Office Business Applications

Building Composite Applications using the Microsoft platform

November 2006

Microsoft Architecture Center: www.microsoft.com/architecture

Microsoft MSDN Solution Architecture Center: www.msdn.com/architecture

Authors' Biographies

Atanu Banerjee

Atanu Banerjee is on the Architecture Strategy Team at Microsoft. He has ten years of experience in the software industry, having worked earlier on solutions for supply chain management and advanced process control. In his current role, Atanu looks at architectures for enterprise solutions that leverage next generation technologies for collaboration, analytics, business process management, and integration. In this capacity Atanu frequently interacts with key decision makers in customer and partner organizations and with thought leaders in industry and academia.

Atanu joined Microsoft from i2 Technologies, where he worked in various roles for more than seven years. He was chief architect for their supply and demand management product line, and prior to that served as development manager, product architect, team lead, and software developer for various teams. During that time he wrote a lot of code, designed new solutions, and worked with some large manufacturing customers. Prior to joining i2, Atanu worked at Aspen Technologies in the advanced control systems group, designing and implementing model predictive control systems for the process industry. Atanu received a Ph.D. from Georgia Tech in 1996, and holds a bachelors degree from IIT Delhi, India. He resides in Redmond, WA with his wife and son.

Email: atanub@microsoft.com

Moin Moinuddin

Moin Moinuddin is a retail industry architect and is part of D&PE organization at Microsoft. He is responsible for evangelizing the Microsoft platform and products to the architects in the retail community. Moinuddin works with architects at customers and partners. He is also responsible for influencing internal product teams in development of new products to meet requirements from the retail community. He evangelizes initiatives and development in the retail industry to the Microsoft product teams and the Microsoft field. As an industry architect Moinuddin generates white papers showing how to build future retail products using the Microsoft platform and tools. He also engages in speaking at the global architecture forums. In addition, Moinuddin represents Microsoft at the National Retail Federation's ARTS group, where he chairs the subcommittee that is working on creating standards for the food services industry.

Prior to joining Microsoft, Moinuddin spent twelve years in retail and payment processing companies. He was a key member that developed a point of service solution for the hospitality industry, and then was part of a start up that built one of the largest payment gateways for processing credit card and electronic check transitions.

On the personal front, Moinuddin lives in Bellevue, WA, USA with his wife and two Children. Moinuddin holds a B.S in Computer Science from Osmania University, India and a M.S. in Computer Science from University of North Carolina at Charlotte, USA.

Email: mmoin@microsoft.com

Blog: http://blogs.msdn.com/mmoin

Mike J. Walker

Mike Walker is the managing architecture strategist for the Financial Services vertical at Microsoft. He 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.

Mike 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, he 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

Email: mikewalk@microsoft.com

Blog: http://blogs.msdn.com/mikewalker/

Technical Editor

Jon Tobey

Copy Editor

Dave Eriksen

Foreword

Companies are adopting service orientation as a design principle to drive enterprise agility and business innovation. With service-oriented architecture (SOA), they can efficiently respond to business changes and evolve their IT systems rapidly. Service orientation is about decomposing the IT assets into easily consumable services. A composite application is the primary vehicle to deliver business value through service orientation and is a great way to aggregate these services to support cross-functional processes. Theoretically, composite applications allow for these services or components to be mixed and matched like blocks, allowing developers to customize applications to evolving enterprise needs relatively easily.

With the release of 2007 Microsoft Office System clients, servers, and tools, Microsoft delivers a true application platform that can be used to create collaborative, role-based, easy-to-use solutions that extend the traditional LOB applications and enterprise systems. Not only has Microsoft significantly improved the capabilities and tools for personal and business productivity, but it has also invested significantly in the extensibility and programmability of the 2007 Microsoft Office System. Microsoft calls these solutions Office Business Applications (OBAs) and recently announced the technology to design and develop OBAs[1].

This book is about composite applications and how they can be developed as OBAs using the 2007 Microsoft Office System. It explains the concept of a composite application and discusses in great detail the different aspects of composition. It provides an overview of the technologies available in the 2007 Microsoft Office System and gives several examples from various industries to build OBAs using composition at the presentation, business logic, and data layers.

This book is meant for solution architects, industry architects, or senior developers who are designing, developing, and deploying composite applications. The OBA examples from manufacturing, retail, and financial services industries will allow solution architects in these industries to understand the concepts in the context of scenarios specific to their industries.

Chapter 1 introduces the concept of a composite application and discusses the benefits of composition as enterprise alignment, adaptability, and agility. For each of the benefits, the author lists best practices that can be applied to achieve these benefits.

Chapter 2 presents the 2007 Microsoft Office System as the platform to build composite applications. The authors introduce the concept of OBAs. They discuss Microsoft technologies for composition in presentation, productivity, application, and data tiers. The chapter ends with guidelines to building an OBA.

Chapter 3 presents guidelines on deploying an OBA in an enterprise. It suggests a step-by-step approach for deploying OBAs to support a single business process at a time without re-architecting the entire backend system architecture in one big bang. It walks the readers through best practices for:

- Deploying departmental SharePoint sites to host local documents and processes.

[1] http://msdn2.microsoft.com/en-us/office/aa905528.aspx

- Connecting multiple departments.
- Connecting business processes to LOB systems.
- Adding data connections for cross-functional processes.
- Connecting business processes to the systems on the edge of the enterprise.

Chapter 4 presents a sample OBA designed to address the collaboration between a retailer and a manufacturer. It describes the challenges a manufacturing enterprise faces today and presents synchronization of demand, supply, and product development as the steps towards becoming a demand-driven supply network (DDSN) to drive enterprise agility and responsiveness. It then presents the architecture of a sample solution built on Microsoft Office SharePoint Server (MOSS) for the manufacturing enterprise to achieve the goals of a DDSN. The sample solution is a great example of a composite application, stitching together documents, processes, business rules and industry standard schemas – all while bringing the power and familiarity of Microsoft Office clients to people-centric processes.

Chapter 5 presents a sample OBA designed to address the process of loan origination in financial banking. It explains the high-level process and identifies the complexities involved in implementing a solution. This OBA uses MOSS as a records repository and the hub for all records management processes. The OBA uses the Workflow Foundation in MOSS for user-centric workflows and implements long-running system workflows in Microsoft BizTalk Server.

Chapter 6 presents various examples of OBAs that can be designed to enhance collaboration between the retail store processes and the retail corporate systems. These OBAs can be built as dashboards for various store management processes such as promotions or workforce management.

Finally, the attached DVD on the back inside cover contains the bits for the supply chain collaboration OBA discussed in Chapter 4. You can also get these bits on MSDN at http://msdn2.microsoft.com/en-us/architecture/aa702528.aspx with documents and decks that explain the solution in detail.

The 2007 Microsoft Office System is a great platform to build composite applications. The investment Microsoft has made in the 2007 Microsoft Office System makes it easy to create collaborative and role-based OBAs that bring the power and familiarity of Microsoft Office to enterprise applications. I am super excited about the new Office System and OBAs, I am sure this will usher in a new era of business productivity. Please visit the Microsoft Architecture Center at www.microsoft.com/architecture and the Microsoft MSDN Solution Architecture Center at www.msdn.com/architecture. Drop me a line and let me know what you think of the book and the new 2007 Microsoft Office System.

Javed Sikander
Director, Industry Architecture
Microsoft Corporation
Email: javeds@microsoft.com
Blog: http://blogs.msdn.com/javeds

Table of Contents

Building Composite Applications Using the Microsoft platform

Chapter 1
What are Composite Applications ?

Introduction

Globalization, specialization, and outsourcing require people to work in more collaborative ways than before. This trend requires a matching change in the tools that information workers use to gain insight, collaborate, make decisions, and take action. Today, most business applications are effective at automating transactions, but do not enable rich collaboration across functional boundaries. This usually leads information workers to use personal productivity tools to perform the complex interactions required to conduct business. However, this in turn leads to a loss in productivity, as users are forced to cross from one set of tools to another, often manually transporting the data through means such as cut-and-paste. These gaps between different business applications and productivity tools need to be bridged for information workers in a way that is seamless, synchronized, and secure.

Ideally, it would be easy to stitch together different business applications and also enable collaboration for information workers, but in a way that complex interactions between people can plug easily into structured business processes. Unfortunately, though, this has not been easy to do, as today's business applications are usually monolithic and difficult to modularize. There has been a lot of discussion in the industry around composite applications that are quick to build and deploy, and also easy to change when the business changes. The stated goal of these kinds of business solutions is make business processes more efficient in the real world, as opposed to automating idealized views of process that no longer meet the needs of the business.

This is certainly a laudable goal, and there have been a number of industry efforts in this direction. However, there really has not yet been an effective way to enable composition in a way that is contextual, collaborative, easy to use, role-based, and configurable. In many cases, the platform, tools, and architecture for composition introduce new technical complexity that is overwhelming in itself, and organizations’ skill sets are not up to this challenge. This book will show how to architect composite applications, and how the 2007 Microsoft Office System provides a good platform for building such composite applications - called Office Business Applications (OBA). The 2007 Microsoft Office System has a broad set of capabilities that enable composition, leading to applications that are familiar to end users and also easy to use. This book also shows how to pull in other platform technologies that are needed for building these kinds of composite applications.

Composition refers to a way of delivering enterprise solutions by assembling them from pre-built components rather than by building them from scratch. It also includes personalization and customization abilities, so that users can easily and quickly modify specific functionality in the solution. This book will also show that moving to a composition architecture of requires a fundamental shift in thinking, both for solution providers as well as for end users of enterprise solutions. This is analogous to getting a pre-fabricated house assembled for you, rather than having it custom built.

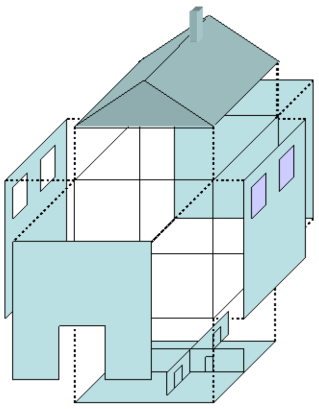


Figure 1: Building an enterprise solution using self-service architecture is like assembling a pre-fabricated house

However the benefits are substantial – as composition provides the means to achieve agility, adaptability, and alignment in the enterprise.

Real World Business Processes, and the Results Gap

Let us look at a typical business process as shown in Figure 2. This is the case where a sales person has a contact lead and wants to convert that into an actual order. If you look at a typical business application for sales automation, then a lot of work will have gone into creating a perfect business process for this - create the lead, decide when it is qualified, create the opportunity, do a quote, close the quote, complete the sale, and finally invoicing the customer. It is this "perfect" process that is shown in Figure 2.

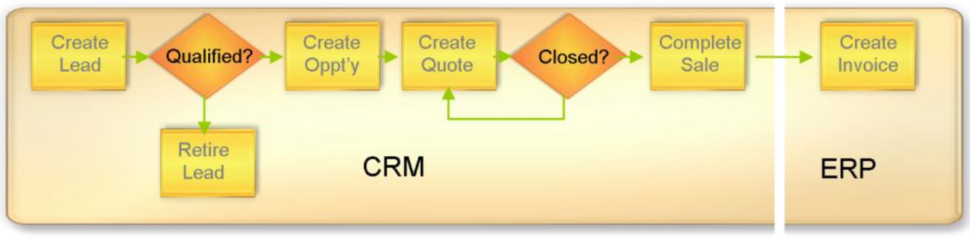


Figure 2: A perfect Sales Automation process

However, in reality the process really looks like Figure 3. To create the opportunity, someone has to obtain specs from the customer to understand what the customer really wants. They have to decide if this opportunity is valid, and if it is something that can be feasibly delivered. They need to cost the solution – probably using Excel - and this goes on. Clearly the majority of real world processes are not typically part of an ERP system or a CRM system, but these steps are integral to the process and just as important.

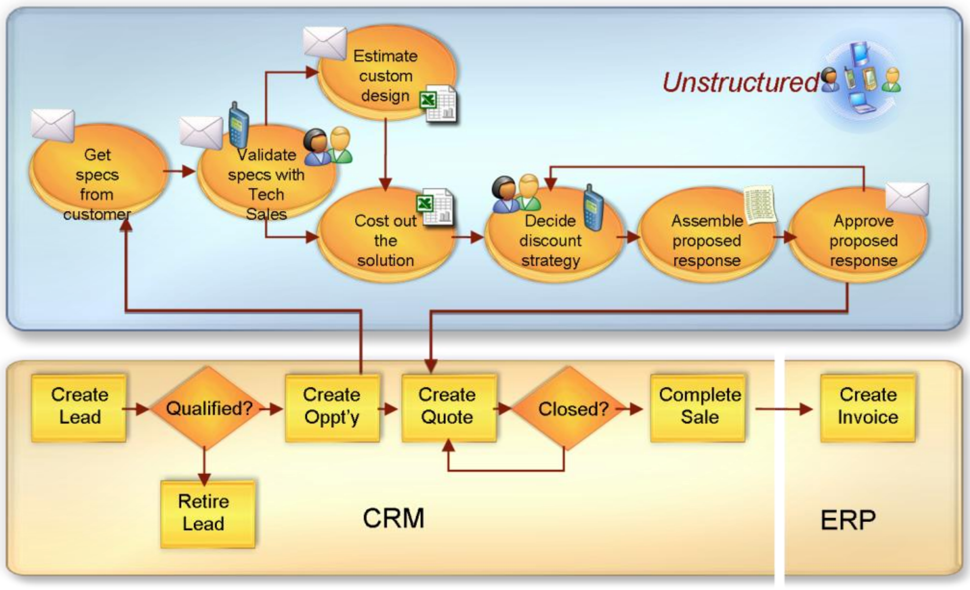


Figure 3: The sales automation process in the real world

We call this failure of traditional business applications to capture the real-world processes and the work that people do the *results gap*. It goes beyond processes and includes problems like finding and accessing the correct data to make the right decisions. Ultimately, it means that IT investments frequently under-deliver on the promises their proponents make.

The results gap has some critical consequences for the enterprise. This is called out in a Gartner, Inc. research note "The Knowledge Worker Investment Paradox,"[2] which states:

In most enterprises, an employee will get 50 percent to 75 percent of his or her relevant information directly from other people.

More than 80 percent of the enterprise's digitized resources are not accessible to the enterprise as a whole because they reside in individual hard drives and in personal files.

The individual owns the key resource of the knowledge economy - tacit and explicit knowledge - and most of that knowledge is lost when he or she decides to leave the enterprise.

The space between traditional business applications and the productivity tools that information workers use on a daily basis is very large, and so there are a number of business processes that are affected by the results gap. For example, business processes like sales and operations planning (S&OP), collaborative product design, new product introduction, and so on often lack out-of-the-box solutions. While enterprises look to vendors to provide applications for these business processes, it is often hard to find packaged applications that exactly fit the business need. That is because these processes are very collaborative, rather than strictly transactional, and the solution space is characterized by semi-structured data and processes.

[2] Gartner, Inc. research note "The Knowledge Worker Investment Paradox,"Regina Casonato and Kathy Harris, July 17, 2002

Often, when packaged applications are unavailable, then custom solutions are built. In most cases, business applications are built on top of an applications platform, rather than built from scratch. The choice of platform has different implications for the IT department that oversees development of solutions, and for the business users that commission them. IT management would like a platform that enables solutions which are quick to build and deploy, easy to modify and extend, and align with business needs. Business users would like a platform that enables rapid decision making; helps the business sense and respond to changing market conditions; and aligns business needs with IT assets. Solutions would need to provide rapid time-to-benefit, in line with the speed at which market conditions change.

The implications and benefits for both business users and solutions providers can be summed up in three words – agility, adaptability, and alignment. These three traits are the central themes of an article on Supply Chain Management called "The Triple-A Supply Chain" by Prof. Hau Lee in the October 2004 edition of the *Harvard Business Review*. In this article Prof. Lee makes the case that, "The best supply chains aren't just fast and cost-effective. They are also agile and adaptable, and they ensure that all their companies' interests stay aligned." It is the thesis of this book that the traits of agility, adaptability, and alignment are requirements on all enterprise solutions, and not just in the supply chain management space. Furthermore, composition is a means to these ends, and this book will show how to architect these kinds of composite applications on the 2007 Microsoft Office System.

