base set of "application services" for our user interface. This will facilitate the core components of the user interface, such as display, navigation, composition, and security.

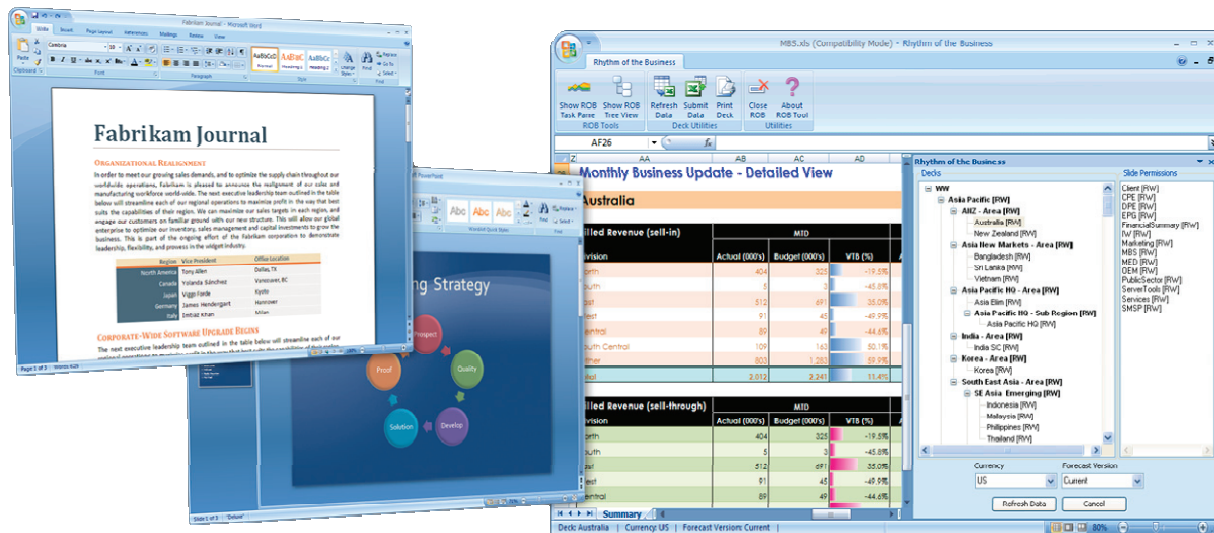


Figure 4. Many choices for user interfaces

The goal is to give users the best possible user experience. This may consist of giving them the tools that they know very well or the tools that provide the functions that are required for the utmost productivity or ease.

When you have a dynamic environment such as this, considering the right architecture for the job can be challenging. Designing this reference architecture for the Wholesale or Broker channel, we have to look at the end user who will consume this solution. For brokers who acquire services from many lenders and who have their own point-of-sale (POS) systems, such as Byte or Calyx, a Web-based portal solution is ideal.

For security and scalability reasons, the presentation and business-logic layers are separated.

Examples of how this can benefit financial services organizations include the ability to:

- Customize themes and logos.
- Display messages to users on specific pages for many reasons, such as marketing, training, or cross-selling products or services.
- Inherit rich navigation features provided by the Office SharePoint Server 2007 portal platform.
- Design user interfaces to manage and version content within the application.

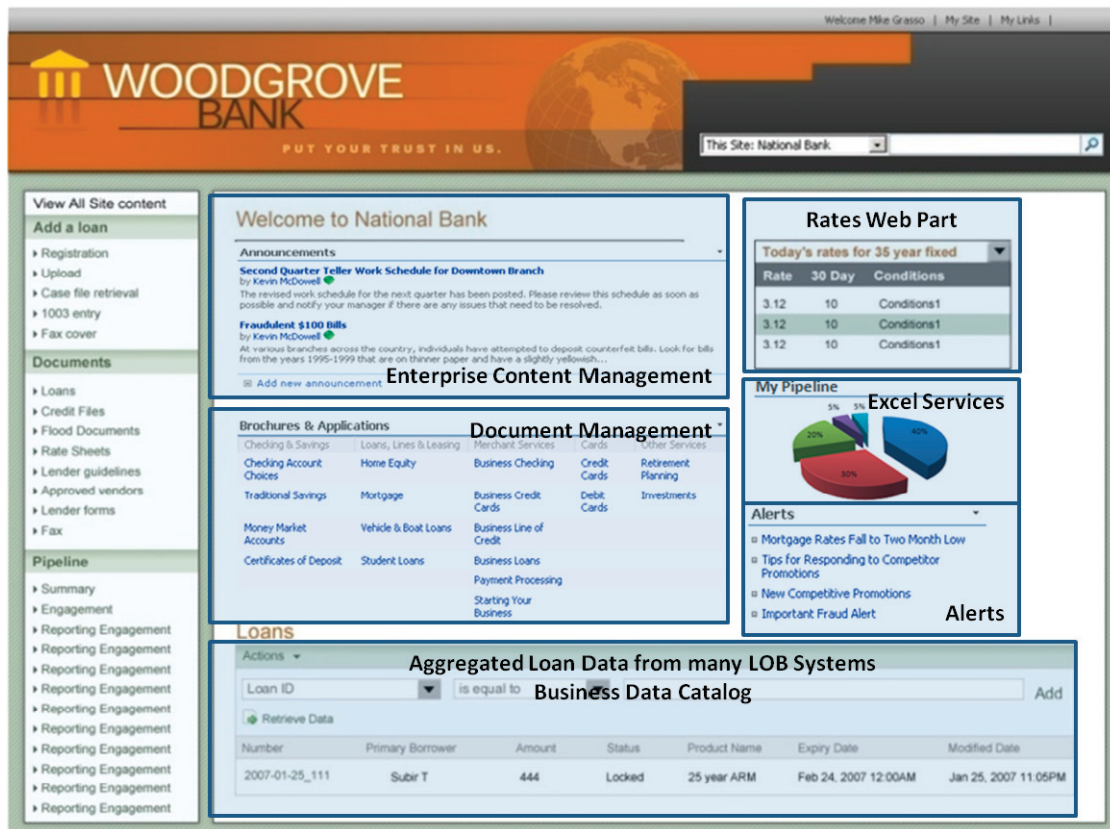


Figure 5. Office SharePoint Server user interface

The greater portion of this page design comes from the use of Web Parts, befitting the composite architecture of the application. Office SharePoint Server will host all of our Web Parts, giving them the necessary cohesion.

Team Sites

Team sites are one of the core capabilities that we want to expose using Office SharePoint Server 2007 as our underlining presentation. This will manage the communication within and across teams by providing discussion boards, issue management, task management, and scheduling.

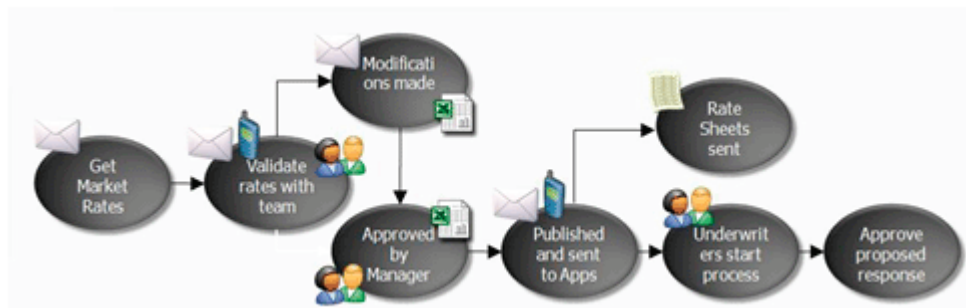


Figure 6. Collaborative behavior exposed in Office SharePoint Server 2007

Several different personas will use these features within our teams. As shown in Figure 6, for example, the secondary marketing personnel will use document libraries to publish rate sheets. This, in turn, will trigger alerts to the underwriters. Underwriters will receive alerts to their "My Site" and to the Microsoft Office Outlook® 2007 client. A *My Site* is a user's personal portal site in which the user can subscribe to discussions and libraries to create their own content.

We are not limited to using just Office SharePoint Server 2007 for team sites. Team sites can be hosted within other LOB applications within your enterprise. Extending these composable parts into other contexts will give our users a much richer user experience and increased productivity.

Web Parts

Deciding what Web Parts to use or develop can be a challenge in itself. You must evaluate your specific needs and choose the best option for your technical issue. The following are the three core Microsoft user interface (UI) technologies that are available:

- **Windows Presentation Foundation (WPF).** Used for rich user interfaces. Similarities with other presentation frameworks would include Adobe Flash. WPF was ruled out for this scenario, because this solution did not require such a rich presentation. However, it could be a fit if organizations want a rich interface.
- **ASP.NET Web Forms.** The .NET form of Web pages, also referred to as ".aspx pages." Web Forms were chosen initially; however, after careful consideration, Microsoft Office InfoPath® 2007 was chosen. Considering the strong design-time XML translation of schemas in Office InfoPath 2007, it provided simplification of the architecture, which reduced the amount of code required for developers by leveraging the Mortgage Industry Standards Maintenance Organization (MISMO) schemas already defined.
- **Office InfoPath 2007 Forms.** This type of UI can be extended to an Office InfoPath 2007 client or exposed as a Web Part on Office SharePoint Server portal. Office InfoPath 2007 is great at form-based user activity. It has the ability to consume a schema (XSD) and generate a UI with the appropriate UI rules.

To help with the decision, we developed a comparison chart to aid in the technology selection.