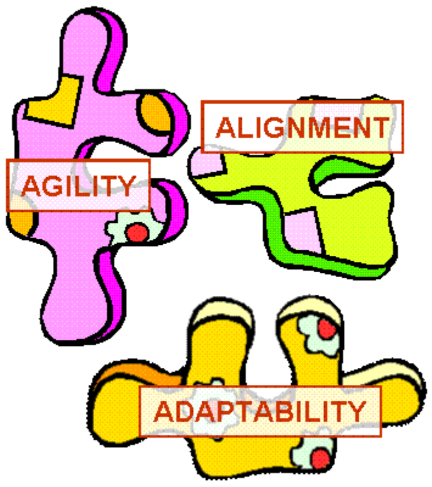


Figure 4: Triple - A benefits delivered by Self Service Architecture

What are Composite Applications?

A composite application is a collection of software assets that have been assembled to provide a business capability. These assets are artifacts that can be deployed independently, enable composition, and leverage specific platform capabilities.

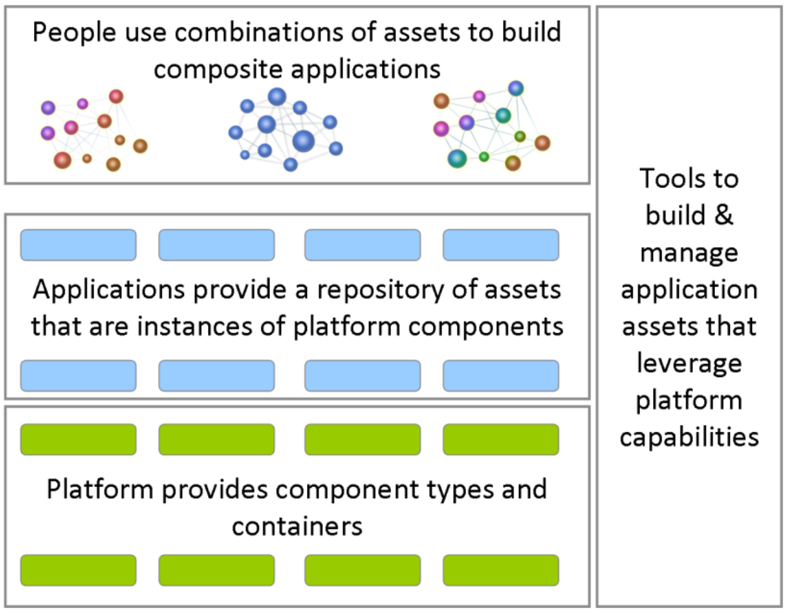

People use combinations of assets to build composite applications
Applications provide a repository of assets that are instances of platform components
Platform provides component types and containers
Tools to build & manage application assets that leverage platform capabilities

Figure 5: High-level representation of a composite application

In the past, an enterprise's software assets were usually a set of independent business applications that were monolithic and poorly-integrated with each other. However, to get the business benefits of composition, an enterprise must treat its software assets in a more granular manner, and different tiers of architecture will require different kinds of assets such as presentation assets, application assets, and data assets. For example, a Web service might be an application asset, an OLAP cube might be a data asset, and a particular data-entry screen might be a presentation asset.

An inventory of software assets by itself does not enable composite applications. This requires a platform with capabilities for composition—that is, a platform that provides the ability to deploy assets separately from each other, and in combination with each other. In other words, these assets must be *components*, and the platform must provide *containers*.

Containers provided by the platform need to be of different types, which map to the different tiers in the architecture. Enterprise architectures are usually decomposed into three tiers: presentation, application (or business logic), and data. So the platform needs to provide containers for these. However the 3-tier architecture assumes structured business processes and data, where all requirements are made known during the process of designing and building the system. By their very nature, composite applications presume that composition of solutions can occur after assets have been built and deployed – and so need to explicitly account for people-to-people interactions between information workers that are essential to get any business process complete. Usually these interactions are not captured by structured processes, or traditional business applications, and therefore it is critical to add a fourth tier - the productivity tier – to account for these human interactions. This is shown in Figure 6.

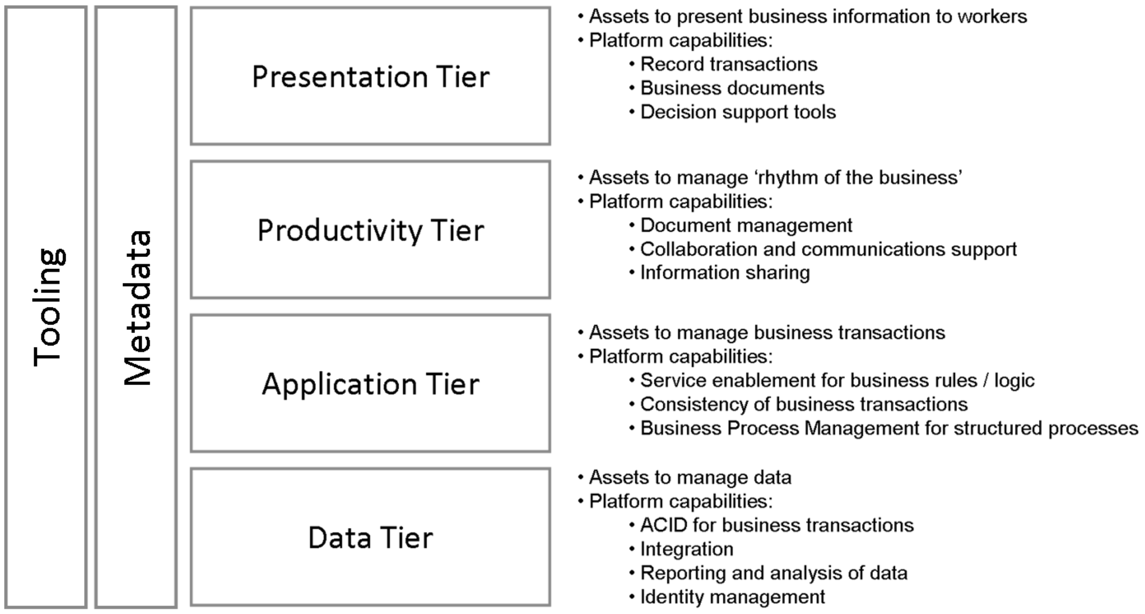

Tooling
Metadata
Presentation Tier
Productivity Tier
Application Tier
Data Tier
• Assets to present business information to workers
• Platform capabilities:
• Record transactions
• Business documents
• Decision support tools
• Assets to manage 'rhythm of the business'
• Platform capabilities:
• Document management
• Collaboration and communications support
• Information sharing
• Assets to manage business transactions
• Platform capabilities:
• Service enablement for business rules / logic
• Consistency of business transactions
• Business Process Management for structured processes
• Assets to manage data
• Platform capabilities:
• ACID for business transactions
• Integration
• Reporting and analysis of data
• Identity management

Figure 6: The four tiers of a composite application

Traditional discussions around the architecture of business applications tend to focus on the application tier as being the connection between people and data. Typically, however, the application tier contains structured business logic; and this holds for discussions around Service Oriented Architectures (SOAs), Enterprise Service Buses (ESBs), Service Component Architectures (SCAs), or most other architectural perspectives in the industry today – including first-generation discussions around composite applications. However, building a composite application requires a mindset that not only is the productivity tier a critical element of the stack, but also contains the most business value.

To expand on the comparison between composite applications and SOA, both of them target flexibility and modularization. However, SOA provides flexibility at just one tier: the structured business logic in the middle tier. Composite applications target flexibility at all four tiers. That said, a composite application is a great way to surface information out of an SOA, and having line-of-business (LOB) applications exposed as services makes it easier to build support for cross-functional processes into a composite application.

Therefore to design a composite application, a solutions architect must:

- **Choose a composition stack** –Pick one or more containers from each tier, and a set of components types that are deployable into those containers.
- **Choose components** – Define the repository of assets that must be built from this set of component types, based on business needs.
- **Specify the composite application** – Define the ways in which those assets will be connected, to provide a particular cross-functional process. The platform should enable these connections to be loosely-coupled.

Then after deployment, users will have the opportunity to personalize both assets and connections, as the composition stack should enable this through loose coupling and extensibility mechanisms.

What does a Composite Application Look Like?

A figurative representation of a composite application is shown in Figure 7, which shows a very abstract representation of an enterprise solution, deconstructed along the lines of Figure 6.

At the top are information workers, who access business information and documents through portals that are role specific views into the enterprise. They create specific documents during the course of business activities, and these activities are part of larger business processes. These processes coordinate the activities of people and systems. The activities of systems are controlled through process specific business rules, that invoke back end Line Of Business applications and resources through service interfaces. The activities of people plug into the process through events that are raised when documents specific to the process are created, or modified. Then business rules are applied to the content of those documents, to extract information, transform it, and transfer it to the next stage of the process.

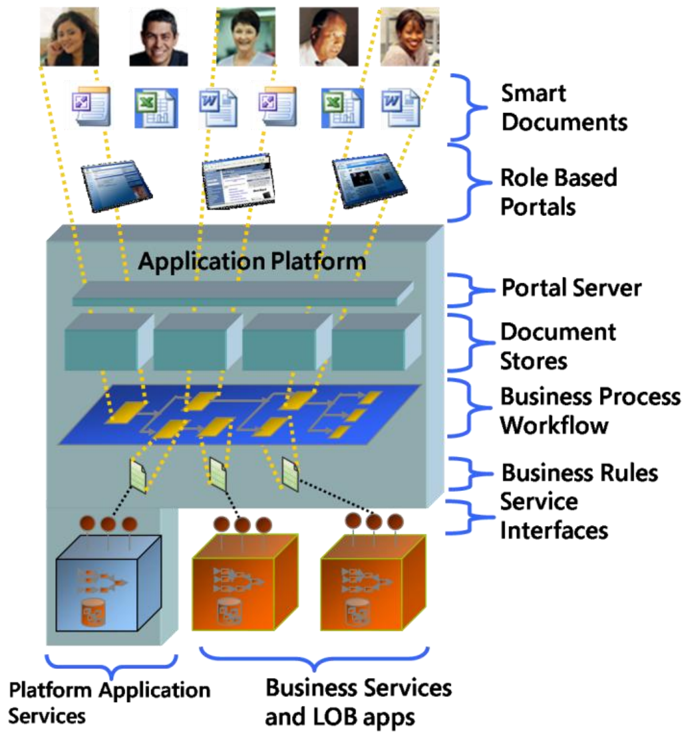


Figure 7: Deconstructing an enterprise application

Today most line-of-business applications (LOB) are a collection of resources, hard-coded business processes, and inflexible user interfaces. However based on the previous section, it is clear that enterprise solutions need to be broken down into a collection of granular assets that can be assembled into composite applications. A high-level approach for doing this to any business process is listed below, and subsequent chapters will elaborate upon these steps.

1. Decompose the solution for a business process into software assets corresponding to the elements shown in **Error! Reference source not found.**.
2. Package all assets corresponding to a given business process into a “process pack” for redistribution and deployment. This would contain metadata and software components, and solution templates that combine them. The process pack would also contain service interface definitions that would enable connections to other IT systems. These connections would be enabled by implementing the service interfaces, for example. to connect to LOB applications and data. The goal is to be able to easily layer a standardized business process onto any heterogeneous IT landscape.
3. Deploy the process pack onto a platform that provides containers for the types of assets that the solution has been decomposed into. The platform should provide capabilities for rapid customization, personalization, reconfiguration, and assembly of assets.
4. Connect the assets within the process pack, to existing LOB systems, and other enterprise resources by implementing the services interfaces. These connections could be made using Web services technologies, other kinds of custom adapters, or potentially even Internet protocols like RSS.

• Documents • Workflows • Business activities • Business rules • Schemas • Interfaces to connect to back end systems (Web service APIs)	• UI Screens • Data connections • Authorizations • Reports • Metrics

Table 1: List of application assets for composition

Expected Benefits of Composition, and How to Achieve Them

Deployment of enterprise applications should be tied to business benefits in the Triple-A sense (agility, adaptability, alignment). These benefits need to be demonstrated from two perspectives:

- **The Solution Provider Perspective (or Development Perspective) –** This is the perspective of the organization that builds an enterprise application. This might be an Independent Software Vendor (ISV), or a Systems Integrator (SI), or even an in-house IT department. The solution provider perspective is concerned primarily with benefits gained in activities relating to designing, implementing, and deploying enterprise applications.
- **The Solution Consumer Perspective (or User Perspective) –** This is the perspective of the organization that uses an enterprise application. Typically this is the business unit that commissioned the enterprise application. The solution consumer perspective is concerned primarily with benefits gained by the business after the solution has gone into production

The benefits of composition that can be reasonably expected in each of these two perspectives are listed here, along with some high-level best practices to achieve these expected benefits.

Alignment

Composition must promote alignment among key stakeholders. This could be internal alignment (align different groups within the enterprise), or external alignment (align with suppliers, customers and partners).

- **Solution Provider Perspective –** Solutions should align existing IT assets with the needs of business owners.
- **Solution Consumer Perspective –** It should be easy to assemble composite applications that align groups within a company, and also across organizational boundaries. Frequently this will require applications that support cross-functional processes and enable collaboration.

Best practices

- In-process documents should be stored alongside the business processes that they support. These might be explicitly defined processes, or ad-hoc collaboration. Policies should be in place to enforce document life cycle management.
- Business processes that are explicitly modeled (as opposed to ad-hoc collaboration) should be modeled using artifacts that are first class citizens of the development process. This implies that business process definitions are always in-synch with the solution, and that these process definitions can become the basis for alignment between business and IT groups.
- Business processes within an enterprise need to be aligned with business processes implemented by customers, suppliers, and partners. This requires a shared understanding and agreement on public processes between the different organizations. Public processes are those business processes that define the organization's behavior that is visible from the outside world; for example, interactions with trading partners. Private processes are those business processes that are internal to the organization. Figure 8 more clearly illustrates this.
- SLA information should be clearly defined for public processes. Integration requirements should be spelled out in the contract.
- There should be appropriate visibility across organizational boundaries. This requires exchanging appropriate data - not just transactional data, but also reports, analyses, status, and exception information from business processes. Reports and metrics should be in place to provide incentives that promote better performance – within the organization, and also partner performance.
- As data flows across organizational boundaries, data ownership should be spelled out, and data quality should be required, measured, and enforced.
- Integration across organizational boundaries should be done through adoption and commitment to relevant public standards. These standards might be for wire protocols (WS-* standards), or for document types (RosettaNet). Standards-based integration can usually lead to more maintainable solutions, and also to cost savings.
- Create deep visibility and alert capabilities to capture events that occur both within the enterprise, and outside it (for example, relating to trading partners).

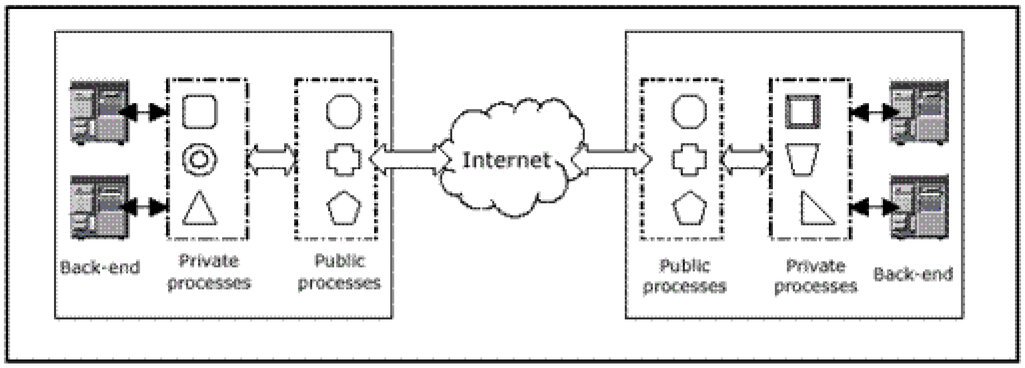


Figure 8: Differentiating between public and private processes

Adaptability

Composition must reduce time-to-response for the business when markets change.

- **Solution Provider Perspective**: The platform for composition should provide adaptive range. This means that it should be easy to externalize points of variability within the solution, so that those artifacts can be changed easily without ripple effects across the rest of the solution. For example, this might mean changing business rules, user interfaces, and business processes.
- **Solution Consumer Perspective:** It should be easy to reconfigure composite applications when the business needs to respond to changing market conditions.

Best practices

- The enterprise architecture should ensure that the implementation of public processes is cleanly separated from the implementation of private processes, as shown in Figure 8. This allows private processes to change without affecting integrations with trading partners. Conversely, it allows changes by trading partners to be isolated from internal systems.
- Avoid hard coding of policies, configuration, rules, or any other kinds of business metadata. Abstract legacy application functions into service interfaces that can be plugged into any kind of business process. Explicit business process models should be separate from service implementations.
- It is not necessary to capture all person-to-person interactions in process models, as the platform should enable ad-hoc document routing. It should be possible for business users to easily set these interactions up without a lot of involvement from IT, but in a way that enterprise document life cycle policies can be enforced.
- Avoid tight coupling of business processes to trading partners. Integration protocols, interfaces, and message formats should depend only on the public processes of your partners and not their private processes.
- Avoid tight coupling of systems, such as through point-to-point interfaces and connections. Reducing system dependencies increases the flexibility to make changes.
- Avoid tight coupling of user interfaces to backend systems. Systems may need to change, but it is hard to re-train users. Similarly it should be easy to change user interfaces in response to business needs, without large scale changes to back end systems.
- Abstract business process workflows from business service implementations. Also abstract business rule definitions from workflows. It should be easy to change a business rule

without changing process definitions, and it should be easy to change process definitions without changing the way a business service is implemented.

- Organizations should differentiate between those business processes that are commoditized, and those processes where innovation through new technologies and business models can provide competitive advantage. Organizations should seek packaged solutions and/or leverage industry standards work for commoditized processes. The organization should be willing to deploy custom or semi-custom solutions for those business processes where they seek advantages through process innovation.

Agility

Composition should reduce the time-to-benefit of enterprise applications by reducing the time-to-deployment of end-to-end solutions, reducing development and deployment costs, and by leveraging best practices. Also, when the business needs to respond to changes quickly or handle external disruptions smoothly, then it should be easy to modify composite applications to accomplish this.

- **Solution Provider Perspective:** Assets that make up a composite application should leverage established industry standards that make the solution quick to build and easy to assemble together.
- **Solution Consumer Perspective:** A deployed enterprise solution should enable quick decision making by business decision makers through real time data flows and business intelligence tools in the context of the automated process

Best practices

- Choose a technology platform that reduces the time to build and deploy composite solutions. This should provide a wide range of component types and containers, so that a rich variety of composite applications can be assembled depending upon the needs of the business process. The platform should also provide the tooling for composition.
- Decompose enterprise solutions into more granular software assets, so that solutions can be assembled, rather than developed from scratch.
- Make it possible to plug and unplug business services into the platform for assembling composite applications. This will require that existing applications be exposed via service interfaces, and that there is a mechanism for governance that enables client applications to register and bind to relevant business services. For those services where it is not necessary to track clients, publish service interfaces that can be subscribed to by composite applications.
- Enable real-time (or near real-time) information flows between the enterprise, its customers, suppliers, and partners (such as logistics providers). Make use of up-to-date data to automate collaborative business processes with customers, suppliers, and partners to reduce response times when change occurs
- Avoid tight coupling of business processes to specific applications. Business processes need to be fluid, but is hard to reconfigure an application beyond boundaries set by the application vendor.
- Reuse existing IT assets by connecting into what is already there, extracting more value from what is already there, and finally extending and evolving it.

Chapter2

The 2007 Microsoft Office System and Other Platform Technologies for Building Composite Applications

Introduction

Chapter 1 provided an overview of composite applications. This chapter will drill deeper into specific platform technologies for building and deploying composite applications.

Figure 5 in Chapter 1 shows that composite applications need to be deployed onto a platform that enables composition. Such a platform needs to provide the following:

- A set of containers for each of the tiers in Chapter 1, Figure 6,and associated component types that can be hosted within those containers.
- A set of capabilities that provide core services for composite applications to leverage.

This chapter will first walk through the capabilities of various platform technologies that make up the Microsoft platform, and then discuss the various containers provided by these platform technologies. Note that these composite business applications are called Office Business Applications (OBAs), as most user interactions are through client and server components of the 2007 Microsoft Office System. However, to build up an OBA for an end-to-end business process, capabilities from other platform technologies must also be leveraged, in addition to those from Office.

The 2007 Microsoft Office System Provides Capabilities to Host Applications

The new version of Microsoft Office – the 2007 Microsoft Office System - is a great platform on which to build composite applications. This system delivers not just the familiar set of Office clients (Word, Excel, InfoPath) but also several key capabilities as services that are delivered on both client applications and also on the server. These services are not just packaged together into a monolithic unit, but are very compartmentalized (as shown in Figure 1) and each has their own Web services interfaces.

Solution providers can take advantage of these services in their own applications – which means that core business processes can now be cleanly integrated into Office clients and Office servers. The composite applications that get assembled upon this platform are called Office Business Applications (or OBAs).

The high level capabilities in the 2007 Microsoft Office System are listed below. Each of these capabilities is a powerful feature when looked at individually - however it is the combination of these different technologies into a single integrated platform that makes composition practical.

This integration enables delivery and deployment of composite applications without an overwhelming increase in complexity in the overall platform, tooling and, architecture.

Web Site and Security Framework

SharePoint provides a common framework for creating a wide range of Web sites - such as team collaboration sites, intranet portals, and Internet Web sites. This is tightly integrated into the Active Directory directory services system to provide authentication and role-based authorization as well as enable federated trust relationships.

Open XML File Formats

This enables rich server-side processing of documents. With prior versions of Office, parsing the document using the object model required an instance of the client application to be started up.

Extensible UI

There are two channels for extension of the user interface – on the server, and on the client.

SharePoint provides a Web portal that can be used to host server side web applications that are composed from a library of building blocks call Web Parts. This library is packaged with a number of web parts out of the box, and can be extended by solutions providers for particular business processes. In addition, users can also personalize and extend applications once they have been deployed into SharePoint. Unlike prior versions of SharePoint, the Web Part framework is now integrated with .Net constructs in ASP.Net. Also the distribution infrastructure for Web Parts is simplified, so that you can wrap up a Web Part-based application, roll it out across the entire farm of the SharePoint site, and manage it in a very simple, central way.

Applications that have been built into Office clients, (Excel, InfoPath, Word) can also be extended by solutions providers, by using Visual Studio Tools for Office to create custom task panes, custom ribbons and so forth.

Business Data Catalog

A metadata repository to define business entities stored in backend data stores, to model relationships between entities, and to define actions permissible on entities.

Enterprise Search

To surface data from various enterprise sources through search.

Workflow

Windows Workflow Foundation (from the .Net Framework 3.0) has been directly integrated into SharePoint – which means that you get the core workflow runtime engine hosted directly in the SharePoint server. This means that a set of core scenarios around workflow are delivered out of the box such as the typical serial-parallel document approval routing across the organization. This set of workflows can also be extended by solutions providers. Workflows can be initiated either from within Office client applications or by server side events (such as a document getting saved). So a basic scenario might be an information worked creating a document in Word,, posting the document to a SharePoint site, thereby triggering an approval workflow.

Solution providers can extend these capabilities using either the SharePoint designer or Visual Studio. While Visual Studio is a developer tool, SharePoint Designer (SPD) is a tool for business analysts. SPD is a WYSIWYG design tool focused on creating SharePoint screens and workflows through a set of wizards that step the user through the process of creating new workflows. Workflow provides a powerful separation of interests from development. Developers can develop reusable modules called *activities* that may be deployed on the SharePoint server, and business analysts may then link these activities together to form workflows to solve specific business problems.

In addition to the core capabilities of the 2007 Microsoft Office System, there are also three major investment areas in terms of scenarios that make use of the base capabilities. These are listed below.

Enterprise Content Management

Manage diverse content, with one topology for Web, document, and records management. Support for document lifecycle management.

Business Intelligence

Server-based Excel spreadsheets, plus business intelligence (BI) components (dashboards, reports, Web Parts) built into the portal and connected to SQL Server Analysis Services.

Unified Communications and Collaboration

Support for unified communications integrated into the platform.

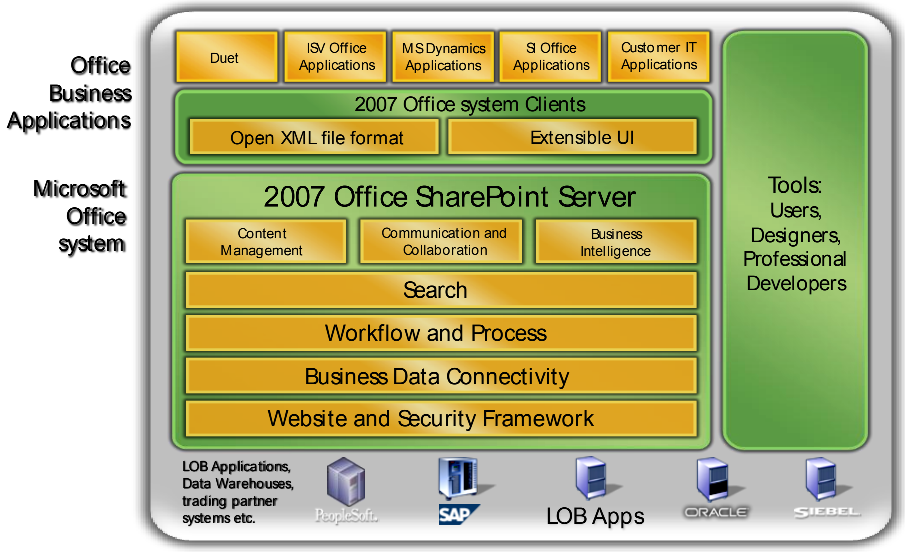

Office Business Applications
Microsoft Office system
Duet
ISV Office Applications
MS Dynamics Applications
SI Office Applications
Customer IT Applications
2007 Office system Clients
Open XML file format
Extensible UI
2007 Office SharePoint Server
Content Management
Communication and Collaboration
Business Intelligence
Search
Workflow and Process
Business Data Connectivity
Website and Security Framework
Tools: Users, Designers, Professional Developers
LOB Applications, Data Warehouses, trading partner systems etc.
PeopleSoft
SAP
LOB Apps
ORACLE
SIEBEL

Figure 1: Capabilities provided by the 2007 Microsoft Office System

SQL Server 2005 Provides Capabilities for Business Intelligence

In addition to core relational database capabilities, SQL Server 2005 provides a complete BI stack, with three core services: Integration Service, Analysis Services, and Reporting Services. These services give rise to the following set of capabilities, which enable composition around Business Intelligence solutions, that can be surfaced into OBAs deployed onto the 2007 Microsoft Office System.

Design

The BI Development Studio built into Visual Studio provides design capabilities for Analysis Services, Reporting Services, and Integration Services.

Synthesis

SQL Server Integration Services provides the ability to build and debug complex integration packages. This enables data acquisition from source systems, integration, data transformation, and synthesis. A usage scenario for this might be sourcing data from transactional systems to feed into a data warehouse

Storage

The Universal Dimensional Model (UDM) is a flexible and powerful way to model data and business rules that blurs the lines between relational and multi-dimensional storage. This is shown in Figure 2.

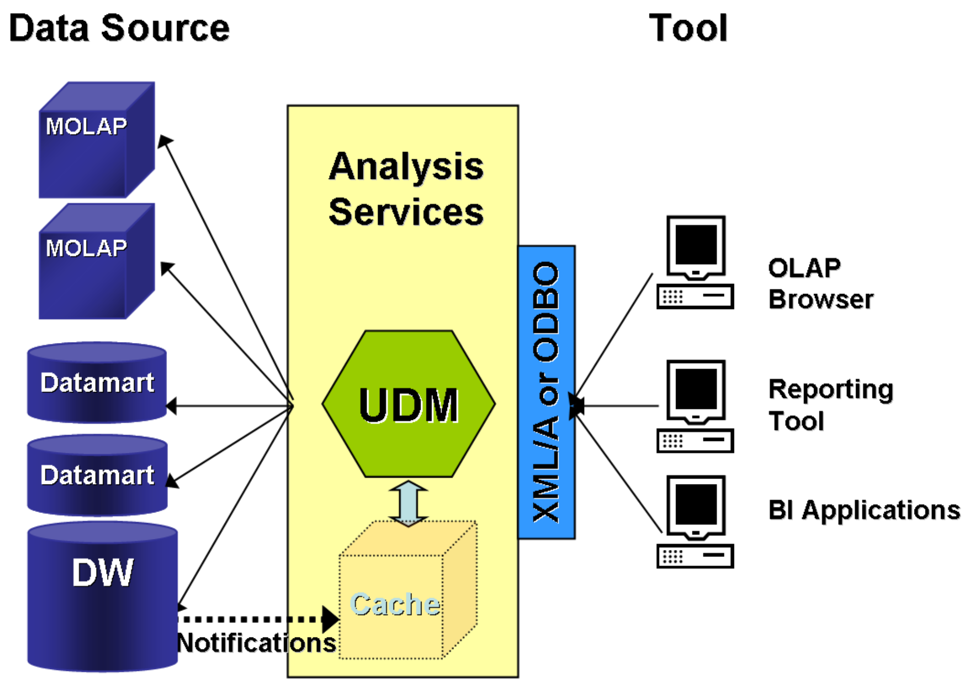


Figure 2: Scalable, high performance UDM server

Analysis

SQL Server Analysis Services provides an OLAP server, a data-mining platform with packaged data mining models that enables historical/predictive analytics, and hidden relationship detection. Solution providers can extend these models as seen in Figure 3, below.

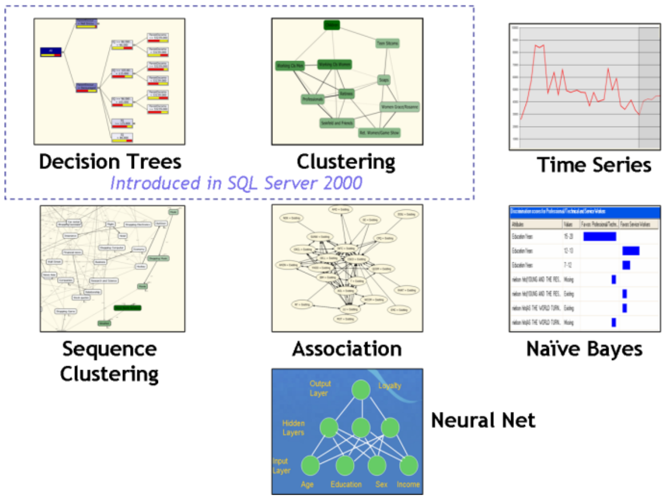


Figure 3: Analysis Services data mining models

Delivery

SQL Server Reporting Services enables traditional reporting, Web-based reporting, custom reporting, dynamic reporting, and is exposed as a Web service.