Table 1. Comparison of technologies

Criteria	WPF	Web Forms	Office InfoPath Forms
User experience	Dynamic next-generation user experience	Application-centric	User entry-centric, form-centric
Content control	Allows customized or fine-grained control over page elements	Allows customized or fine-grained control over page elements	Dynamic UI made possible by creating multiple views of the same form
Page flow and navigation	Need custom code to handle navigation	Need custom code to handle navigation	Need custom code to handle navigation
Development environment	Expression Design Suite	Microsoft Office SharePoint Designer/Microsoft Visual Studio	Office InfoPath Client or Microsoft Visual Studio 2005 Tools for the Microsoft Office System
Integration with Office SharePoint		Need to add pages to Office SharePoint sites Need custom code to persist data from the page into a document library	Need to deploy forms to an Office SharePoint form library; after filling in a form, data can be automatically stored in the form library as XML
Validation of page data when user enters value	Need custom code to validate fields	Need custom code to validate fields	During form design, validation can be added to fields with minimal code; at runtime, the fields are validated without requiring post-back to the server
XML integration	Need custom code to validate fields	Need custom code to map XML schema to Web page fields	Native support for binding XML schema to page fields
Reusability	By using the XAML language to describe UI, extending and reusing interfaces can be achieved	Can be used in a Web client only	The forms can be used in an offline scenario or embedded within e-mail
Platform	Built on .NET 3.0 Technologies that are based on the .NET 2.0 Framework	Built on ASP.NET 2.0	Built using ASP.NET 2.0 and Forms Services, which is part of MOSS

Using the Office InfoPath Web Parts, you can integrate into the Office SharePoint Server Web Parts framework in a more appealing way. Instead of coding every detail, Office InfoPath can automatically supply much of the user input and navigation aspects. Additionally, all the data is stored in XML, with all the user data in an industry-standard format called the Mortgage Industry Standards Maintenance Organization (MISMO). This will be the format of the loan documents on the document libraries, which will make the data much more consumable by other processes in our architecture.

Coupling Application Services with Presentation-Layer Architecture

Referenced all through the presentation-layer architecture is the use of the applications-services layer. This layer, as shown in Figure 7, uses Office SharePoint Server 2007 technologies. This reusable layer will provide the necessary framework to integrate our presentation layer.

The nice part about Office SharePoint Server 2007 is that there is not much integration involved. There is a clear separation, without tightly coupling the different layers. We will use this for the portal layer, but we will use it also for the other clients that are used in this architecture.

This composite application architecture will allow for the unification of multiple disparate LOB applications. This allows for multiple views into the lending business. Through the OBA architecture, we can get a high level of extensibility by integrating other systems that are based on Office Excel, custom .aspx Web pages, and data from other systems.

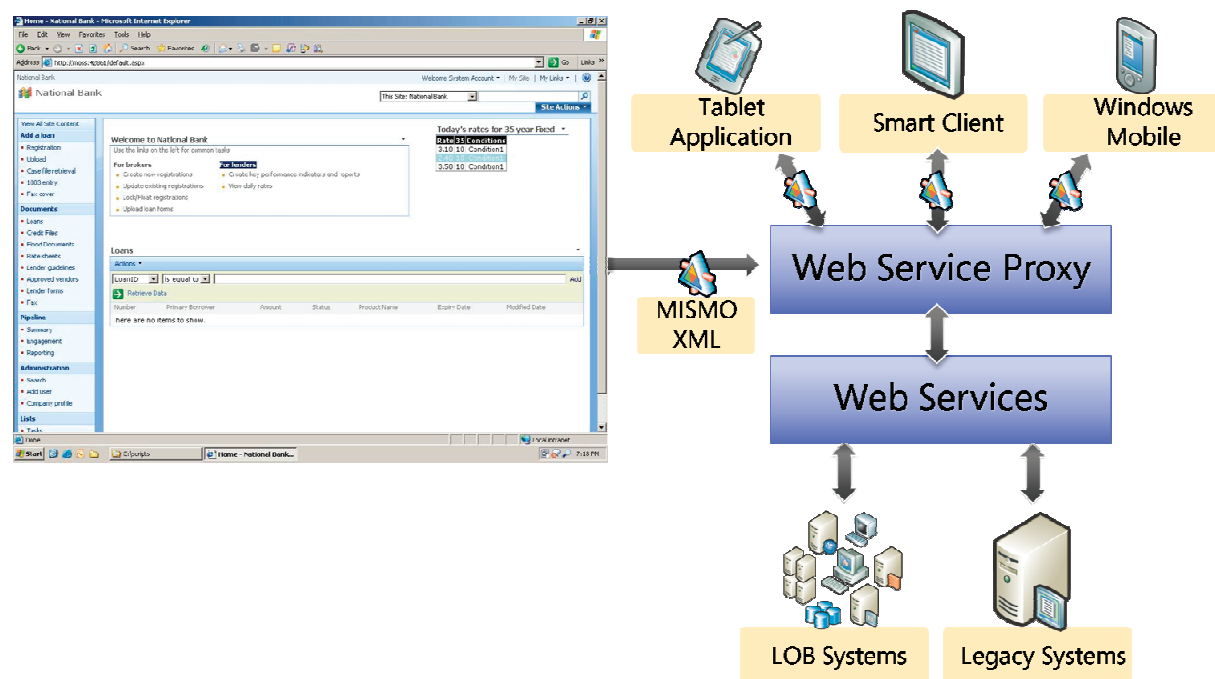


Figure 7. Integrating LOB solutions with Office SharePoint Server through standard Web services

As shown in Figure 7, we can augment and/or wrap existing IT assets using the Office SharePoint Server technologies, making Office SharePoint Server truly a reusable application-service layer. As referenced in the above business section, the loan LOB is complex and LOS systems have proliferated within an organization. Office SharePoint Server can bring some order to this chaos by unifying workflow, communication, and business logic. In addition, where legacy systems are used, Office SharePoint Server coupled with BizTalk Server can provide rich interoperability between these platforms.

The usage in our scenario is most appealing to lenders because it could have a substantial impact to their business. Within this portal environment, we can plug in other systems that allow us to cross-sell other services. For example, in loan origination, cross-selling services such as checking and savings accounts can generate quite a bit of revenue. In addition, it can lower the risk of the loan portfolio by allowing lenders to get a better look at the financials of an individual.

Using the Business Data Catalog

The Business Data Catalog (BDC) is a new framework that provides Office SharePoint Server 2007 portal sites and standard Windows SharePoint Services 3.0 sites with integration into back-end LOB systems. Additionally, the BDC provides the means to integrate data directly from database systems, such as SQL

Server, Oracle, or DB2. It was possible to integrate portal sites with back-end systems previously. However, it required you to write custom code to manage connections and retrieve the data that you needed to display. The BDC will make it much easier for developers to retrieve data from back-end systems or applications.

The BDC accomplishes this because its design is based on standardized metadata that describes the location and format of a back-end system and data entities defined within it. The BDC also provides an execution component that is capable of reading BDC metadata and retrieving external data from back-end systems, then returning that data to Office SharePoint Server in a standardized format.

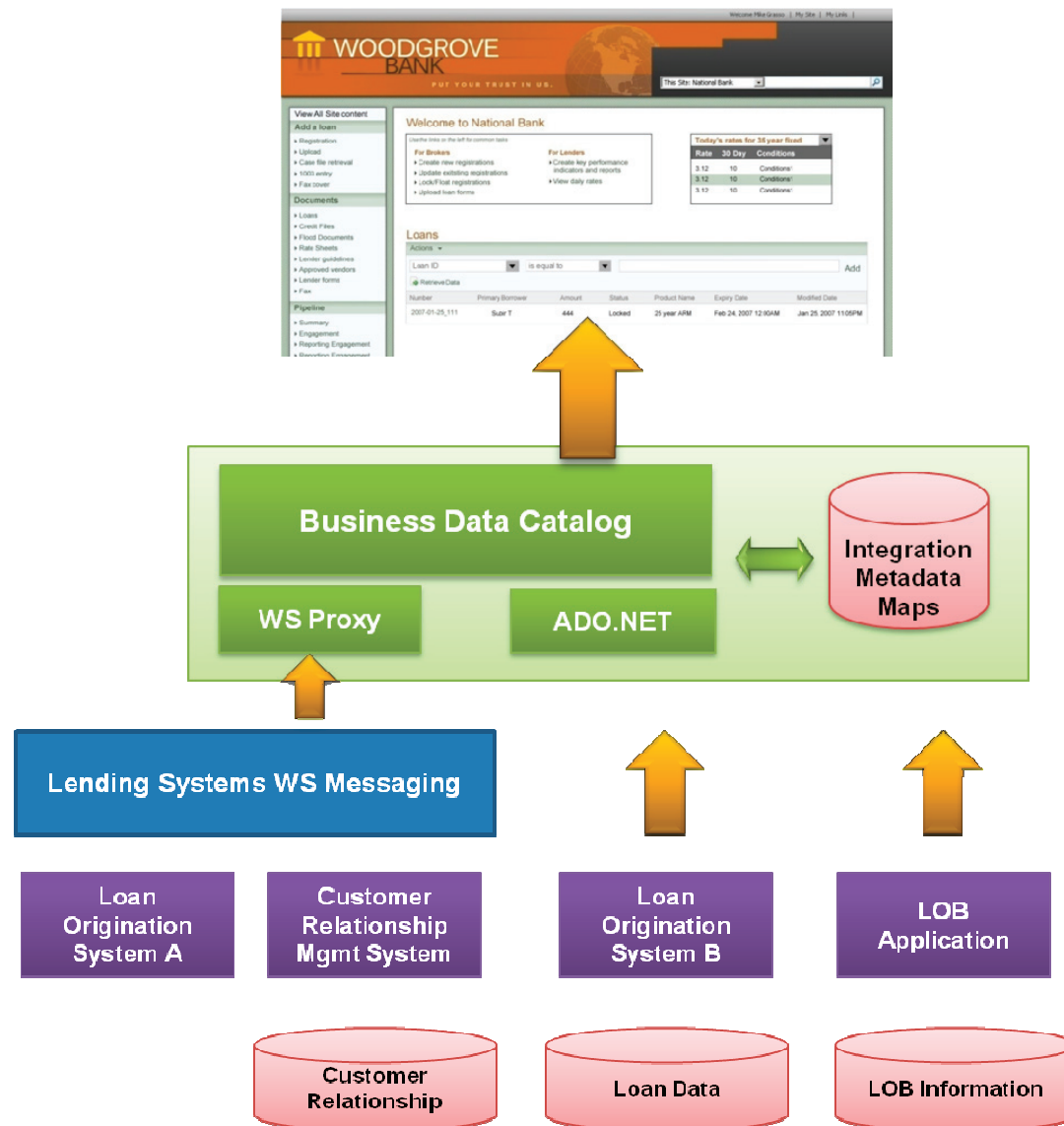


Figure 8. Pipeline Web Part using the BDC

We will use the BDC to expose our pipeline data on the home page of the loan-origination solution. We chose this Web Part over a standard data Web Part for the following reasons:

- **Reusability.** We can leverage the BDC to expose common pipeline data that will be used for several different purposes.
- **Extracting data from other systems.** In scenarios in which there is more than one LOS, the BDC provides an easy way to unite different data sources.
- **Performance.** Eliminating multiple layers of architecture will improve performance.

Extending to Microsoft Office Clients

Training users on new applications can be timely and costly. In contrast, clients of this system will all be very familiar with the 2007 Microsoft Office system client tools, such as Office Excel 2007, Office Word 2007, and Office Outlook2007. These tools will integrate in various ways.

The architecture around using 2007 Office system client tools is similar as a Smart Client application. With the new features of the 2007 Office system client tools, there are several extensibility points.

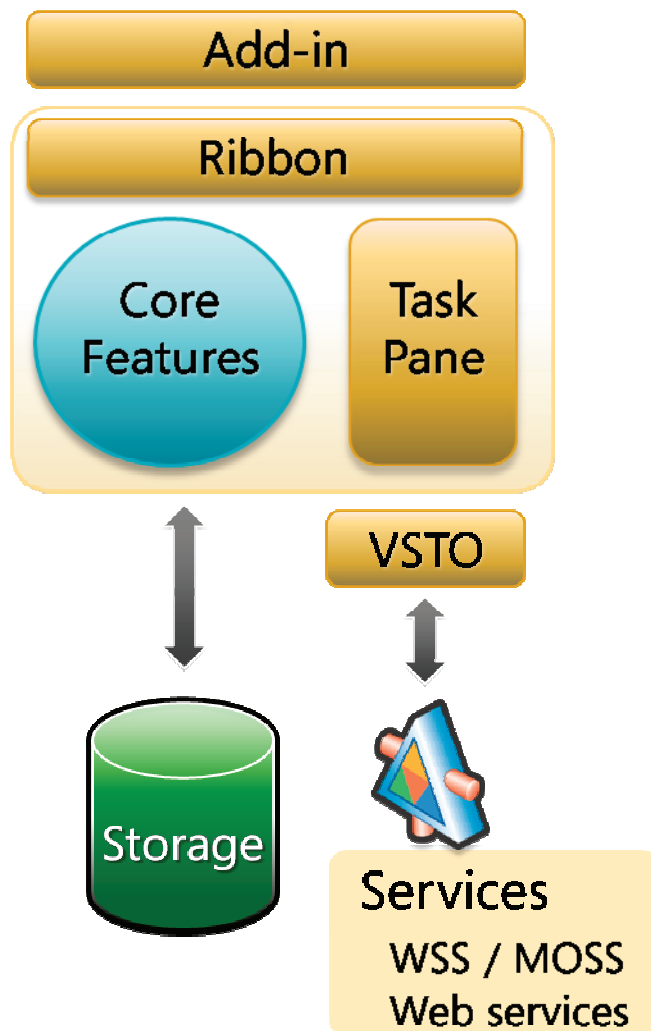


Figure 9. Extensibility points in 2007 Microsoft Office system clients

Figure 9 shows several points of extensibility in the 2007 Office system clients, such as Microsoft Office Access™ 2007, Office Excel 2007, Office InfoPath 2007, Office Outlook2007, Office PowerPoint 2007, and Office Word 2007. For example, the ribbon that is the new toolbar for 2007 Office system clients will allow for custom add-ins, so that within our applications we can build context-driven Office ribbons to increase productivity within our solutions.

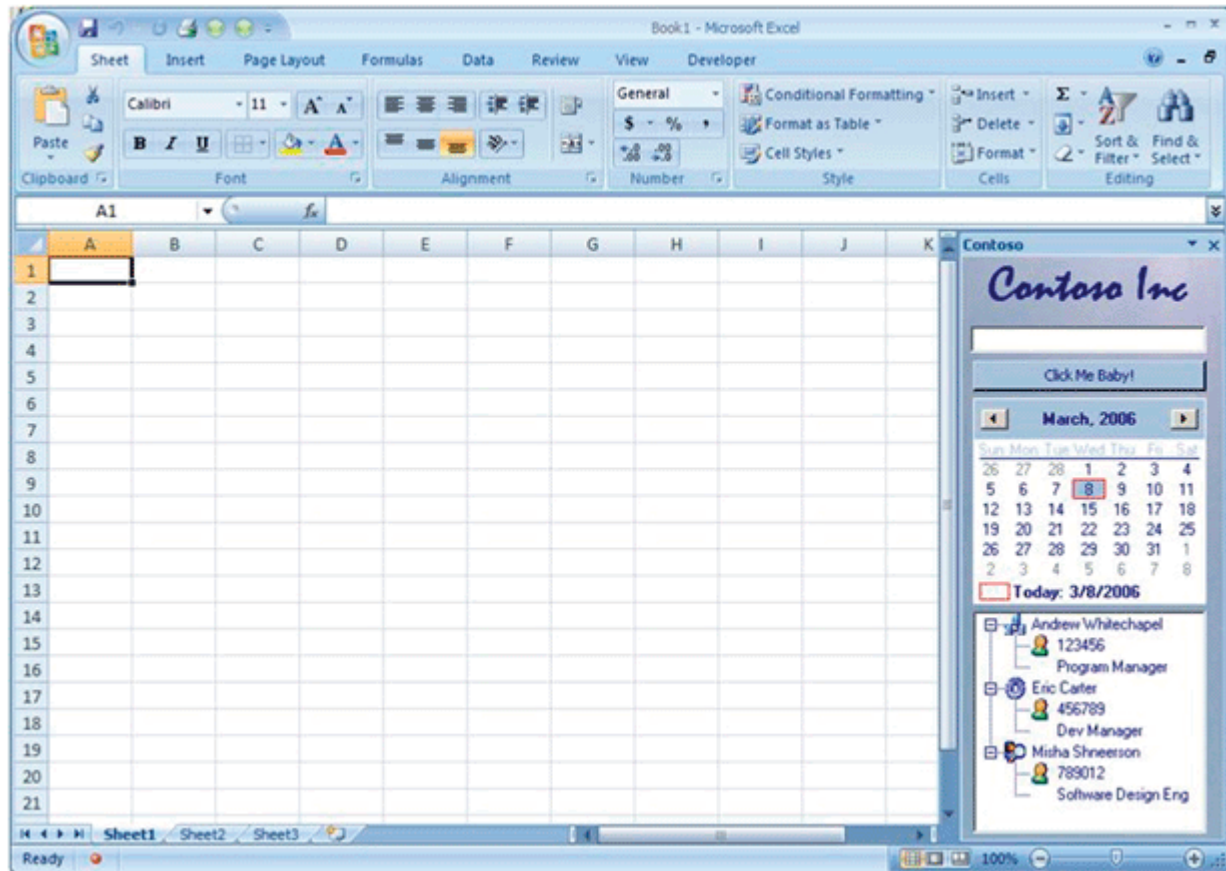


Figure 10. Office Excel example of a task-pane extension

Another area of extensibility is building out custom task panes. As with ribbons, when building out task panes, developers can create Microsoft Visual Studio 2005 Tools for the Microsoft Office System custom add-ins either as managed or unmanaged code.

When we look at the end business user, the key is to use the right tool for the right job. At times, architects struggle with choosing the ideal interface for users to consume. At the end of the day, architects should choose the interfaces that provide the richest user experience to the user. Sometimes, that is a Web-based application, such as we are using for our brokers. Other times, a smart-client application like Office Outlook will be used. The tools that we choose for the varying personas are the following:

- **Office Outlook.** Used to send tasks to the underwriters, processors, and secondary marketing personnel.
- **Office Excel.** Used to generate rates and enter them into the system (see Figure 6).
- **Office Word.** Used to read generated documentation, integrate with document libraries, and publish content and communiqués to internal staff and brokers.

- **Office SharePoint Server.** Provides a scalable and efficient records-management system. The Records Repository acts as the hub for all record-management processes, including content collection (spreadsheets, documents, e-mail, and non-digital items), policy enforcement, item retention in response to external events, and content disposal.

As shown in Figure 11, Office Excel will be used for various tasks. The primary use of Office Excel will be used by secondary marketing users. The reason for using Office Excel here is its rich data-manipulation capabilities. Additionally, in most loan departments, Office Excel has been the tool of choice for many years.

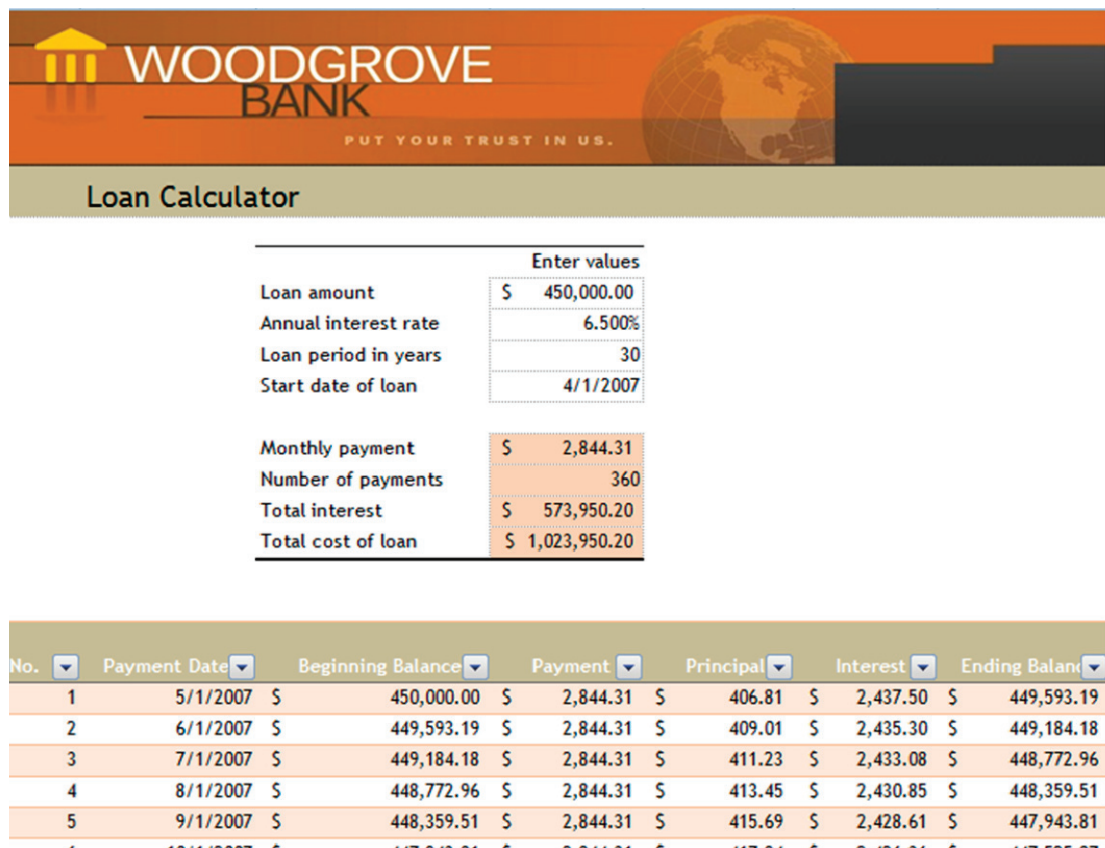


Figure 11. Using Office Excel as a client

Office Word also will be used to help with document generation. One of the more challenging areas in the scenario is generating the necessary documents to close a loan. It is not as simple as pulling data from a database and composing an Office Word document. There are complex business rules that need to be applied and that are based on several parameters, such as the following:

- **Regulatory.** Text such as disclaimers (for example, HMDA)
- **Geographic concerns.** Certain disclaimers for differing state laws
- **Building classes.** For example, low-income housing
- **Portfolio loan disclaimers and rules**

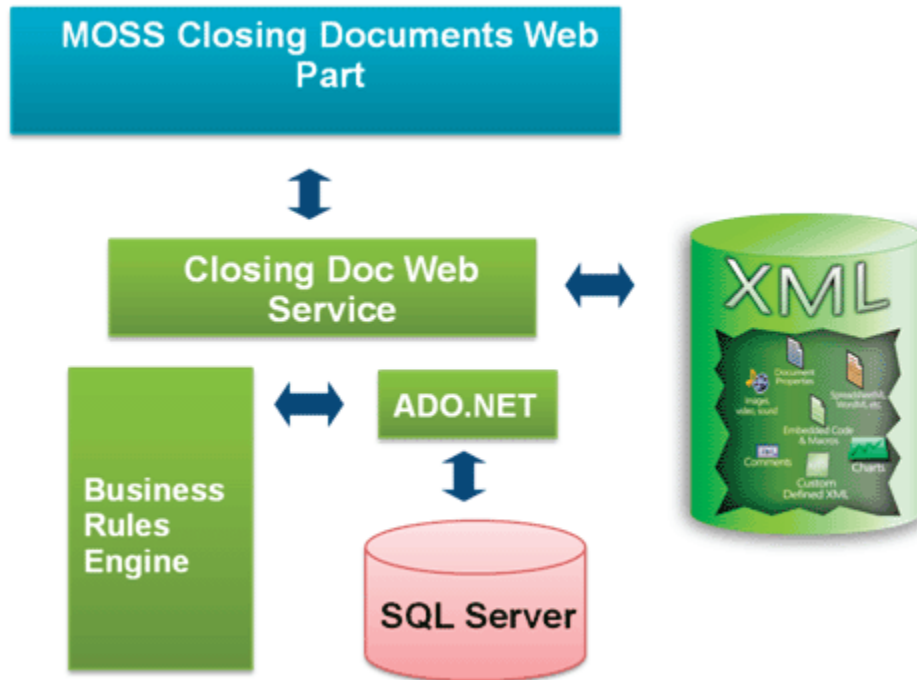


Figure 12. Business-rules architecture for delivering closing documents

Using our Business-Rules Engine, we can compose Office Word documents using the new 2007 Office system formats. The new formats conform to the Open Document XML standards. This will provide additional extensibility to our architecture. Not only can the 2007 Office system open these documents, but other applications also can read these documents. This is important with loan closing documents, because the documents are sent out to other companies to print and get signatures.

Application-Services Layer

Throughout this paper, we have been referencing the Office SharePoint Server architecture. It is really several logical layers (see Figure 14). This consists of the following:

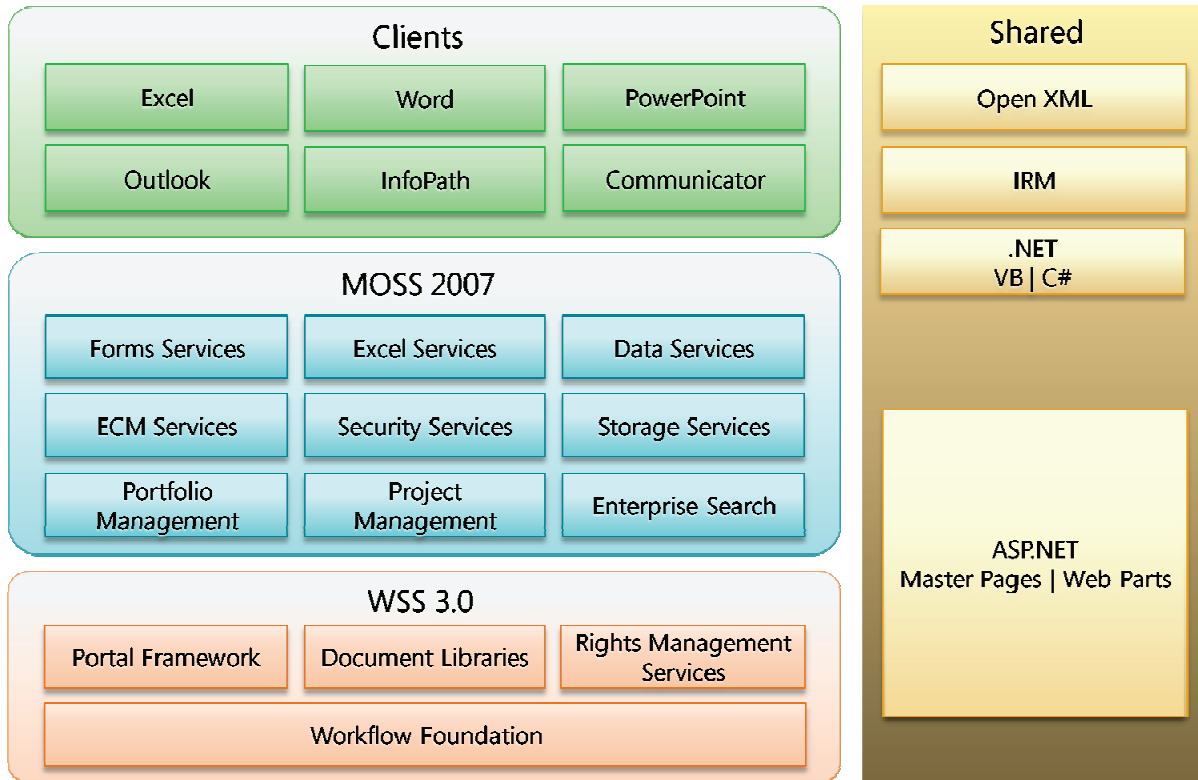


Figure 13. The Office SharePoint Server layers

The Office SharePoint Server platform provides a rich application-services layer that enables many reusable services for your enterprise's SOA. These services include, but are not limited to:

- **Enterprise Content Management (ECM).** Can be leveraged to publish dynamic and user-manageable content.
- **Workflow.** Uses Windows Workflow Foundation (WF).
- **Composable portal platform.** A framework for building rich, composite Web-based user interfaces.
- **Web Parts enabled through the portal platform.** There are several out-of-the-box parts for both Office SharePoint and SQL Server Reporting and Analysis Services.
- **Digital Rights Management (DRM).** To control who can view, distribute, or print content.
- **Document storage facilities.** To aid in versioning and document-auditing capabilities.
- **Business Data Catalog (BDC).** To expose LOB data to the presentation layer.