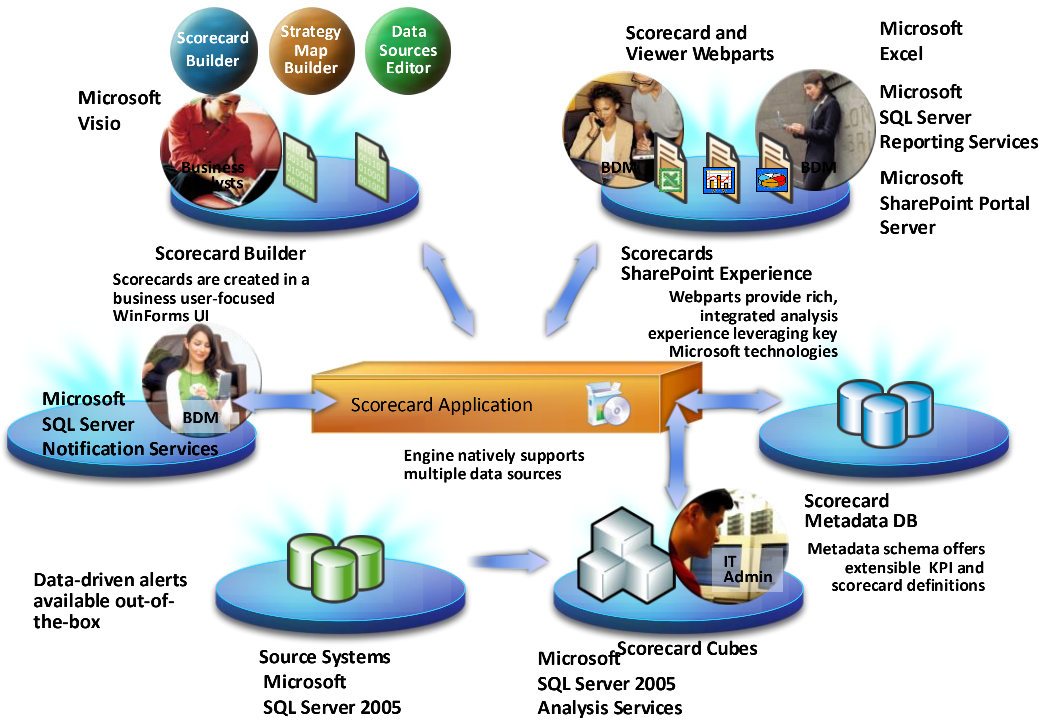

Scorecard Builder
Strategy Map Builder
Data Sources Editor
Microsoft Visio
Business Analysts
Scorecard and Viewer Webparts
Microsoft Excel
Microsoft SQL Server Reporting Services
Microsoft SharePoint Portal Server
BDM
BDM
Scorecard Builder
Scorecards are created in a business user-focused WinForms UI
Scorecards SharePoint Experience
Webparts provide rich, integrated analysis experience leveraging key Microsoft technologies
Microsoft SQL Server Notification Services
BDM
Scorecard Application
Engine natively supports multiple data sources
Scorecard Metadata DB
Metadata schema offers extensible KPI and scorecard definitions
Data-driven alerts available out-of-the-box
IT Admin
Source Systems Microsoft SQL Server 2005
Scorecard Cubes
Microsoft SQL Server 2005 Analysis Services

Figure 4: Business Scorecard server is a set of reporting tools that leverage SQL Server reporting services, and can be used to surface reports into the 2007 Microsoft Office System

Management

SQL Server Management Studio provides management capabilities for SQL Server database and analysis services cubes.

BizTalk Server 2006 Provides Capabilities for Messaging and Orchestration

The BizTalk Server 2006 provides a broad range of capabilities around messaging and orchestration, as shown in Figure 5. These capabilities are listed here.

Messaging

BizTalk provides the ability to communicate with a variety of other systems through messages. Pluggable adapters enable different kinds of communication, such as those to support different types of wire protocols and message formats.

Orchestration

Graphical design-time tools to create business processes, and an engine to run these that is built on top of the messaging capabilities. Typically BizTalk is used to orchestrate business processes that involve systems and not people.

Business Rules Engine

Design-time tools and run-time engine for complex sets of business rules.

Business Activity Monitoring

Capabilities for monitoring a long-running business process that is targeted to business users.

Business Activity Services

Allows information workers to set up and manage interactions with trading partners.

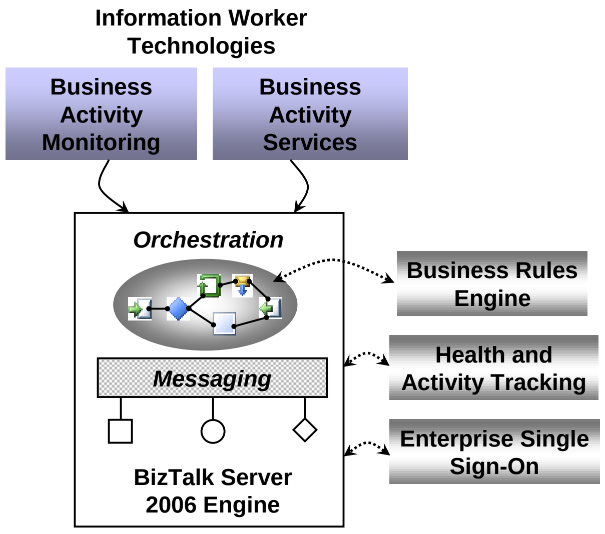

Information Worker Technologies
Business Activity Monitoring
Business Activity Services
Orchestration
Messaging
BizTalk Server 2006 Engine
Business Rules Engine
Health and Activity Tracking
Enterprise Single Sign-On

Figure5: BizTalk Server provides capabilities for messaging and orchestration

Office Business Applications are Composite Applications

OBAs t leverage capabilities from these various platform technologies to form composite applications. Let us now look at each of the tiers in Chapter 1, Figure 6, and examine the types of containers and component types that the platform provides.

Composition in the Presentation Tier

The first type of container in the presentation tier is the Microsoft Office client application such as Excel, Word, or InfoPath. These applications are customizable containers that can now more easily surface information and functionality from LOB applications through custom task panes, ribbons, and actions that present users with data and actions in the context of their current activity. The component type that can be hosted within these containers is the Open XML document. This is the new XML representation for Office documents which enables rich server-side processing of information stored within. This is shown in Figure 6, below.

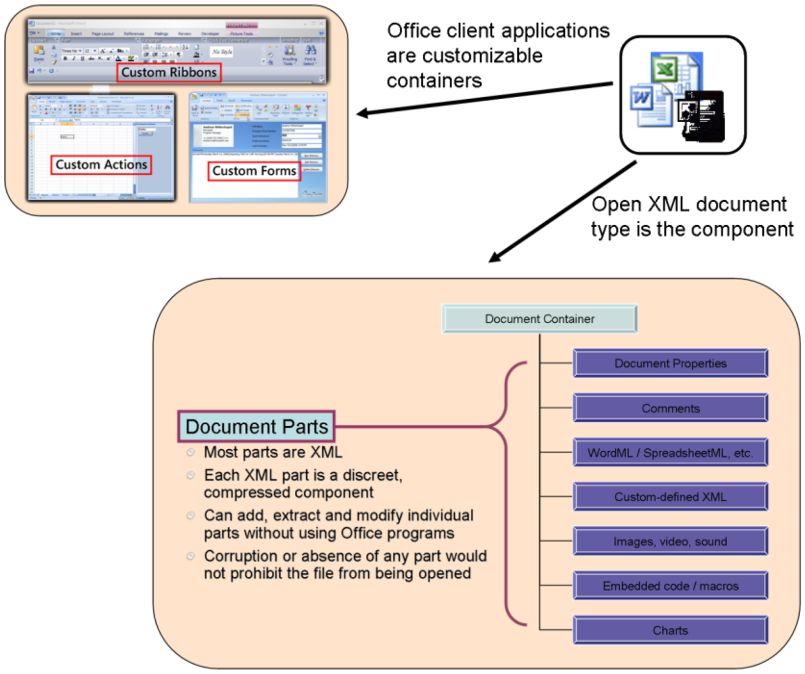

Custom Ribbons
Custom Actions
Custom Forms
Office client applications are customizable containers
Open XML document type is the component
Document Container
Document Properties
Comments
WordML / SpreadsheetML, etc.
Custom-defined XML
Images, video, sound
Embedded code / macros
Charts
Document Parts
Most parts are XML
Each XML part is a discreet, compressed component
Can add, extract and modify individual parts without using Office programs
Corruption or absence of any part would not prohibit the file from being opened

Figure 6: Office client applications are customizable containers for Open XML content

The second type of container is a Web portal, as enabled by Windows SharePoint Services (WSS) and Microsoft Office SharePoint Server (MOSS).

WSS v3.0 provides the following hierarchy of containers, as shown in Figure 7:

- **Farm** - Installation of one or more load-balanced Web servers and backend servers, with a configuration database.
- **Web Application** - IIS Web site, extended to use WSS, that can host site collections.
- **Site Collection** - Container for WSS sites, that exists within a specific content database. A site collection contains a top-level site, with optional child sites, and is the unit of ownership, secureability, and recoverability.

- **Site** - Container for child sites, pages, and content such as lists and document libraries.
- **Page** - Container for Web Part zones, and Web Parts.
- **Web Part Zone -** Container for Web Parts.
- **Web Part** - Components that display content on a page in modular form, and are the primary means for users to customize/personalize pages.

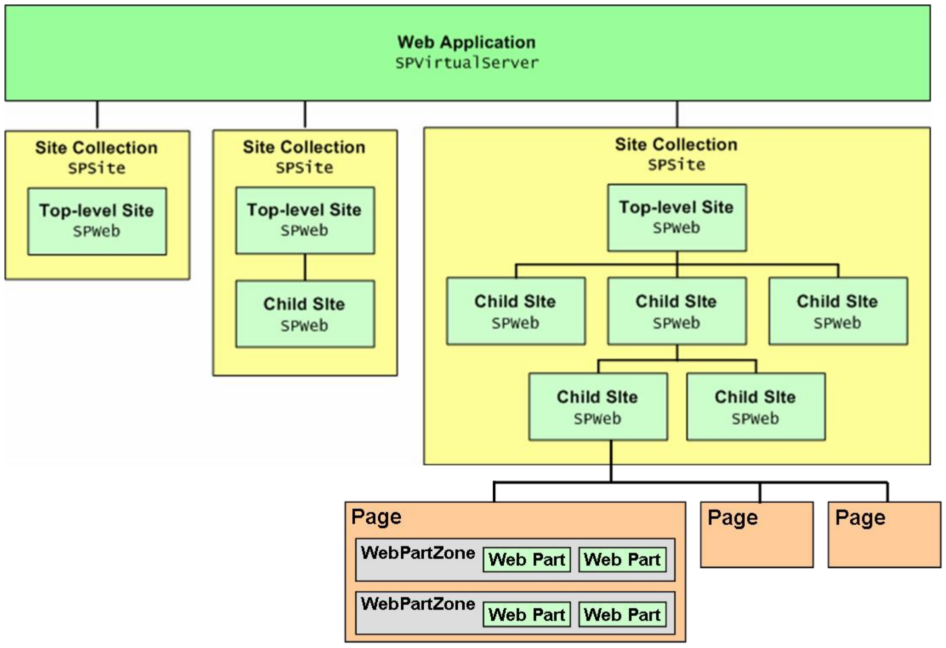


Figure7: Containers provided by Windows SharePoint Services

While WSS provides support for building collections of sites for team collaboration, MOSS enables portal functionality on top of WSS, with additional capabilities. For example, MOSS comes with capabilities for enhanced search, connectivity to business data stores, and for Business Intelligence. However a MOSS portal is a WSS site collection, and so is a hierarchy of WSS sites. MOSS also comes with a Web Part gallery that contains a rich set of Web Parts out-of-the-box, for example to surface Office Excel spreadsheets and charts. Solution providers can also provide their own custom Web Parts for application-specific or domain-specific functionality. These can then be uploaded into WSS. Information workers can then personalize pages, by adding Web Parts from the gallery, removing Web Parts from a page, or rearranging the layout.

Composition in the Productivity Tier

The 2007 Microsoft Office System provides a number of different ways to share information and collaborate upon it. For example, server-side storage provides containers for in-process documents and other kinds of heterogeneous content with version-control. This allows for collaboration in unexpected or un-planned situations. It is hard to capture all the complex human interactions in a business process management (BPM) system, as work usually never happens as planned. In traditional three-tier architectures, there is little support for this beyond storing in-transit documents on personal hard drives or e-mail servers, and it can sometimes be hard to decide which document is the correct version. However, the 2007 Office system can set up versioned document libraries to store documents at the intermediate stages of a business process. This improves manageability and also improves the likelihood of graceful recovery from unplanned events.

WSS provides the following types of storage:

- **List -** Container for items, which could be from built-in list types, or from custom list-types. Traditionally this has been the atomic unit of storage, but now a list can store huge amounts of data such as knowledge bases or Web content. There is also built-in index and query support.
- **Document library** - Special types of list that supports versioning and source-control of documents. For example, InfoPath forms may be stored in forms libraries, reports in report libraries, and other documents in document libraries.

Information workers can add document libraries themselves, and specify particular document templates to be used by all documents in those libraries. For example, a business analyst might use the InfoPath WYSIWYG design tool to create a form, and then use that form as a template to create a forms library. Whenever users create a new document within that library, this template will be used to create an empty form.

The 2007 Microsoft Office System also has in-built server-side support for Windows Workflow Foundation (WF), which is a technology that can be used to design and execute workflows. Figure 8 shows the different components within WF.

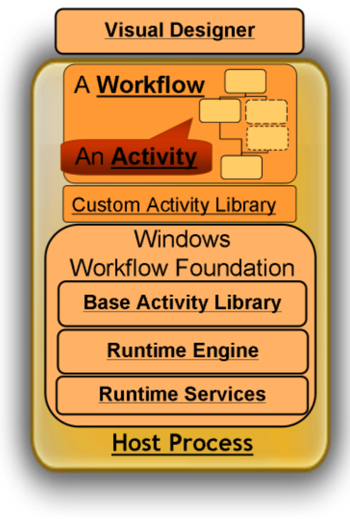


Figure 8: Windows Workflow Foundation

How does this enable composition? Workflow Foundation makes it possible to model workflows through easy-to-use graphical design tools and to represent workflow structures and data flow declaratively in an XML representation called XAML.

The WF run-time engine is hosted within MOSS and acts a container for business logic that can be attached to work items and documents in the form of workflows. These workflows may be associated with lists, document libraries, or with particular content types. They are started and completed by user actions, and are managed using WSS task lists. For example, workflow activities create and update task items as required, and users can track workflow progress through history tables. 2007 Microsoft Office System client applications are also workflow-aware. They can be used for workflow initiation, configuration, completion, and ad-hoc customization (forwarding/delegation).

Workflows can be associated with lists, document libraries, or particular content types. These associations to WF templates are tracked in a farm-wide workflow association table. WF templates are defined by XML metadata, and can include both workflow assemblies and forms. For example, when users create a document library, they can choose to associate that library with a particular workflow, and specify the conditions under which the workflow gets triggered (for example, a document in a given library might have been modified or created). This workflow could support a particular business process, or support document lifecycle management.

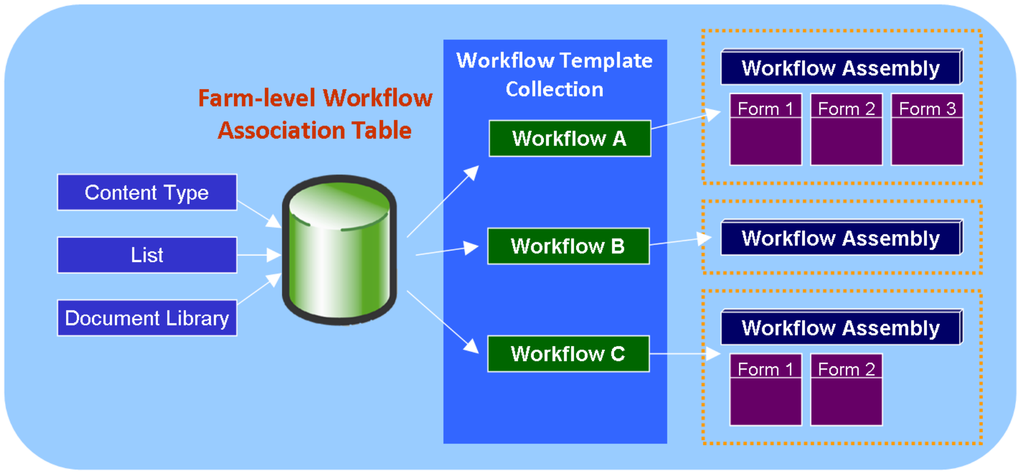


Figure 9: SharePoint Workflow architecture

Integration of the 2007 Microsoft Office System with WF provides a variety of benefits. Simple business processes can be automated in a way that is seamlessly integrated into the SharePoint UI. Users are empowered through self-service capabilities, such as support for a broad range of user-controlled document routing and tracking scenarios, in a way that reduces the involvement of IT staff in putting together simple applications. WF also provides vertical solutions providers with an extensibility point to deploy their own business rules and logic into the containers provided by the 2007 Microsoft Office System.

Finally, the productivity tier must also provide a lightweight way to create and publish information and reports. There is support in the 2007 Microsoft Office System to do this, which integrates into SQL Server Reporting Services, and provides the following components:

- **Report Center** - A template to create WSS reporting sites.
- **Report Library** - A document library with special support for storing reports.
- **Dashboard** - A WSS page assembled from reporting Web Parts

- **Report Viewer** - Web Part to view reports made available from SQL Server Reporting Services
- **Web Part** - to view Excel graphs and tables
- **KPI (Key Performance Indicator) Web Part and list** - Information workers can choose to source metrics for this in several ways.

A dashboard might be provided to users as a reporting template built around a specific business functions such as sales, marketing, or inventory management.