The following built-in Office SharePoint Server services (see Figure 15) are used in this solution:

- **Portal framework.** Has the ability to provide a composite architecture that exposes Web Parts for discrete pieces of functionality.
- **Content management.** Used to control the content that is exposed to the users.
- **Office Excel Web Services.** Used for all calculations.
- **Document libraries.** Provides management functionality for auditing and versioning.

- **Windows Workflow Foundation (WF)**. Used for orchestrating the business process.

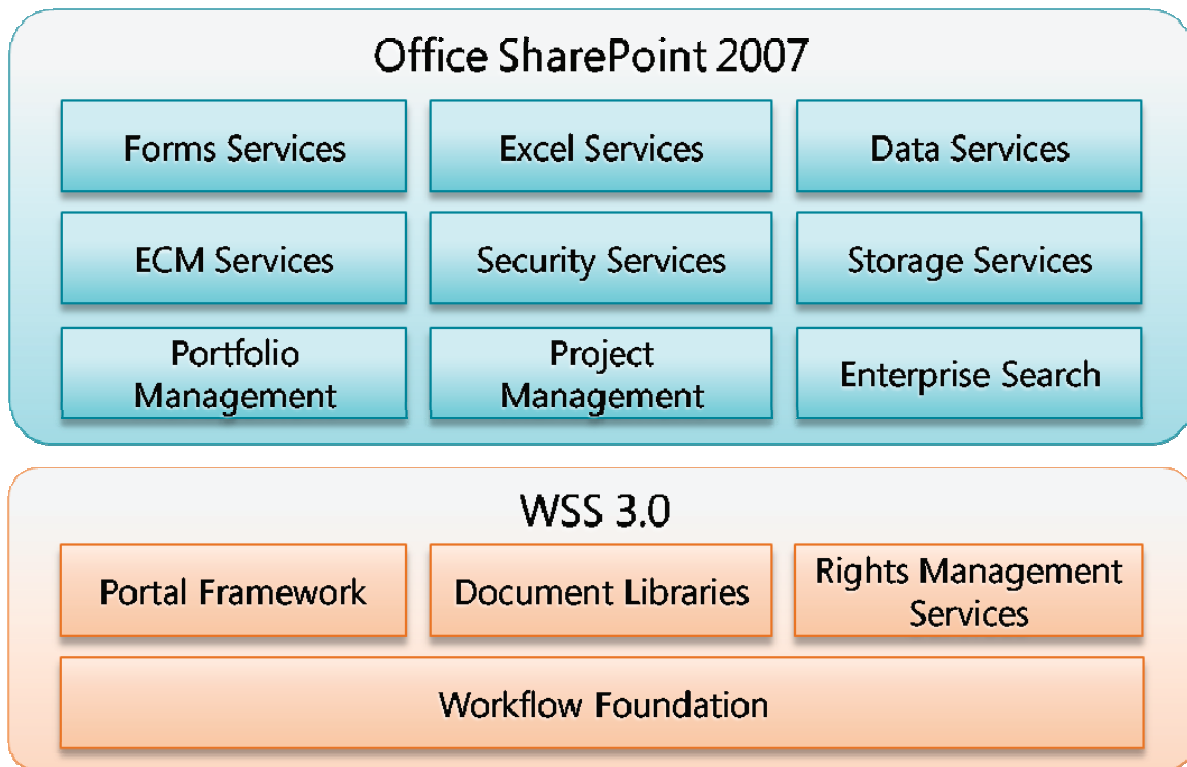


Figure 14. Application-services implementation reference

WF and WCF Design Considerations

In this solution, you will notice that there are two workflows used; one is contained within Office SharePoint Server and one is contained in BizTalk Server. The built-in workflow used in Office SharePoint Server is WF. This will be used to manage the user-centric aspects of the architecture and govern the overall process A-to-Z.

The workflow in BizTalk Server is used to manage workflow for integration, as shown in Figure 17. We will cover this in more detail later.

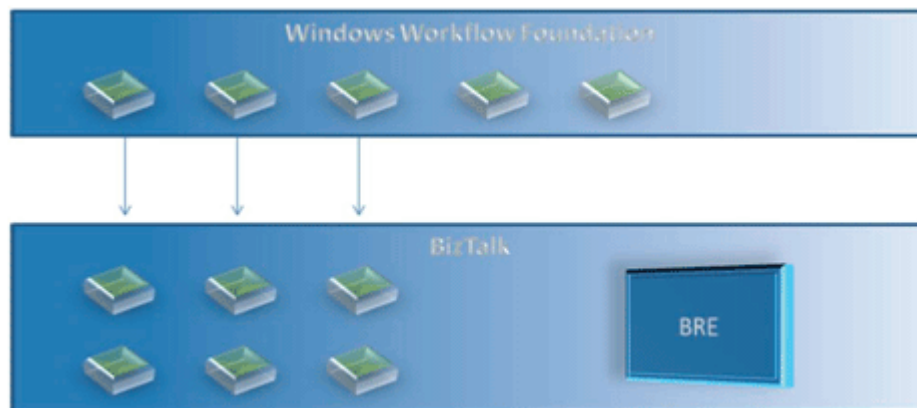


Figure 15. WF interaction with BizTalk Server

Increasingly, there is a need to coordinate the interactions between real human actors in software. Humans are, of course, key participants in almost every software system, especially in collaborative processes, and they play a key part in a composite application. There are some common challenges faced when involving humans in structured workflow systems. These include scenarios such as approvals, when a participant is out sick or on vacation, and provisioning a user.

Within human workflow (see Figure 18), people communicate with various systems and other people in a business process implemented in software using a workflow model. In this model, prebuilt units of behavior and defined workflows can be coordinated. The key to human workflow is that those units of behavior represent not only system-performed actions, but also actions and decisions undertaken by human actors.

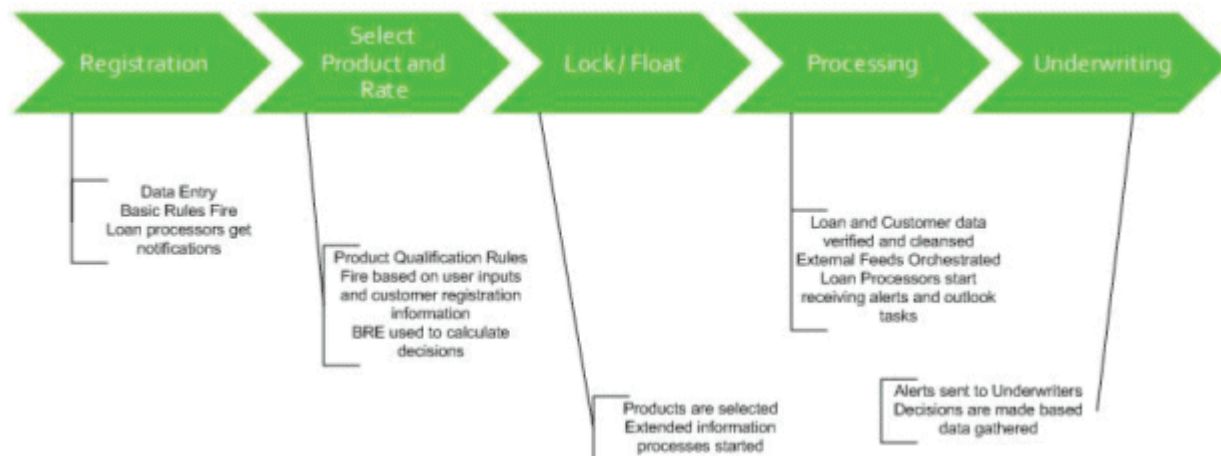


Figure 16. Business process modeled

The detailed scenarios will be discussed later. The referenced scenarios will cover how WF can extend the following business capabilities:

- Products and pricing
- Loan registration

Here, we will discuss the overarching workflows. There are two primary workflows that govern the loan-origination solution:

- **Master Loan workflow.** Controls the full lifetime of the loan, from the original data entry of the registration to the underwriting decision (see Figure 19).
- **Pricing workflow.** Used by the secondary marketing employees who are responsible for importing and applying daily rates on products.

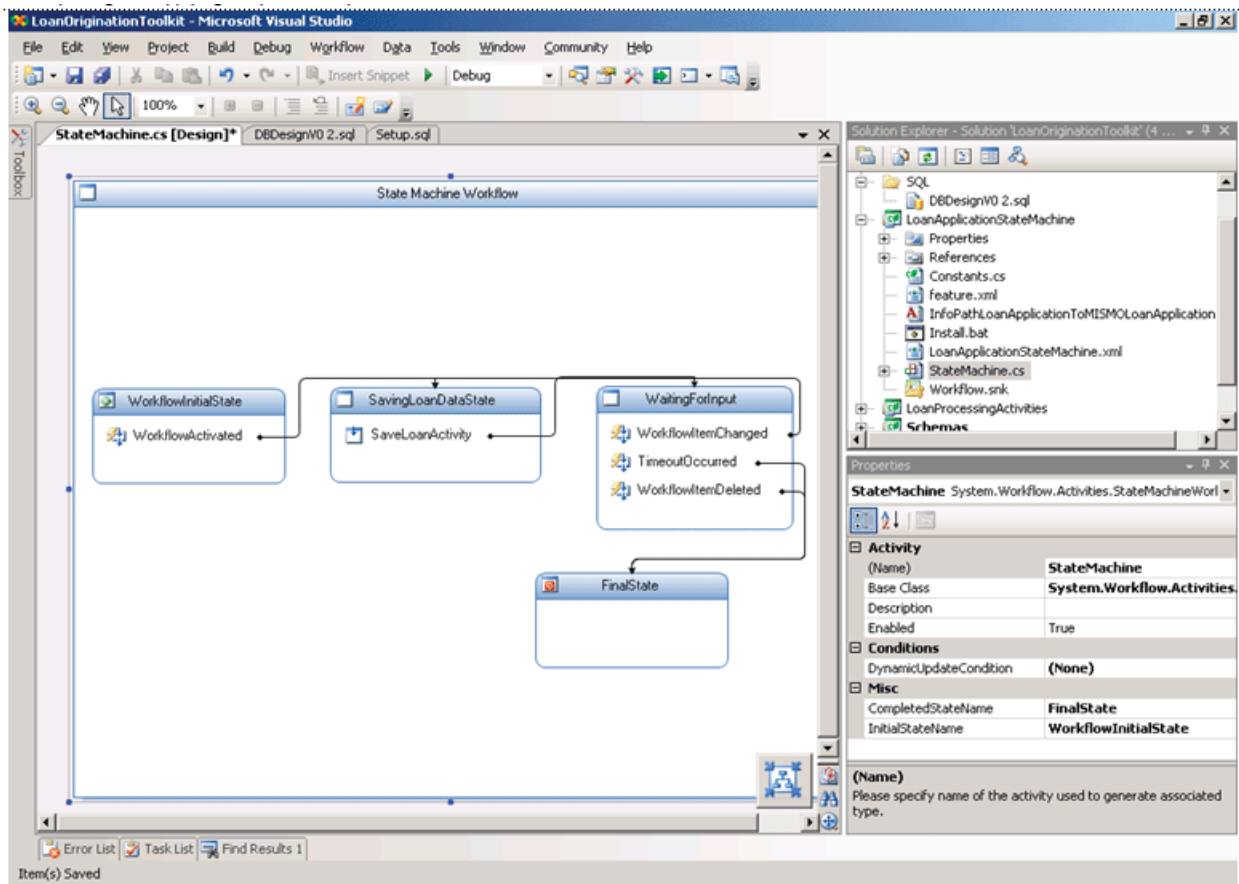


Figure 17. Master Loan workflow

Services-Layer Architecture

With this solution, we decided to have a dedicated messaging layer for integration. This layer is consistent with terms that are floating around the industry today, such as Enterprise Service Bus (ESB). However, for the sake of this solution, it will simply be referred to as a message bus.

Separating out the integration layer gives this solution many advantages. The architecture decisions leading to this course of action (instead of just using WCF for point-to-point integration) are the following:

- **Centralized messaging layer.** In a heterogeneous environment, such as banking, a centralized messaging layer reduces complexity by removing the endless number of point-to-point integrations, thus reducing code complexity, cost, and error rates.
- **Message management.** A message bus will provide Business-Process Management (BPM) facilities to monitor the overall health of the messaging in the solution.
- **Scalability.** This is crucial when the business needs a resilient and high-traffic solution, such as loan origination.
- **Extensibility.** A requirement in the loan origination. There are many external and internal touch points that are interwoven within the process.

- **Reduction of integration complexity.** With so many services and varying applications, lack of integration is—in most cases—a business inhibitor.
- **Achieving agility.** Having this layer will provide the necessary reductions in code, streamline business workflow, and tend towards an SOA.
- **Eliminate application redundancy.** By using a message bus and Business Rules Engine (BRE), banks can slowly migrate off of redundant solutions or platforms.

The integration layer consists of Web services stubs exposed from Office SharePoint Server, in conjunction with the BizTalk Server Web services endpoints.

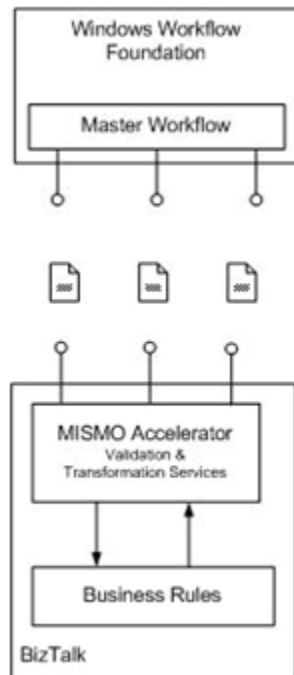


Figure 18. WF and BizTalk Server interaction

As shown in Figure 18, we used Web services to interoperate with Office SharePoint Server. This enables other external applications and/or workflows to interoperate using standard Web services with any layer in the loan-origination architecture, giving the solution ultimate flexibility.

Message Architecture

Messages that flow through the message bus will not only be Web services-compliant, but they will also use the business-message standard called MISMO. This is a U.S.-centric mortgage-messaging standard created by the Mortgage Bankers Association (MBA). It is critical to be compliant at both WF and BizTalk Server layers to ensure interoperability. To address this need, the messaging layer contains a MISMO accelerator, as shown in Figure 19.

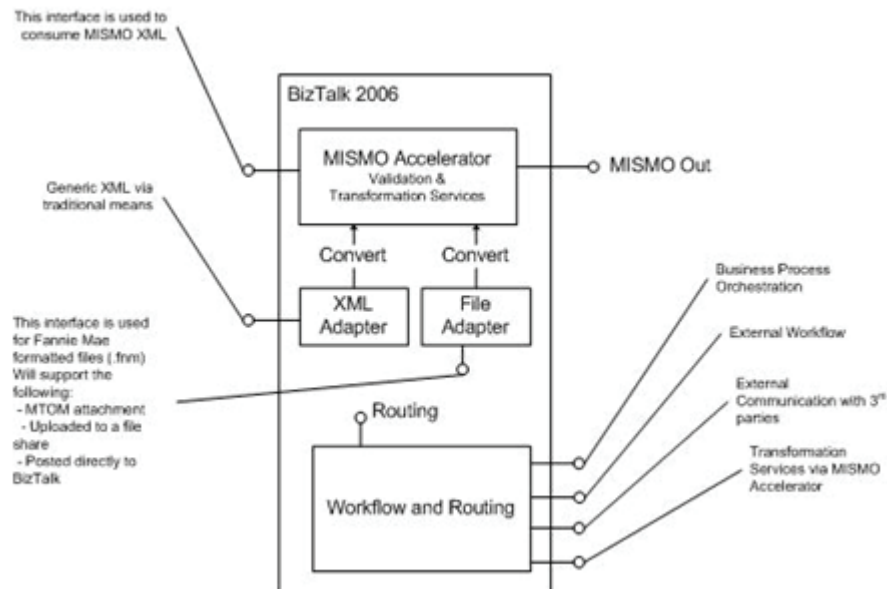


Figure 19. MISMO accelerator in BizTalk Server

Message Schemas

Throughout the solution, it was critical to use standardized XML (see Figure 20) to ensure standardized messaging and protocols for both composability and extensibility. In addition to standardized XML, the MISMO schemas—which provide the proper business context—ensure interoperability with existing commercial off the shelf (COTS) applications and existing third-party managed services.

This level of extensibility provides the ability to orchestrate several other external business processes that might be housed in other applications. This is a very common scenario. In many cases, there could be several diverse loan systems in varying capacities that must be integrated. Most times, these are used for specific business channels (for example, correspondent banking versus wholesale banking). In this architecture, BizTalk Server will coordinate the Web services using out-of-the-box Business-Process Management (BPM) facilities.

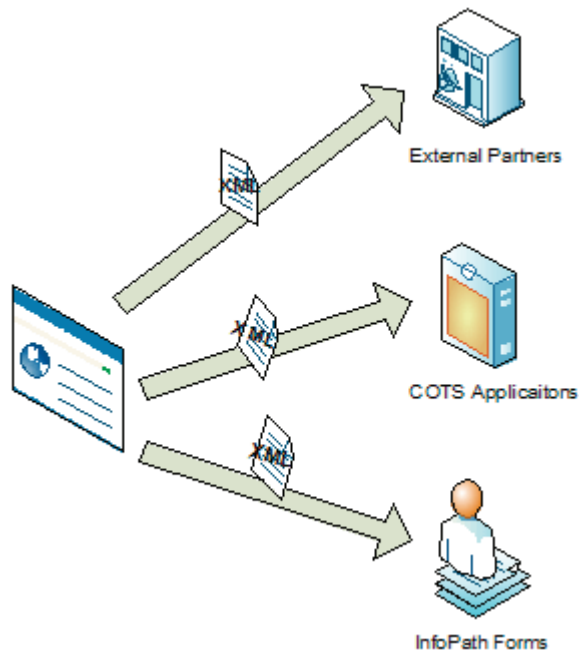


Figure 20. Extensibility of the platform

When using the MISMO schemas shown in Figure 21, there were some gaps, because the last update of the MISMO DTDs were a few years old. Listed here are the missing components, from both technical and business perspectives:

- **Outdated XML contracts.** Converted DTDs to schemas
- **Additional schemas.** Created for needed business-processes functionality