Composition in the Application Tier

Typically, structured business logic lives within the application tier. This might be an LOB application such as an ERP system or an orchestration across multiple systems such as a BPM (Business Process Management) system. The application tier will typically include both transactional systems and decision-support systems. There are multiple ways for OBAs to enable composition in the application tier, as well as ways to consume composite services exposed by other platform technologies (Microsoft and non-Microsoft).

The first level of composition in the application tier is through packaged activity libraries created using WF, and deployed into the 2007 Microsoft Office System. Previously, we discussed how workflows can be attached to lists, document libraries, and particular content types. Figure 10 shows how the WF run-time provides a container for application-specific activities that are packaged and deployed as activity libraries and on top of which workflows are assembled.

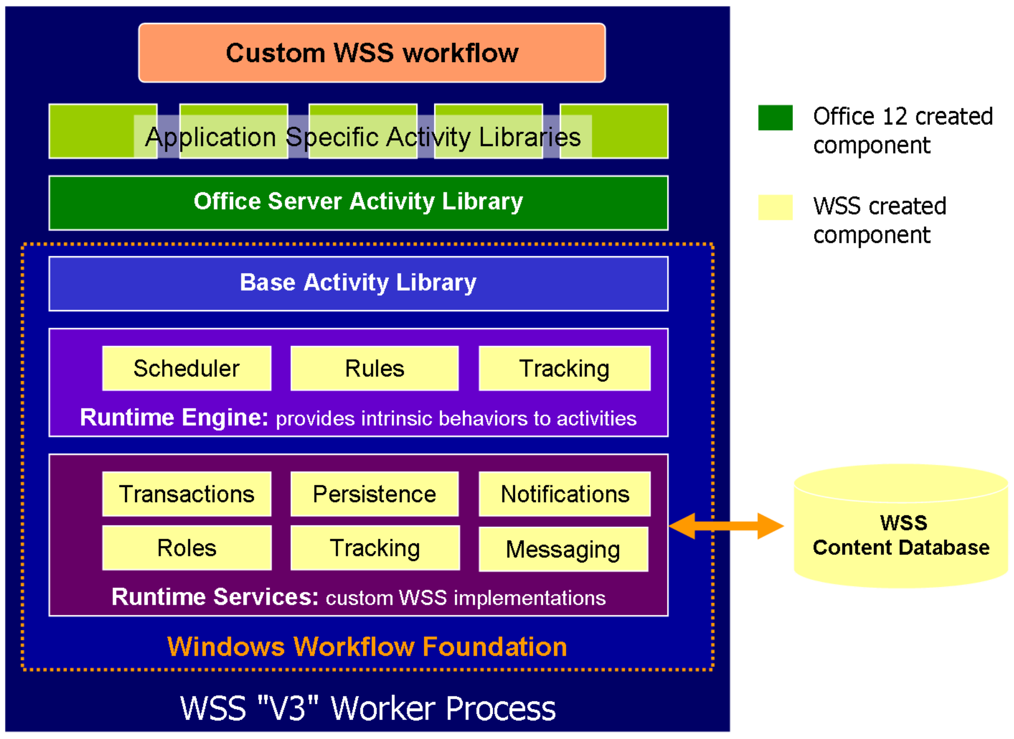

Custom WSS workflow
Application Specific Activity Libraries
Office Server Activity Library
Base Activity Library
Scheduler
Rules
Tracking
Runtime Engine: provides intrinsic behaviors to activities
Transactions
Persistence
Notifications
Roles
Tracking
Messaging
Runtime Services: custom WSS implementations
Windows Workflow Foundation
WSS "V3" Worker Process
Office 12 created component
WSS created component
WSS Content Database

Figure 10: Workflow hosting with WSS

Note that workflows are a set of activities, and while activities may be represented as workflows themselves, they may also be authored using code. This makes it possible for solution providers to package solution-specific activity libraries. This way the workflow designers can use the design tool to select from the library of activities and string them together into a workflow. Figure 11 shows how custom activity libraries can be packaged.

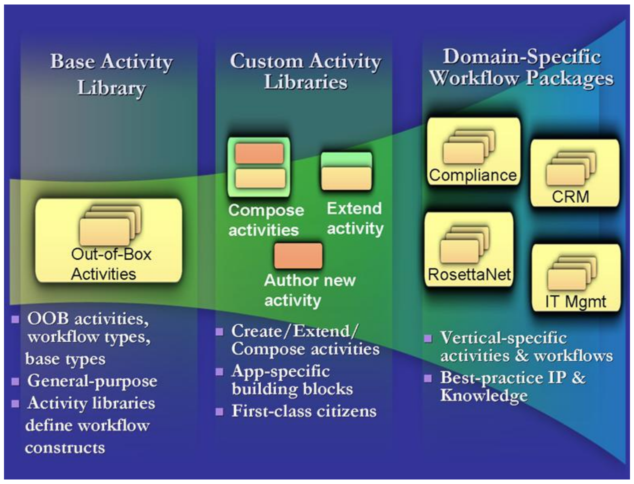

Base Activity Library
Out-of-Box Activities
OOB activities, workflow types, base types
General-purpose
Activity libraries define workflow constructs
Custom Activity Libraries
Compose activities
Extend activity
Author new activity
Create/Extend/ Compose activities
App-specific building blocks
First-class citizens
Domain-Specific Workflow Packages
Compliance
CRM
RosettaNet
IT Mgmt
Vertical-specific activities & workflows
Best-practice IP & Knowledge

Figure 11: Packaging Activities for Windows Workflow Foundation

The second level of composition in the application tier provided by the 2007 Microsoft Office System is through Excel Services. This is a server-side Excel calculation engine that is built into SharePoint Server. It provides browser access to live, interactive spreadsheets that are deployed on the server, and also Web service access to server-side Office Excel calculations. With Excel Services, existing Excel power users can now provide server-side application logic in a way that is familiar to them. MOSS is now a container for application logic, as shown in Figure 12.

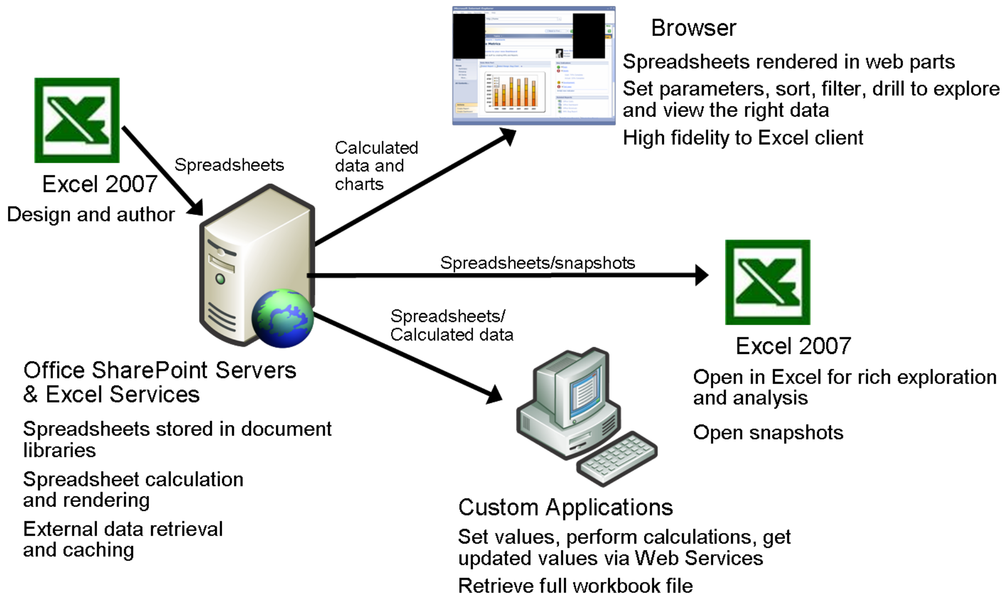

Excel 2007
Design and author
Spreadsheets
Calculated data and charts
Browser
Spreadsheets rendered in web parts
Set parameters, sort, filter, drill to explore and view the right data
High fidelity to Excel client
Spreadsheets/snapshots
Excel 2007
Open in Excel for rich exploration and analysis
Open snapshots
Spreadsheets/ Calculated data
Office SharePoint Servers & Excel Services
Spreadsheets stored in document libraries
Spreadsheet calculation and rendering
External data retrieval and caching
Custom Applications
Set values, perform calculations, get updated values via Web Services
Retrieve full workbook file

Figure12: Excel Services in the 2007 Microsoft Office System

A third way for OBAs to enable composition in the application tier is to consume composite services exposed by other platform technologies. The 2007 Microsoft Office System can be plugged seamlessly into a services-oriented architecture (SOA). If the enterprise has a services backbone already under development, then these interfaces can be consumed from the 2007 Microsoft Office System. This can be done in a couple of ways. The first is by invoking Web service interfaces in activities that have been built into workflows deployed into the productivity tier. The second is through the Business Data Catalog (BDC), which is described in the next section.

Composition in the Data Tier

The 2007 Microsoft Office System also comes with a Business Data Catalog (BDC) that runs within the server as a shared service. It can read data from multiple types of data sources — databases, analysis services cubes, and Web services — and then surface this back into the portal through SharePoint tables and lists. This acts as a metadata repository for descriptions of business data entities and their attributes and for mappings of these entities back to data stores within the enterprise, as shown in Figure 13.

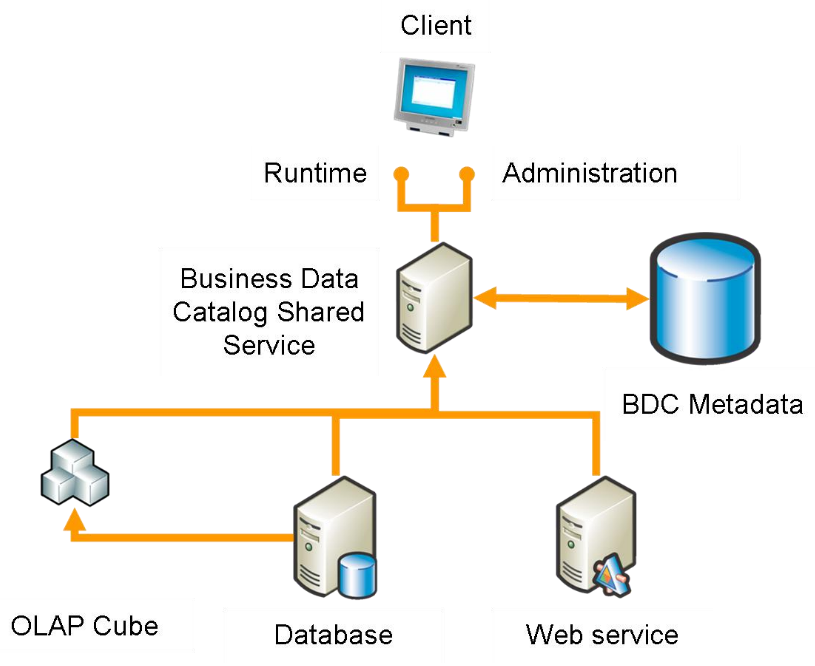


Figure 13: Business Data Catalog (BDC)

While the BDC cannot be used to create an entity that maps across multiple data stores, it is possible to define relationships that link entities such as parent-child relationships. So, the BDC can be used to create lightweight linkages between data entities across the enterprise—almost like an enterprise thesaurus. BDC-defined entities can then be plugged back into the 2007 Microsoft Office System, for example in SharePoint lists. This allows users to compose pages with linkages to backend data, and to traverse data tables by following relationships between entities.

While the BDC enables data composition, it provides only read-only access to data. However, you can use the BDC to model actions that can be taken on a data entity, where an action is defined by a name, a URL, and a set of attributes from the entity definition to be passed back to this URL. This can then be used from within a drop-down menu on a SharePoint list. The URL can correspond either to a Web service or to a server-side document such as an InfoPath form with code behind to pre-populate the form from the context that gets passed in from the BDC. Information workers can then use the portal to create lightweight applications in SharePoint with tables and lists that surface BDC data and actions.

Summarizing Capabilities of the Microsoft Platform that Enable Composition

Figure 14 shows how some of the capabilities of the 2007 Microsoft Office System map to the tiers in Chapter 1.

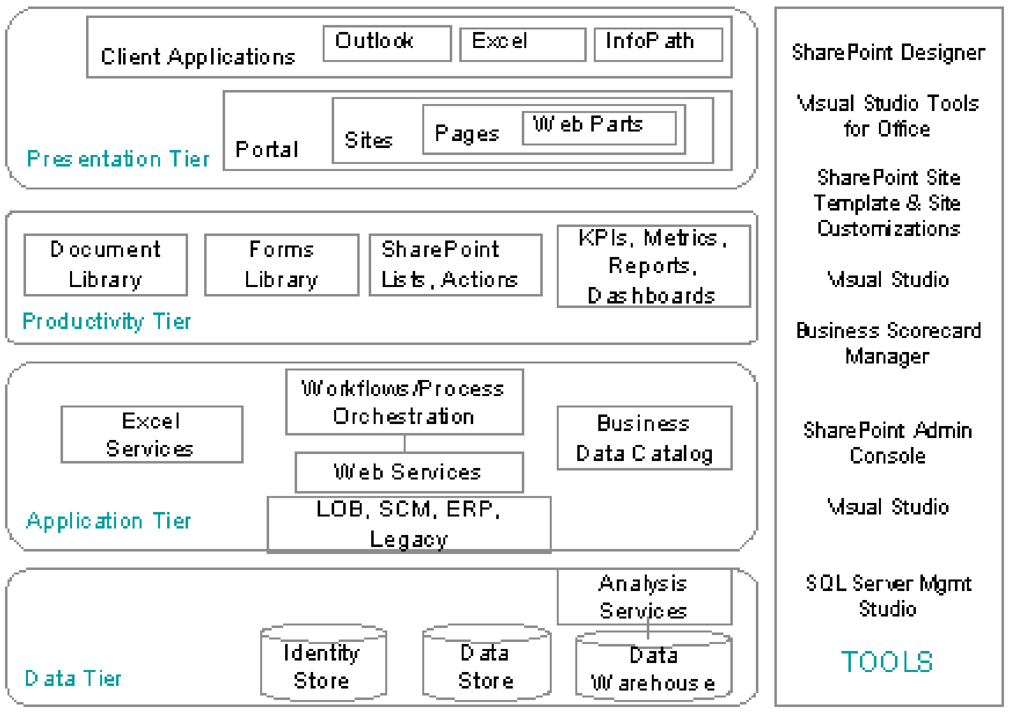


Figure 14 Capabilities of the 2007 Microsoft Office System application platform, mapped to tiers.

Table 1 below provides a list of asset types that are candidates for composition.

• Open XML Documents	• Web Parts
• Workflows	• Dashboards
• Business activities	• Sites
• Business rules	• Pages
• Schemas	• Data connections
• Metrics	• Authorizations /claims
	• Reports

Building an Office Business Application

There are two steps to building a standard business process as an OBA.

1. Design the metadata and packaged application code and package them.
2. Deploy the package onto production systems.

Build a Process Pack

1. Build user interfaces for document-centric processes into Microsoft Office client applications, using Visual Studio Tools for Office (VSTO).
2. Build WSS sites with task-specific Web Parts, pages, dashboards, lists, and document libraries where each site is designed for a particular business function or process. These sites can be used to create site templates for packaging a standard business process into an OBA solution.
3. Use Workflow Foundation to wire together lists and document libraries in the sites, with server-side rules and business logic for server-side processing of in-process documents. Package these workflows into assemblies for deployment.
4. Define touch points from workflows in the OBA to backend LOB systems. The process pack metadata should include the Web services interfaces for this integration. The actual connections will need to be made during the deployment phase.
5. Define BDC entities for data connections required for cross-functional processes built into the OBA.
6. Add decision support by defining metrics, reports, dashboards, and Excel charts and tables to be used by the WSS sites.
7. Package solution as a process pack (WSS site templates, WF assemblies, Office documents, and so forth) using the component types in **Error! Reference source not found.**.

Deploy Process Pack to Production Systems:

1. Deploy WSS sites to the SharePoint server on the production system.
2. Configure connections from workflows to LOB applications using Web services – or other custom adapters.
3. Configure data connections by connecting BDC data entity definitions to actual data stores.
4. Provision users.