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns="http://tempuri.org/"
MORTGAGE_INSURANCE_APPLICATION_REQUEST_v2_1"
elementFormDefault="qualified" targetNamespace="http://
tempuri.org/"
MORTGAGE_INSURANCE_APPLICATION_REQUEST_v2_1"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="MI_APPLICATION">
    <xs:complexType>
      <xs:sequence>
        <xs:element
          ref="MI_REQUEST" />
        <xs:element
          ref="REQUESTING_PARTY" />
        <xs:element
          minOccurs="0" maxOccurs="1" ref="SUBMITTING_PARTY" />
        <xs:element
          minOccurs="0" maxOccurs="unbounded" ref="CREDIT_SCORE" /
        >
        <xs:element
          minOccurs="0" maxOccurs="unbounded" ref="KEY" />
        <xs:element
          ref="LOAN_APPLICATION" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="MI_REQUEST">
    <xs:complexType>
      <xs:sequence>
        <xs:element
          minOccurs="0" maxOccurs="1"
          ref="THIRD_PARTY_ORIGINATOR" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Figure 21. MISMO schemas used

Table 2. MISMO messages used

Message name	Sender	Receiver	Purpose
MortgageApplication	Office InfoPath	BizTalk Server orchestration and business-rules engine	Registering in loan system
RequestFlood	BizTalk Server orchestration	Stub third-party provider	Retrieving flood data about a property
RequestCredit	BizTalk Server orchestration	Stub third-party provider	Retrieving borrower's credit data
RequestMI	BizTalk Server orchestration	Stub third-party provider	Retrieving property's mortgage insurance
RequestTitle	BizTalk Server orchestration	Stub third-party provider	Retrieving property's title data

Table 3. Additional schemas created

Message name	Sender	Receiver	Purpose
Request3rdPartyServices	Office SharePoint Server loan workflow	BizTalk Server orchestration	Retrieving third-party data like Flood, Title, and so forth
Store3rdPartyResponse	BizTalk Server orchestration	Web service within Office SharePoint Server	Sending third-party data back to Office SharePoint Server
GetEligibleRates	Office SharePoint Server loan workflow	BizTalk Server orchestration	Retrieving products for which a borrower is eligible
GetProviders	Office SharePoint Server workflow	BizTalk Server	Retrieving list of providers for a specific loan

Message Flow

Messages flow in and out of the centralized message hub using standardized MISMO messages, as discussed earlier. Additional messages had to be developed for this solution.

Points to note:

- The majority of the messages flow through the message hub.
- All integration-based orchestrations and messaging go through the message bus.
- The message hub brokers the BRE.
- WF is the overarching control that coordinates the entire business process.
- Web services will be used as the primary integration technology.

Orchestration Architecture

Orchestration is handled in two places in this solution. As described in the Office SharePoint Server section, WF handles the human and page workflow of the solution. The WF workflow integrates with the second orchestration layer, BizTalk Server.

BizTalk Server will provide all orchestrations around the following:

- Integration with third-party services, internal applications, and management of external workflow
- Interaction with the BRE
- Interface for all external integrations with the loan system

Business-Rules Engine Architecture

The BRE is an abstraction layer to separate the workflow from the solution. This will give the solution the necessary flexibility required in today's agile banking environment. The rationale for this is the intensive business rules in loan origination, in which there are many business rules that have strictly similar logic. Separating the workflow from the solution generalizes these and provides a greater level of reusability.

Rules contained in the BRE will be built using .NET Framework 3.0 technologies. It is intended that these components will be called from BizTalk Server or exposed as Web services at an extensibility point. Because the BRE is just another .NET assembly, the BRE can be extended to any .NET application by referencing the specific implementation of the BRE.

It is assumed that the reader has a basic understanding of BREs. Therefore, we will minimize the explanation of these concepts and devote our conversation to the implementation of BRE in the context of BizTalk.

Terms

The following terms will be used:

- **Rules.** The specific business-rule functionality that needs to be computed or verified.
- **Policies.** Logical groupings of rules.
- **Facts.** Arguments passed to the policies and rules. These facts are evaluated and used to perform some level of business functionality.
- **Vocabulary.** Rule conditions and actions, usually expressed in domain-specific or industry-specific nomenclature.

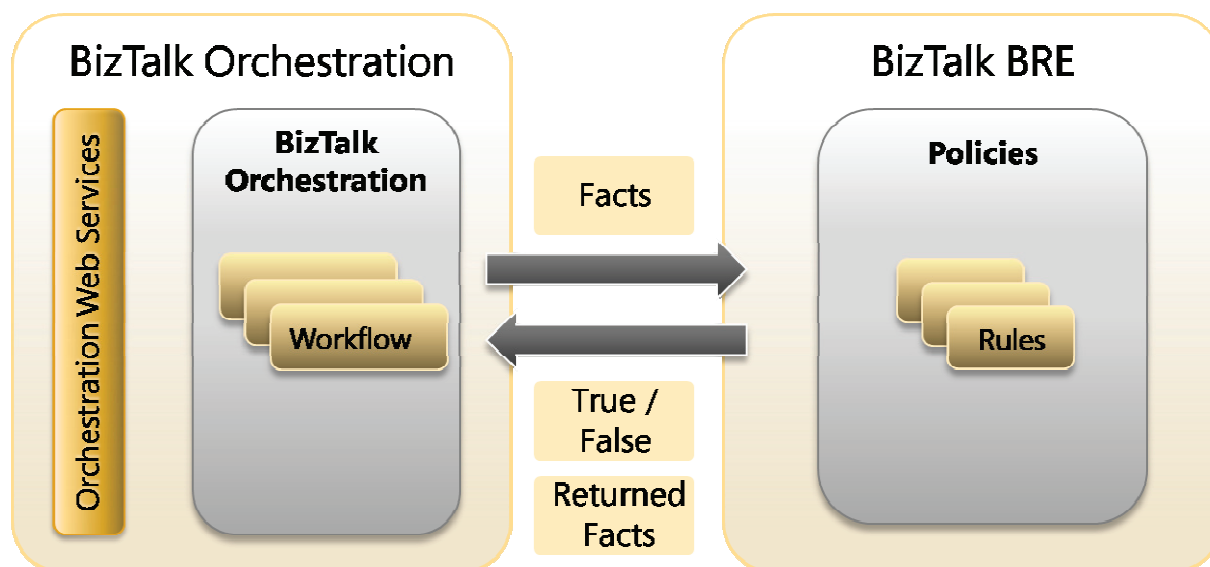


Figure 22. Integrating with the BizTalk Server BRE

The process shown in Figure 22 is managed through a common set of highly integrated tools. The Orchestration Designer, together with the BizTalk Server Editor and the BizTalk Server Mapper, provide an effective way to define a business process and the rules it uses. However, it can be useful to have an easier way to define and change business rules.

Making the Case for BRE for Loan Origination

Usage of BREs in a formal manner is still not as common as one would think. BREs are not a new concept; they actually have been around for some time now. These engines were first used in the 1950s for experimentation with artificial intelligence. Through the course of the years, they have been used to isolate business rules in one discrete place. Developers will most often use the BRE; but, in BizTalk Server, business-oriented users can create and modify sets of business rules using a tool named the Business-Rule Composer, as shown in Table 4.

Table 4. The BRE case

Business challenge	BRE case
Loan origination has complex rules based on a specific workflow.	BizTalk Server has an open and extensible way of creating orchestrations with shapes and linked to specific business-rules implementations.
The majority of rules and calculations do not change often. However, invocation, order, and parameters of those rules change.	Generic or reusable business rules can be created very easily on the BizTalk Server platform. There is clear separation of the Orchestration and Business Rule integrated development environments (IDEs). Developers or business analysts can modify rules in a WYSIWIG manner.
Loan systems have many business channels, with varying applications accessing them.	With BizTalk Server and its adapter framework, different interfaces can be used for just about any protocol. The most open would be the Web services or XML adapter. We use the Web services adapter for the loan-origination architecture.
With increased merger and acquisition, it is common to have redundancy of loan systems.	Managing multiple systems is a cumbersome and large development effort. BizTalk Server can manage internal and external orchestrations in a short-running or long-running workflow environment.
Multiple loan systems with unique business rules.	Redundant loan systems can be managed in a similar method. Sometimes, it makes sense to keep these systems around or to have a phased consolidation approach.
Loan systems have an increasing need to integrate with internal and external systems, such as Demand Deposit Account (DDA), Customer Relationship Management (CRM), or marketing, and external providers, such as credit or mortgage insurance.	Integration is the strong suit of BizTalk Server. With an extensive adapter framework and message-bus architecture, integration with disparate systems has never been this easy. Relevant adapters include: MISMO (reference implementation) Interactive Financial Exchange (IFX) SWIFT 3270 emulation

This implementation of the BizTalk Server BRE is useful when a complex set of business rules must be evaluated. Deciding whether to grant a loan might entail working through a large set of rules, based on the customer's credit history, income, and other factors. In addition to specific rules, these factors must be in synchronization with several business processes determined by the bank. Underwriters are alerted whether to approve an applicant, depending on a number of things, including the applicant's age, gender, and external data requests from third-party services. Expressing all these rules as conditional statements using, say, an orchestration's Decide shape might be possible, but it would be fairly complex to implement. For rule-intensive processes like these, the BRE is the best design choice.