Chapter 3

Deploying Office Business Applications in the Enterprise

Introduction

Chapter 1 provided an overview of composite applications, and Chapter 2 walked through platform technologies to enable composition. This chapter will drill deeper into some architectural patterns for deploying composite applications based on 2007 Microsoft Office System (OBAs) in the enterprise. These patterns will then be leveraged in subsequent chapters to provide guidance specific to particular industry verticals.

One approach to deploy an OBA in the enterprise is as follows:

1. Build team collaboration sites to host local documents and processes.
2. Connect multiple departments.
3. Connect business processes to LOB applications.
4. Add data connections for cross-functional processes.
5. Connect business processes to systems "at the edge."

The best way to do this is to follow a pragmatic approach. This means that the OBA being deployed is really a composite application to support a single business process at a time, and the goal is not to try to re-architect all the backend systems in one big-bang. The rest of this chapter makes this assumption.

Step 1: Build Team Collaboration Sites to Host Local Documents and Processes

SharePoint sites should be set up at the departmental level to enable team collaboration, as shown in Figure 1. These sites will have document libraries to store in-process documents. Information workers within the team will have their own personalized pages, customized from the templates available on the team site.

Note that this is a logical view of the architecture, as all of these departmental sites would not typically be running on separate servers. Instead, multiple sites could be running on a single server in line with Figure 2Figure, and the physical architecture (that is the deployment landscape for SharePoint servers) would be chosen based on other factors like load, availability, and the teams' geographic dispersion.

In-process documents are stored in document libraries and are associated with workflows that get invoked whenever a document is created or edited. Such a workflow might run validation rules on documents; apply approval policies and actions to the data; cleanse, validate, or filter the data contained within; or update backend systems.

In addition to business process workflows, in-process documents undergo a lifecycle of their own – from authoring and collaboration, through management and publication, to archiving or destruction. Whenever a document reaches one of these stages, the appropriate workflow can be set to be triggered, such as for managing the archival process.

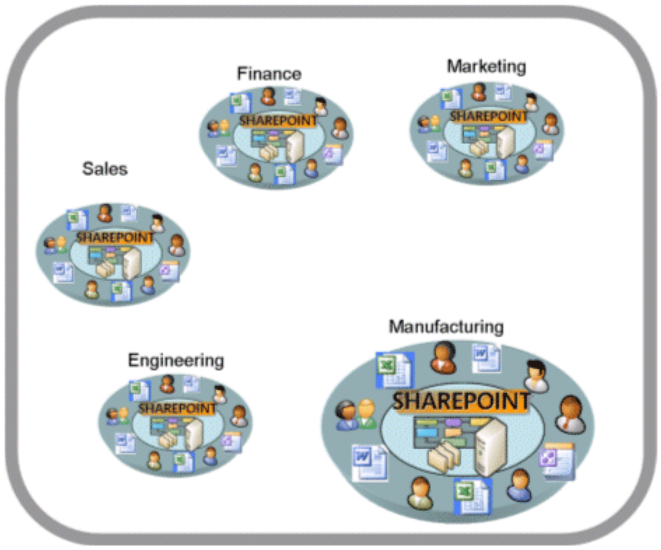


Figure 1: Departmental sites for team collaboration, with folders for shared documents and tasks

Pattern: Segment user personas and work scenarios, and select user interface technologies appropriately

The user experience can cover a wide range of work scenarios. These work scenarios typically string together a set of activities where a user is interacting with a particular type of document, screen, or other type of interface. However, it is hard to build and maintain all the individual user interfaces that make up day-to-day work scenarios in a way that these can all be personalized by different users. Segment user personas and work scenarios, and choose user interface technologies appropriately for each segment. For example:

- Processors and synthesizers of business documents.
    - Capabilities required Support for document-centric workflows and processes.
    - Proposed user experience Provide document processing with Office client–based applications. Provide server-side storage of these documents to avoid proliferation of document copies (using SharePoint, for example), and process the information stored within these documents (such as Open XML) in server-side application logic (workflows deployed into SharePoint). In effect SharePoint becomes the role-based access point into the enterprise
    - There should be support for different types of collaboration – both ad-hoc (informal document sharing) and structured approval processes. To facilitate server-side processing, make an association made between the content within the document and

XML schemas of business entity models that server-side application logic can access. For example:

(i) Target data entry with InfoPath forms.

(ii) Target data analysis with Excel spreadsheets.

(iii) Target proposals with Word documents.

- Business decision makers:
  - Capabilities required: Support for business intelligence and insight.
  - Proposed user experience: Ability to search for and view business performance metrics in the form of spreadsheets, reports, and dashboards through browser-based interfaces.
- Users of high-volume data entry systems (power users):
  - Capabilities required: high volumes of data entries, keyboard shortcuts, and rich navigation.
  - Proposed user experience: thick clients.

Step 2: Connect Multiple Departments

As shown in Figure 2, business-process models coordinate activities both within a team and across departments. Within a department, business processes can be modeled using WF activities that are deployed into the SharePoint server supporting that department. Coordination of activities across departments can be accomplished using collaboration processes that manage both the lifecycle of individual business entities (orders) and also the lifecycle of business processes (procure-to-pay).

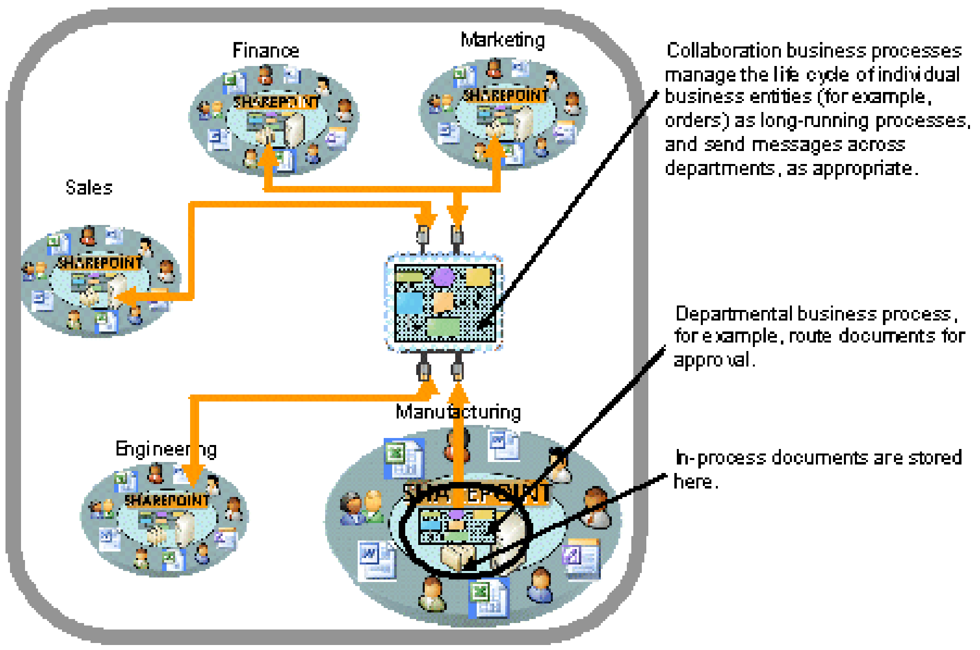

Finance
Marketing
Sales
Engineering
Manufacturing
SHAREPOINT
Collaboration business processes manage the life cycle of individual business entities (for example, orders) as long-running processes, and send messages across departments, as appropriate.
Departmental business process, for example, route documents for approval.
In-process documents are stored here.

Figure 2: Business-process models coordinating activities within a team, as well as across departments

The inter-departmental business processes could either be located centrally in IT data centers or closer to the information workers within the departmental servers. For example, long-running workflows running centrally might be running on Microsoft BizTalk Server, whereas departmental processes would be running in WF workflows within the SharePoint server.

Pattern: Use workflow to model business processes that coordinate the activities of automated services and humans

To gain flexibility, we need to externalize our business process workflows from our enterprise applications' business services. Figure 3 shows how this might look.

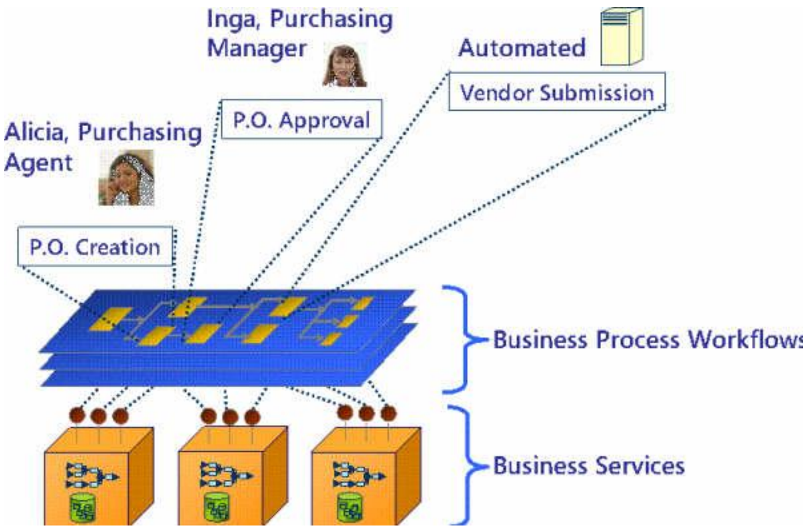


Figure 3: Separating workflows from services

A natural question to ask is whether LOB systems already installed (such as an ERP system) have all necessary business processes integrated? If so, is there any need for further externalization of business processes in this manner? It is not always easy to synchronize business processes across multiple systems. For most organizations, data pertaining to business processes is kept in multiple systems. Either these are legacy systems or they were acquired through business acquisitions. Also the business will frequently want to introduce new business models, such as re-selling sub-assemblies to new channel partners or selling new modules assembled from components procured from around the world. In many cases, the existing LOB systems will not handle these new business models out of the box and there will be a lot of custom development required.

The solution to this is to decompose business processes as units of work, and represent these as activities. This decomposition need not be top down – activities can be added as needed – and over time you will build out a library of these software assets. Also, these assets should be self contained – with encapsulated logic and associated metadata required to configure their behavior. Examples of such activities could be:

- Document-centric –. Send notification email, move document to approved document library, publish document.
- Domain-specific – Expedite purchase order, generate lead, create production order.

Workflows then become re-configurable organization of those work units, or activities. The natural representation of data flow between activities in a workflow is an XML document. This mirrors the real world, where activities represent units of work being performed by different people or systems, and documents are being exchanged between them. Since 2007 Microsoft Office System documents use the Open XML document format, you can create workflows that closely mirror the real world.

Pattern: Segment business processes and model their workflows differently

Most business processes can be segmented into strategic, tactical, and operational levels

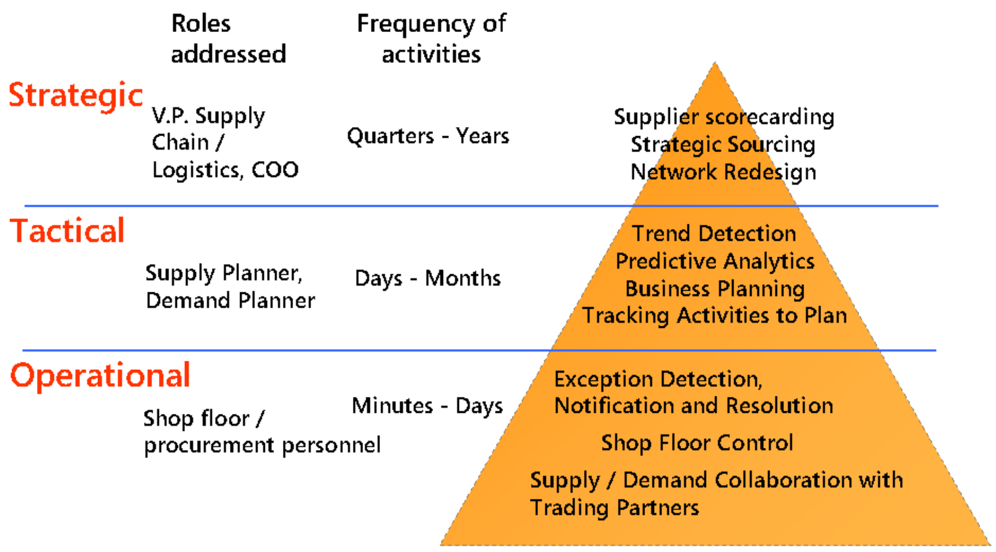


Figure 4: Hierarchy of enterprise business processes

Each of these different kinds of business processes has different technology and architectural needs. Enterprise solutions at different levels in this hierarchy have different functional requirements and also make use of different technical capabilities of the underlying platform. For example:

- Strategic solutions will require more analytics and business intelligence.
- Operational solutions will require more real-time integration (such as, through messaging and collaboration) and instrumentation and processing closer to the edge of the network (such as through RFID services and smart devices).
- Tactical solutions will need both business process management tools and reporting tools to manage operational workflows.

While business processes in all three of these levels will require workflows to be modeled and then managed, the types of workflow model for each level may be different. For example, operational processes might be best modeled using state machines where outside events or entities are in control and where processes can be cancelled or modified at any time. Strategic processes might be best modeled using sequential workflows where the workflow is in control and there are well-defined sequences of activity. Tactical processes might require a mix of sequential workflows and state machines. In some cases, it may be better to use business rules instead of either sequential workflows or state machines, especially when the set of actions is determined by the interaction of a number of independent rules. For both tactical and strategic processes, consider using long-running and state-driven workflows.

Figure 5 illustrates that workflows can organize activities in two ways – sequentially, as in a flow chart or as a state diagram.

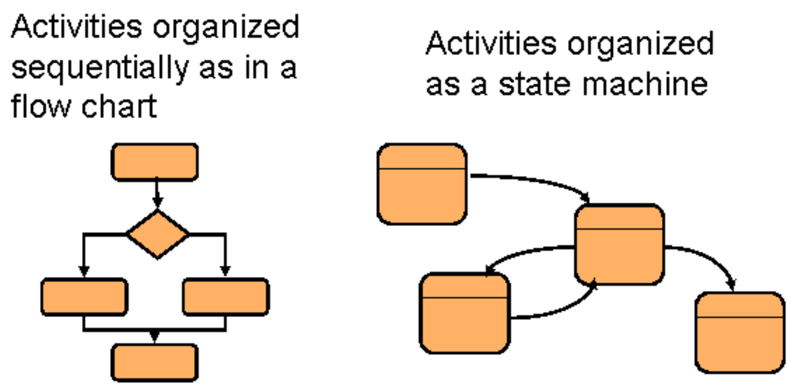

Activities organized sequentially as in a flow chart
Activities organized as a state machine

Figure 5: Types of workflow

The primary difference between these two styles is how control gets passed from one activity to another. For sequential workflows, this is determined by the workflow itself. However, for state machines, this control gets passed based on events triggered externally to the workflow, although the state machine controls the set of choices.

Pattern: Workflows should be decentralized

In a world of globalization, specialization, and the virtual enterprise, there is increasing pressure on IT to enable decentralized decision making and empower the individual information worker. In such a world, there are no useful, omniscient views of business processes within an organization. It would take too long to document these, and it would add little value. That is why traditional BPM technologies have not worked out very well, as they emphasize modeling of business process to a high level of detail and use these models to drive end-to-end flows of information across applications, systems, and information worker activities.

However, workflow and business process modeling technologies can be extremely useful for building composite solutions – if used appropriately. Workflows should be fluid – easy to assemble and easy to change. Workflows should be local – ownership should live close to the activities that are being coordinated. It is all right for small snippets of process to be modeled independently of each other. Another important need is for built-in flexibility in the process. This does not mean attempting to model every possible decision point and every possible outcome. Instead, it means empowering information workers to choose the business process, or workflow, that they need to participate in depending upon their current situation. This could be achieved by allowing users to explicitly initiate a workflow from their client applications (as is possible with Office client applications like Excel), or by controlling the flow of documents on the server (as is possible through SharePoint).

In practice, what does it mean to say that workflows need to be decentralized? Consider the following levels of business process, and imagine workflows running at each of these levels and operating on data models that are aggregated to that level.

- Business processes that operate on the following data streams, when moving from the factory floor to the center of the enterprise:
  - Smart devices on the factory floor (sensors, actuators, controllers, operator terminals).
  - Edge servers that process (filter, aggregate, transform) data streams from smart devices on the factory floor.
  - Factory execution systems (MES, SCADA, Adv. Proc. Control).
  - Factory business applications (planning, scheduling).
  - Enterprise business applications (ERP, CRM, SCM).
- Business processes that relate to the following levels of decision making:
  - Corporate decision making.
  - Divisional decision making.
  - Factory decision making.
  - Work group decision making.

Step 3: Connect Business Processes to Line-Of-Business Applications

Service orientation is one of the ways to expose current business applications into an OBA, as shown in Figure 6.

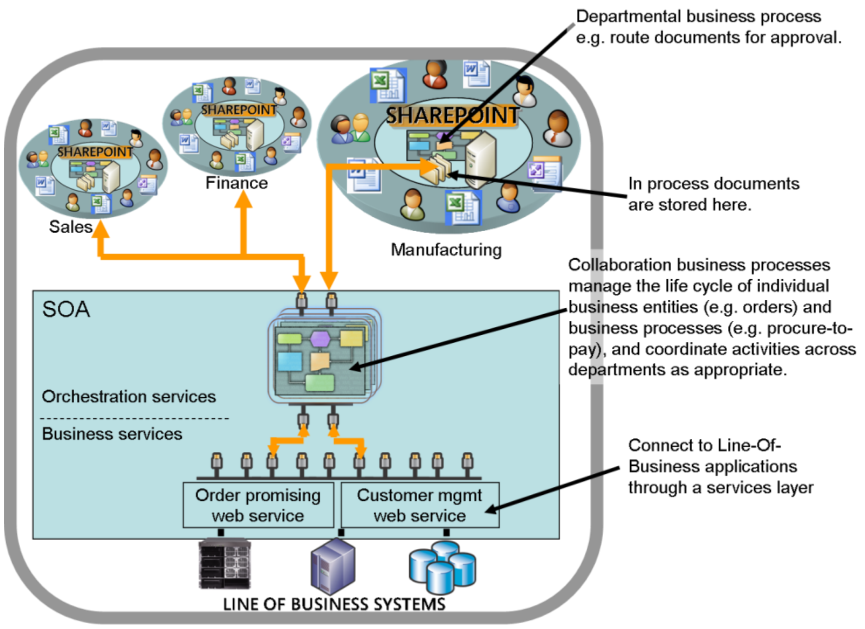

Departmental business process e.g. route documents for approval.
SHAREPOINT
SHAREPOINT
SHAREPOINT
Finance
Sales
Manufacturing
In process documents are stored here.
Collaboration business processes manage the life cycle of individual business entities (e.g. orders) and business processes (e.g. procure-to-pay), and coordinate activities across departments as appropriate.
SOA
Orchestration services
Business services
Order promising web service
Customer mgmt web service
Connect to Line-Of-Business applications through a services layer
LINE OF BUSINESS SYSTEMS

Figure 6: Provision OBAs in departmental sites for team collaboration, coordinate activities with business-process models, and connect to LOB systems through a services backbone

In that sense, SOA promotes modularization in the application tier – which is a basic requirement for composition – and is why OBAs and SOA are complementary solutions. This allows the assembly of new cross-functional business applications that extend beyond the boundaries of current applications.

SOA is not the only way to connect LOBs to OBAs. A services backbone can also be exposed using other integration technologies, such as custom adapters.

Pattern: Wrap existing IT systems with a services layer mapping to specific capabilities to be surfaced into the composite application

Service orientation enables reuse of existing IT assets by wrapping them into modular services that can be plugged into any business process that you design. The goals for doing this should be:

- Connect into what is already there – Layer business process management, collaborative workflows, and reporting on top of existing IT assets.
- Extract more value from what is already there – Enable existing applications to be re-used in new ways.
- Extend and evolve what we already have – Create IT support for new cross-functional business processes that extend beyond the boundaries of what the existing applications were designed to do.

For years, software development has focused on how to best reuse the code that we write. Ultimately, people and businesses want long-term return on their short-term investments in code. How do we do this?

1. Cleave applications apart into services:
    a. Disentangle data
    b. Separate functionality
    c. Add messaging between
2. Wrap applications with services:
    a. Decide the functionality to wrap – Often a subset of functionality is OK, depending on the scope of the business process.
    b. Create messaging.
    c. Tap into business logic or user interface of application.
3. Build new solutions with services.

Pattern: Follow the four tenets of service orientation while designing services

The four tenets of service orientation are as follows.

- **Boundaries are explicit** – Services need to interact across service boundaries, but crossing service boundaries may be costly. Therefore it is important to know your boundaries. Keep service surface areas small, avoid RPC interfaces, and avoid leaking implementation details outside a service boundary.

- **Services are autonomous** – Ideally services should be stable, but business needs change frequently. The space between services changes more frequently than the service boundaries themselves. Avoid assumptions on the environment into which the services are deployed. Design, deploy, and manage services independently from one another. Communicate only through contract-driven messages and policies.

- **Services share schema and contract, not class** – Service consumers will rely upon a service's contract to invoke and interact with a service - and will insist that the contract remain stable over time. However, changing business needs will force change upon a service. Therefore service interaction should be based solely upon a service's policies, schema, and contract-based behaviors. Ensure stability of services (public data, message exchange patterns, policies). Form explicit contracts designed for extensibility. Version services when change occurs. Avoid blurring the line between public and private data representations.

- **Service compatibility is determined based on policy** – Services need to be compatible with each other. Not just structural compatibility (public data, message exchange patterns), but also compatible with other semantic capabilities that are expressed through configurable capabilities and service levels. Operational requirements for service providers should be manifested in the form of machine-readable policy expressions.

Pattern: Segment the services that you plan to create – not all services are created equal

What kinds of services should you create? One way of organizing services is as follows:

1. Infrastructure services.
2. Data services - Simple atomic operations on an entity.
3. Activity services - Coordinate data services for business process execution.
4. Process services – Long-running business processes, possibly complex workflow and human interaction.
5. Event services - Notify subscribers of events

It is not necessary to create all of these types of services. A better way would be to expose services incrementally, in the context of well-defined projects that create business value. Figure 7 shows a set of sample business services that could be deployed in the enterprise. However, with the rise of Software as a Service (SaaS) you cannot assume that services will be co-located or developed in-house. Also it may be necessary to build a hierarchy of services through aggregation and composition.

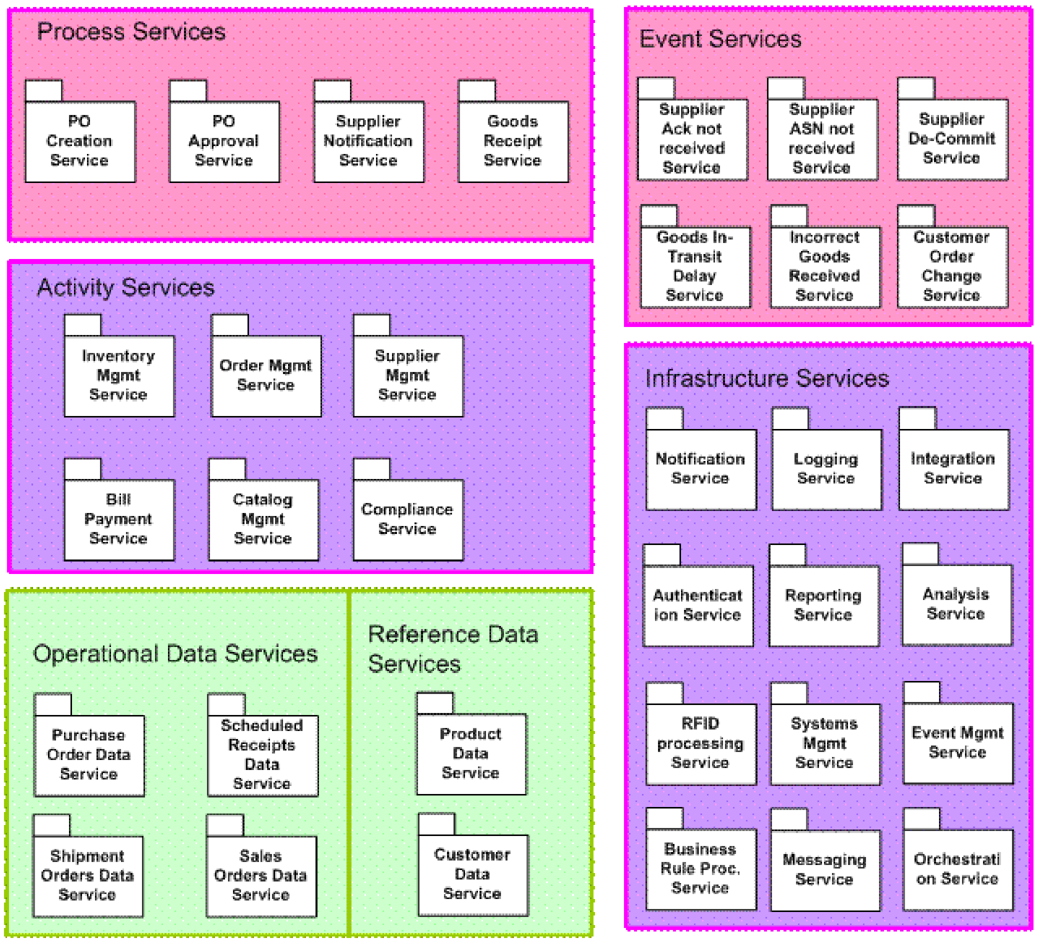


Figure 7: Sample business services that could be deployed in an enterprise

Step 4: Add Data Connections for Cross-Functional Processes

The BDC (Figure 8) can be used to connect backend data stores to the 2007 Microsoft Office System to surface data into SharePoint lists and Web parts. This makes it possible to build composite applications for cross-functional processes in the SharePoint portal, using a combination of BDC, SharePoint lists, and Workflow. For example, the BDC can be used to define entities that have a parent-child relationship (such as order header, and order details), and SharePoint lists could display them. It would be possible by following the parent-child relationships to drill down from the list displaying header information, to the corresponding details information. Furthermore, actions can be modeled in BDC metadata. This means that these actions can be surfaced as menu items on the SharePoint list and selecting a dropdown from this menu will mean that context from the currently selected row will be passed into the URL defined for the action.

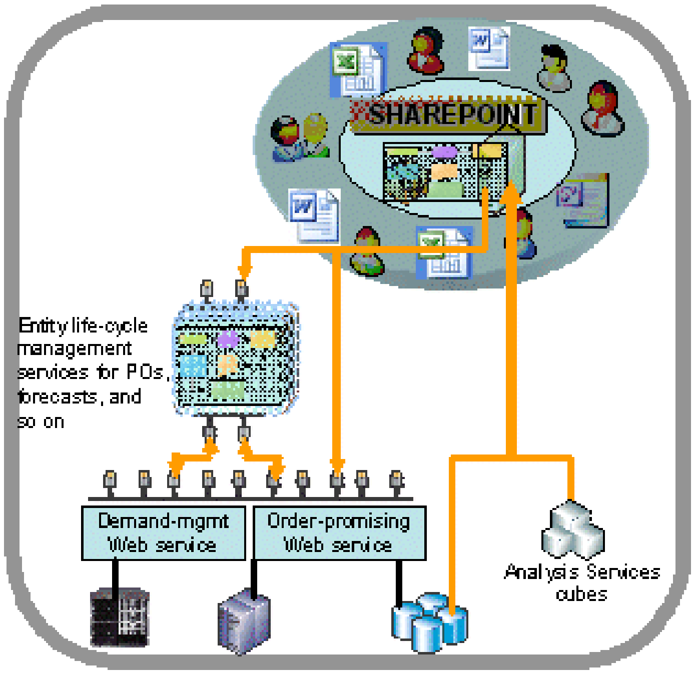

SHAREPOINT
Entity life-cycle management services for POs, forecasts, and so on
Demand-mgmt Web service
Order-promising Web service
Analysis Services cubes

Figure 8: Adding data connections for cross-functional processes

Pattern: Break away from the model of monolithic data stores, and adopt a model of federated data.

Over the last 30 years, enterprises have moved towards data consolidation. That is they have moved from multiple data-processing systems, to OLTP databases, to single ERP instances. However, it is difficult to ever get to a single ERP instance, and maintain it. Some of the reasons for this lie in the normal way that business is conducted nowadays. For example, mergers, acquisitions, globalization, matrixed organizational structures, and outsourcing all have a tendency to pull data apart. A combination of service orientation and the BDC provide a way to manage this distribution of data. SOA discussions typically revolve around messaging, but there are important issues around how data is used by services such as: how does data flow between services; how are messages defined; what data is shared; how is data inside of a service different from data outside a service; and how is data represented inside and outside services.?

Pattern: Treat data inside services differently from data outside services

Typically, data outside services should just be messages that are passing from one service to another, while data inside a service is private to that service, bounded by transactions, and encapsulated by service code. Usually, transactions take a database from one consistent state to another. However, ensuring data consistency across services is hard. For example, services deal with data in the present, but data on the outside exists in the past. Maintaining shared collections of data across services can be hugely challenging and distributed transactions are not the answer for a loosely-coupled and service-oriented world. The implication is that data must be segmented into different types and treated differently. For example:

- **Reference (or Master) Data (such as a product catalog)** – This data can be used to create service requests in a format interpretable by all parties. This data can be replicated and cached multiple times, because it doesn't have to be 100% consistent. For example, it may be okay to order parts in April from a March catalog. However reference data does need to have a stable identifier so that it can be referenced by a service request. For example, Item 23 in the March 2005 catalog. These entities are usually not dependent on other entities (such as through foreign key relationships). Typically updates to this data are infrequent, and tightly-controlled to prevent synchronization problems arising from replication.
- **Resource Data (SKUs, inventory)** – These entities have a very long lifetime. The format and update of these entities is private to a service that owns it. Updates to this data are very frequent, such as updates to the stock on hand field for an item. Typically, this kind of data has dependencies (such as through foreign key relationships) to reference data.
- **Activity Data (orders, itineraries**) – These entities have a lifecycle that is bounded by business activities, such as order life cycle from creation to fulfillment to payment receipt. However, these lifecycles may be extended by reporting needs - although expiration policies need to be put into place. The format of these entities should be private to the service that owns it. Typically this kind of data has dependencies (such as through foreign key relationships) to reference data.
- **Service Interaction Data (PO request form)** – These are the messages that travel between services. It is important to ensure guaranteed delivery of these messages. XML is a good representation for such data, and this data is implicitly immutable.

Pattern: Use entity aggregation to create an authoritative service for managing the state of a shared data collection

In most enterprises, data is spread across multiple data stores and across geographic and organizational boundaries. It is common to see data integration approaches (both real time, as well as batch) being used to move data between these disparate stores in an effort to maintain data consistency (Figure 9). However, often data integration is applied in an inconsistent fashion, leading to duplication, and often it may not be clear which system contains the single version of the truth for a particular entity. This problem is particularly common for reference (master) data. For example it is not uncommon to have, duplicates of product, vendor, or customer data. Therefore, what the enterprise needs is less data integration, and more data synchronization. The goal should be to synchronize data among multiple enterprise systems to ensure a single system of record for any data entity.

There is a need to create an authoritative service to manage the state of a shared collection of data and distribute a recent version to requesting (consuming) services. This service would provide a holistic view of an entity and its relationships with other entities, enforce business rules on access to that entity, and interact with the system of record for that entity.

Ideally entity aggregation services could hide complexity in a processing layer that applied straight through processing (STP) to queries. However, STP is not adequate to handle complex cases.