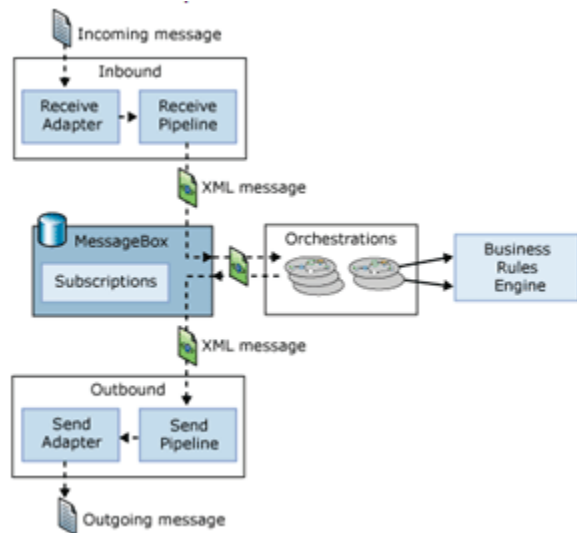


Figure 23. BizTalk BRE message flow

As Figure 23 shows, a message is received through a Receive adapter. Different adapters provide different communication mechanisms, so a message might be acquired by accessing a Web service, reading from a file, or in some other way. The message is then processed through a Receive pipeline. This pipeline can contain various components that do such things as converting the message from its native format into an XML document, validating a message's digital signature, and more. The message is then delivered into a database named the MessageBox, which is implemented using Microsoft SQL Server.

The logic that drives a business process is implemented as one or more orchestrations, each of which consists of executable code. However, these orchestrations are not created by writing code in a language such as C#. Instead, a business analyst or (more likely) a developer uses BizTalk Server designers to graphically organize a defined group of shapes in order to express conditions, loops, and other behavior. Orchestrations will use the BRE, which provides a simpler and more easily modified way to express complex sets of rules in a business process.

Each orchestration creates subscriptions to indicate the kinds of messages it wants to receive. When an appropriate message arrives in the MessageBox, that message is dispatched to its target orchestration, which takes whatever action the business process requires. The result of this processing is typically another message, produced by the orchestration and saved in the MessageBox. This message, in turn, is processed by a Send pipeline, which might convert it from the internal XML format used by BizTalk Server 2006 to the format required by its destination, add a digital signature, and more. The message is then sent out using a Send adapter, which uses an appropriate mechanism to communicate with the application for which this message is destined.

A complete solution built on the BizTalk Server 2006 engine can contain various parts (sometimes referred to as artifacts): orchestrations, pipelines, message schemas, and more. These parts, or artifacts, can be worked with as a single unit, referred to as a BizTalk application. A BizTalk application wraps all of the pieces required for a solution into a single logical unit, making it the fundamental abstraction for management and deployment.

Different kinds of people perform different functions using the BizTalk Server 2006 engine. For example, a business analyst might define the rules and behaviors that make up a business process. The analyst also determines the flow of the business process, defining what information gets sent to each application and how one business document is mapped into another. After the business analyst defines this process, a developer can create a BizTalk application that implements it. This includes such things as defining the XML schemas for the business documents that will be used, specifying the detailed mapping between them, and creating the orchestrations necessary to implement the process. An administrator also plays an important role by setting up communication among the parts, deploying the BizTalk application in an appropriately scalable way, and performing other tasks. All three roles—business analyst, developer, and administrator—are necessary to create and maintain BizTalk Server 2006 solutions.

Not only does the BRE implementation make developers more productive, but there are other architecture considerations to keep in mind. The architectural case for BRE in loan origination consists of:

- Composite-style architecture that is not tightly coupled with any one system or technology implementation. .NET, Java, or even COBOL systems can interoperate with the BRE through Web standards.
- Clear separation of orchestrations and rules for maximum reusability.
- Unique definition of calculations or business rules, which can be invoked based on a specific workflow.
- Scalable infrastructure for rules and orchestrations.
- More manageable regulatory issues, such as separation of duties.

Database Architecture

The database tier of this solution is built out using Microsoft SQL Server 2006. Aside from the table structure, other database services are exposed:

- **SQL Reporting Services** will be used to expose Web Parts in Office SharePoint Server.
- **SQL Analytical Services** will be used to expose Web Parts in Office SharePoint Server.

Enabling Scenarios

Now that the architecture is defined, we can show some possible usage scenarios based on this architecture. As discussed in the business description, we will model and describe how these scenarios can be implemented.

Products and Pricing

It can be difficult to manage pricing from multiple sources. With all the different manipulations of data needed, the Office Excel formulas are an easy-to-use and empowering tool for secondary marketing analysts.

With this scenario, we want to demonstrate the following:

- **Data management.** Data kept in Office Excel files can be difficult to manage. We want to keep this data on the server to keep the lender compliant with data-retention regulations.

- **Auditability.** In many cases, data is difficult to audit when there are many users manipulating data in client-side tools. With this solution, the Office Excel files can reside on the Office SharePoint document libraries, where auditing is performed out-of-the box.
- **Approvals.** Workflows can be developed around documents for approval processes. This is often needed in these environments when the data generated is depended on by many processes.
- **User experience.** Office Excel is the client for our solution, because business users want to have familiar tooling; business owners do not want to pay for retraining; developers do not want to create redundant tools; and architects want to reduce complexity and cost.

By using Office Excel, the business users are more productive and empowered. However, this does not prohibit additional or alternate user interfaces. For example, Office SharePoint Server can also be used if a Web user interface is required. From a technical perspective, Office Excel is used for this scenario to demonstrate the capabilities of document workflow and rich calculation functionality coupled with workflow. For larger banks, this might not be the optimal solution. For those implementations, a Web-based Office InfoPath or Office SharePoint application could be appropriate.

Figure 24. Product and pricing technical overview

Figure 24 shows the overview of the technical implementation of the scenario. The process exposes several capabilities. (There are background processes to keep in mind. The industry-standard "blackout period" will apply at the beginning of this scenario. A *blackout period* is the time period in which the daily rates have not been defined, and all loan-product and pricing selection is disabled for brokers.) The following capabilities are exposed by the process:

1. The secondary marketing analyst starts by opening a standard Office Excel template that contains all the necessary calculations. The lender is ensured that the same calculations are used by all personnel by storing the calculations on the server.
2. The secondary marketing analyst reviews feeds and market data and puts it into Office Excel.
3. Manipulations are performed on the data, and the document is saved to the document library.
4. The workflow is then kicked off. The appropriate manager is sent an e-mail alert to review and approve.
5. The manager approves the rates.
6. Workflow persists product and rate data to the LOS SQL Server.
7. The Web site blackout period ends, and registration can now continue.
8. The workflow then starts the distribution process:
 - a. Data is transformed and sent to disparate LOB systems.
 - b. E-mail alerts are sent to appropriate lender staff (processors).
 - c. E-mail alerts are sent to brokers, with a link to the current rates.
 - d. Rates can now be used by the application.

Loan Registration

With loan registration, we expose a Web portal through Microsoft Office SharePoint Portal Server 2003. As described in the UI description, an array of Web Parts makes up the UI. With this composite-style UI, custom workflows are easier for the user to orchestrate.



Figure 25. Registration technical overview

Figure 25 shows the overview of the technical implementation of the scenario. The process exposes the following capabilities:

1. Broker comes to lender portal and authenticates.
2. The UI is dynamically generated, based on broker's role.
3. The broker enters in the customer's information into the registration pages.
4. Alternatively, the broker can choose to send a MISMO-compliant XML feed or standard Fannie Mae files to the MISMO accelerator to reduce the duplication of data entry. (Additionally, by sending Fannie Mae files to the MISMO accelerator, we eliminate multiple payload formats that result in consistency throughout architecture.)

The Master Loan workflow is now started.

Appropriate messages are sent to BizTalk for business rules.

External data is retrieved automatically when workflow events occur, based on loan processors and required data.

Conclusion

In this article, we walked you through how to build sustainable banking architectures. The loan-origination reference architecture provided an enterprise-scale application with all of the common needs of banking systems. Microsoft Office SharePoint Server 2007 and Microsoft BizTalk Server provide a powerful set of capabilities to build composite applications. Composition is enabled at the presentation, application-services, business-logic, and data tiers. This enables cross-functional solutions that offer a composite user interface that exposes business functions and capabilities across a heterogeneous set of back-end IT assets. These solutions also provide collaborative business capabilities that fill the white space between traditional business applications and personal productivity tools.

This article demonstrates how to build an enterprise Office Business Application (OBA). Providing the business context, the OBA described was developed as a Reference Application Pack for Loan Origination. This reference application covered a variety of collaboration scenarios between the different levels of banking.

References

- MSDN Industry Center - Banking - Reference Application Pack for Loan Origination Systems (<http://msdn2.microsoft.com/en-us/architecture/bb265266.aspx>)
- MSDN Industry Center - Financial Services (<http://msdn2.microsoft.com/en-us/architecture/aa699365.aspx>)
- BizTalk Server Developer Center (<http://msdn2.microsoft.com/en-us/biztalk>)
- Office Business Applications (OBA) (<http://msdn2.microsoft.com/en-us/architecture/aa699381.aspx>)
- Mike Walker's Blog (<http://blogs.msdn.com/MikeWalker>)

About the Author

Mike J. Walker

Architecture Strategist, Financial Services, Architecture Strategy Team

Microsoft Corp.

Mike Walker is the managing architecture strategist for the Financial Services vertical at Microsoft. Walker is responsible for driving and evangelizing Microsoft's worldwide industry and vertical strategy in the banking, insurance and securities segments. Specifically, Walker ensures that financial services institutions around the world realize the full extent of Microsoft's vision and value proposition, by overseeing areas such as industry strategy, marketing, solution development, partner development, thought leadership, and executive relations.

Walker joined Microsoft in early 2006. His background is as a financial services strategist, specializing in business transformation around technology, but he combines this experience with a strong focus on strategic execution. Prior to joining Microsoft, Walker 's worked as an enterprise architect in banking, where

he developed business strategies, conducted strategic infrastructure planning, and implemented technology projects. His experience includes bespoke and package implementation, enterprise process design and management, applications management and systems selection.