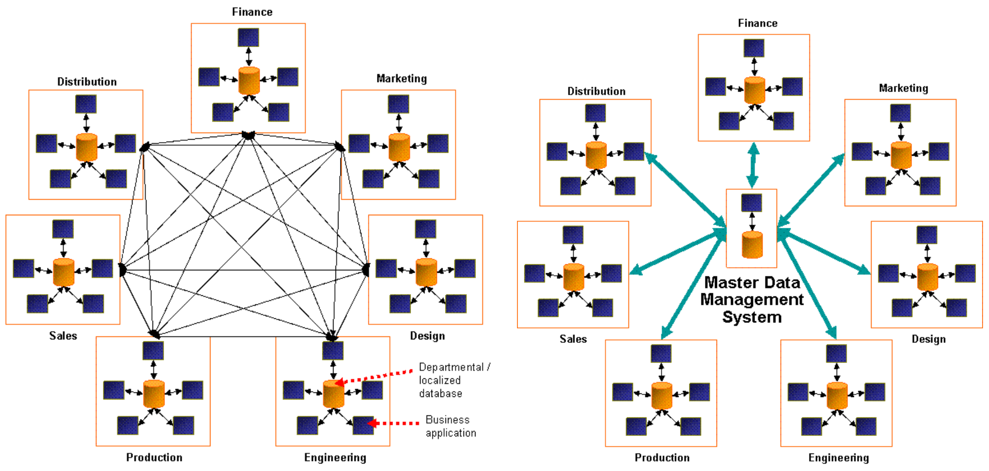


Figure 9: Data architecture before and after implementing entity aggregation services

Based on complexity, an entity aggregation service could be implemented using STP, or with partial or even full data replication. In each of these cases, there are design issues to be considered around schema reconciliation, ownership determination, instance reconciliation, CRUD semantics, and lifecycle issues. In the case where attributes of an entity are being maintained in multiple systems, then there are multiple systems of record to update when change occurs. Also, the schema for an entity should include enterprise-wide standard fields, but might also require extensions used by special applications.

Pattern: Align enterprise data models for effective performance management

You cannot measure business performance, gain business insight, convert data into information, and information into actionable plans without consistent, aligned enterprise data. The goal should be instrumentation of the enterprise at the operational, tactical, and strategic levels to enable effective performance management, as shown in Figure 10.

For example, a manufacturing corporation might have multiple facilities, and might be organized into multiple divisions or business units. Executives will want to measure the performance of the enterprise. They will want visibility into performance at the corporate, division, and factory levels. They need to make sure that service levels are high, costs are low, and that the business is growing. To do this, they need IT systems that are integrated all the way from the plant to the enterprise. However, more than just connectivity, it is necessary to have data models and data systems at each level that synchronize vertically, so that meaningful management decisions can flow down and meaningful business insight flow up. To further complicate matters, these data models need to be aligned across multiple functional areas. These include might include sales, marketing, distribution, procurement, inventory, financials, operations, and product engineering.

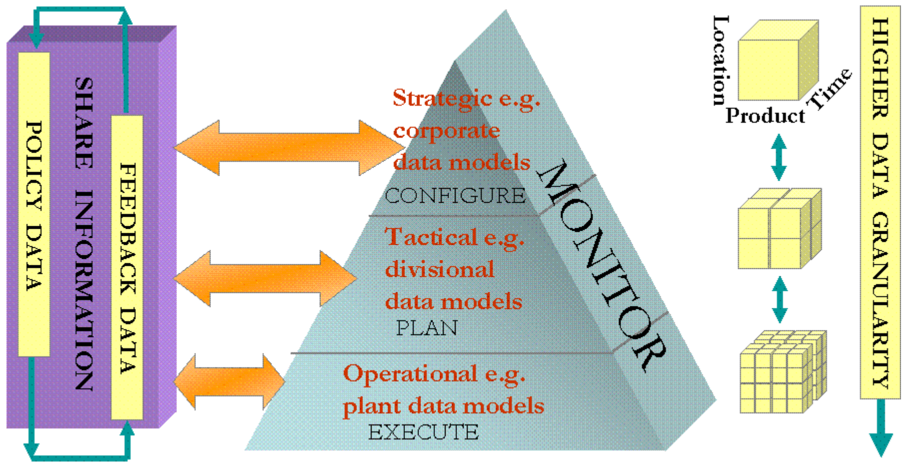


Figure 10: Data needs to be aligned between strategic, tactical and operational processes.

Unfortunately, in many enterprises data models are not aligned all the way up through the organization. This is the plant-to-enterprise (P2E) problem. This problem has also been described as data not being aligned from "top floor to shop floor."

Pattern: Manage the life cycle of individual entities

The goal should be to manage the life cycle of a single entity:

- A product data record – All the way from new product introduction through to obsolescence.
- A customer order – All the way from order creation, through delivery, to financial settlement.
- An enterprise event, or exception – All the way from the event being raised, to assignment to a person, to root-cause analysis, and finally to resolution.

For example, some initial steps in a new product introduction might be as follows – although a complete product lifecycle is a lot more complex.

1. Engineering design A manufacturer might design a new item, collaborating with suppliers to do so.
2. Preliminary planning:
    a. Sales forecasts for the new product based on projected demand.
    b. Product configuration decisions.
    c. Planning for how the new product must flow through the supply chain.
3. Setting up the item for sale at retail outlets:

a. Getting the item entered into the retailer's system – Not just physical attributes of the product, but also profiling and program setup based on the profile.

b. Preliminary forecast and replenishment collaboration between manufacturer and retailer in the context of the program.

Now as an individual business entity goes through its lifecycle, different business processes are impacted, different events are generated, and different roles are involved. Typically this might lead to many different kinds of cross-functional, or collaborative, business processes that require assembly of composite applications – which require data connections to backend.

Step 5: Connect Business Processes to Edge Systems

Often the scope of an OBA is not contained within an organization. For example, an OBA might support a business process that needs to consume a service that is offered by a hosting provider (Software as a Service – SaaS – scenario). Alternatively, the OBA might need to support a business process that offers a service to another organization. This is especially common in supply chain management scenarios where trading partners are involved. Here, there needs to be a way to transfer documents from one information-worker to a counterpart in the trading-partner organization. This needs to be secure, reliable, asynchronous, and transparent.

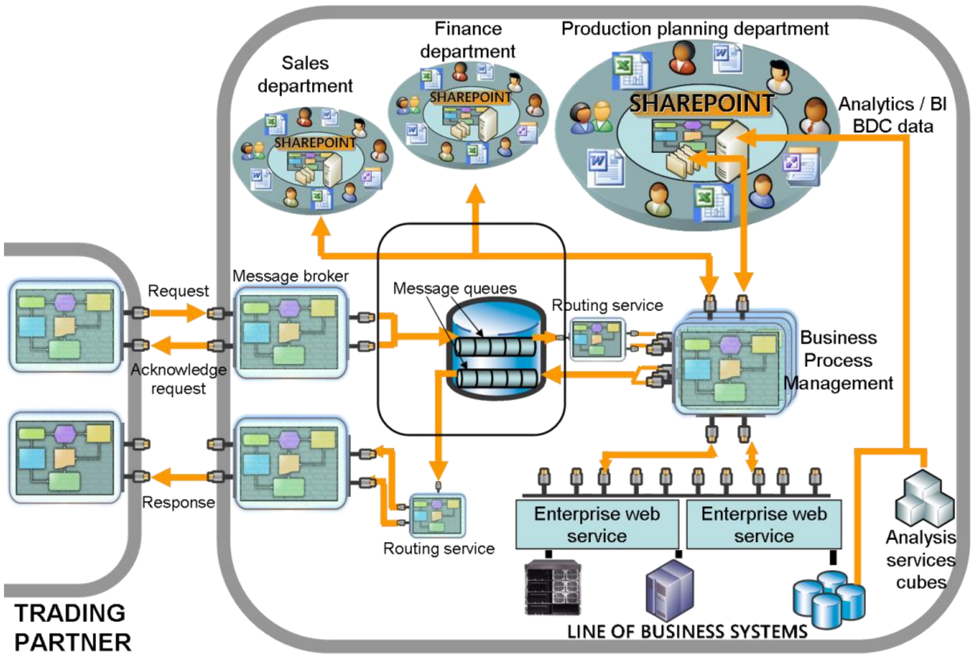


Figure 11: Connecting the OBA to systems at "the edge"

One way of putting together an end-to-end architecture for this is shown in Figure 11. Message brokers have been set up at the edge of the organization to send and receive messages and documents from trading partners. Messages from different trading partners can potentially be received in multiple message formats and delivered over multiple channels, such as Web services, EDI, e-mail, RosettaNet, and so on. Furthermore, messages can be exchanged in a variety of different patterns: one-way, asynchronous two-way, or synchronous two-way messaging. These message brokers must handle each combination of these message-interchange patterns and message formats.

After the message is received by the message broker, it is processed into the single canonical format that is required for downstream services, and the transformed message is persisted to a message queue to decouple public processes from private ones. Next, the message is retrieved from the queue by a routing service that examines the message and routes it to the intended recipient.

But before the document reaches its intended recipient, it might have to be preprocessed by enterprise application services such as LOB applications, or BPM orchestrations. Business rules might be applied to messages, to ensure validity and to enforce corporate policies. The result of all this processing is a document enough information that it can be processed by a human who can make a quick decision. For example, a purchase-order request from a customer might be fed into an order-promising service, and the response from this service might be used to generate an XML document that corresponds to an InfoPath form with a candidate PO Confirmation. Next, this generated form might be placed into a SharePoint forms library for an information worker in the sales department to approve.

After the information worker has reviewed the proposed confirmation and made any necessary changes, the worker submits the form. This kicks off the workflow for the return trip, which updates the LOB systems and then posts the information from the form as an XML document into a queue for outbound messages as a response to the original request. The message broker then converts back into the format used by the trading partner.

There are multiple ways to implement the message brokers at the edge, such as BizTalk Server. This would provide a scalable and manageable solution that would also come with standards-based accelerators and adapters, such as the RosettaNet accelerator for trading partner collaboration. The queues that decouple internal and external processes could be implemented using Microsoft SQL Service Broker 2005.

Pattern: Employ federated identity technologies to share identities and entitlements across organizational boundaries

IT organizations face two conflicting needs:

- Enable collaboration across organizational boundaries and networks. For example, collaboration to support trends like globalization, outsourcing, and new business models such as virtual enterprises.
- The need to protect organizational assets by providing ever-tighter network security.

Trying to accommodate both needs has led to a proliferation of passwords. This leads to two problems: lost productivity, and a greater security risk through a larger attack surface. For example, some industry estimates show enterprise customers averaging 12 external user IDs and passwords to manage, requiring between 15 to 20 minutes per day. This not only saps user productivity, but also opens security risks as users unable to memorize so many passwords jot them down on paper that can be lost or seen by others.

IT architectures need to enable federation of identity, which means delegation of identity related functions like authentication, authorization, or profile management to partners in line with previously established relationships of trust. This requires identity management technologies to securely share digital identity and entitlement rights (or "claims") across security and enterprise boundaries, such as Active Directory Federation Services.

Pattern: When federating identity, think of identity as a set of claims, rather than just a combination of username and password, along with a set of accompanying attributes.

The traditional model of identity has been the following:

- A digital identity is a combination of identifier, credentials, and attributes. Attributes may include a set of core attributes and a collection of context-specific attributes.
- Directory services are stores for digital identities. Along with digital identities, a directory service will also store the entitlements (irights and privileges) associated with each digital identity, in line with the security policies of the organization.
- An identity provider (IP) creates (or issues) digital identities by making entries into a directory service. A relying party (RP) consumes digital identities by authenticating users against the entries in the directory service.

The traditional model works when identity providers and relying parties are members of a single organization (or domain), as it is easier for identity providers and relying parties to have a shared understanding of identity models, processes, and technologies. Federated identity requires that an organization be able to consume identities issued by other organizations. This means that an administrator in an organization can control resources that users in that organization can access — both within the organization and at partner organizations. It also enables an administrator to configure resources that users in other organizations can access. Thus the idea of a unique identity model for each user becomes less meaningful when federating identities across organizations.

In a federated identity scenario, the new model of identity is as follows:

- A digital identity is a set of claims that one party makes about a subject.
- A claim is an assertion of the truth of something.
- A subject is the person or object that is being described across organizational boundaries.
- Claims are typically packaged into a security token, that can travel across process and machine boundaries.
- A digital identity is issued by an identity provider (IP), and consumed by a relying party (RP). There may be multiple digital identities issued to a single subject, issued by multiple IPs, each making a different set of claims.

Therefore, a claim is an expression of a right to access a protected resource or operation – much like a key. Access to a service or a resource is determined by comparing the claims that a user has to the set of claims that the user requires. A claim is implemented as a structure that contains the name of a claim type, the type of right that is being claimed, and finally the name of a resource. For instance, the statement "an entity can read c:\temp.txt file" might be modeled as:

ClaimType=File,
Right=Read
Resource=C:\temp.txt

In addition, a claim might also describe a property that an entity possesses. For instance, the statement "an entity's name is John Smith" might be modeled as:

ClaimType=Name
Right=PosessProperty
Resource=John Smith.

The implications are that in federated scenarios, a digital identity should become analogous to a drivers license:.

- Identity Provider: Directorate of Motor Vehicles.
- Claim: Ability to drive.
- Relying Party: Rental car agency that accepts the claim made by the DMV that the bearer can drive a car.

Suggested Reading

Following articles provide more information on topics in this chapter.

References on Service Orientation and Messaging

1. Service Orientation and Its Role in Your Connected Systems Strategy (Mike Burner): http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/SrOrientWP.asp
2. Service-Orientated Architecture: Considerations for Agile Systems (Lawrence Wilkes and Richard Veryard): http://www.architecturejournal.net/2004/issue2/aj2service.aspx
3. Principles of Service Design: Service Patterns and Anti-Patterns (John Evdemon): http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/SOADesign.asp
4. Razorbills: What and How of Service Consumption (Maarten Mullender): http://www.architecturejournal.net/2005/issue4/aj4razor.aspx
5. Principles of Service Design : Service Versioning (John Evdemon): http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/SOADesignVer.asp
6. Dealing with the Melted Cheese Effect: Contracts (Maarten Mullender): http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/MeltedCheese.asp
7. An Introduction to the Web Services Architecture and Its Specifications (Luis Felipe Cabrera, Christopher Kurt, Don Box): http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnwebsrv/html/introwsa.asp

8. Messaging Patterns in Service Oriented Architecture, Part 1 and Part 2 (Soumen Chatterjee):
Part 1 : http://www.architecturejournal.net/2004/issue2/aj2mpsoarch.aspx
Part 2 : http://www.architecturejournal.net/2004/issue3/aj3messaging.aspx

9. Metropolis: Envisioning the Service Oriented Enterprise (Pat Helland):
http://msdn.microsoft.com/seminar/shared/asp/view.asp?url=/architecture/media/en/metrov2_part1/manifest.xml

References on workflow and process

10. Build Applications on A Workflow Platform (David Green):
http://www.architecturejournal.net/2006/issue7/F1_Workflow/

11. Developer Introduction to Workflows for Windows SharePoint Services 3.0 and SharePoint Server 2007 (Andrew May):
http://msdn2.microsoft.com/en-us/library/aa830816.aspx

12. Of People, Processes, and Programs (Barry Briggs):
http://www.edithere.com/barry/stories/storyreader$1102:

13. Simplify Development with the Declarative Model of Windows Workflow Foundation (Don Box, Dharma Shukla):
http://msdn.microsoft.com/msdnmag/issues/06/01/WindowsWorkflowFoundation/default.aspx

14. Understanding BPM Servers (David Chappell):
http://www.davidchappell.com/articles/Understanding_BPM_Servers.pdf

References on integrated user experience

15. Choosing the right presentation layer architecture (David Hill):
http://www.architecturejournal.net/2005/issue4/aj4choosing.aspx

16. Metadata-Driven User Interfaces (John deVadoss): http://msdn.microsoft.com/library/en-us/dnbda/html/MetaDriveUI.asp

References on federated data

17. Data on the outside vs Data on the inside (Pat Helland):
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnbda/html/dataoutsideinside.asp

18. SOA Challenges : Entity Aggregation (Ramkumar K.):
http://msdn.microsoft.com/architecture/default.aspx?pull=/library/en-us/dnbda/html/dngrfsoachallenges-entityaggregation.asp

References on identity and access

19. Identity and Access Management (Fred Chong):
http://www.architecturejournal.net/2004/issue3/aj3identity_pf.aspx

20. Microsoft Identity and Access Management Series:
http://www.microsoft.com/technet/security/topics/identitymanagement/idmanage/default.mspx

21. Microsoft's Vision for an Identity Metasystem (Kim Cameron):
http://msdn2.microsoft.com/en-us/library/ms996422.aspx

22. Web Service Security Patterns: http://msdn.microsoft.com/practices/default.aspx?pull=/library/en-us/dnpag2/html/wssp.asp

23. The Laws of Identity (Kim Cameron): http://msdn2.microsoft.com/en-us/library/ms996456.aspx

Chapter 4

Sample Office Business Applications in the Enterprise

Introduction

Chapter 1 introduced composite applications Chapter 2 introduced platform technologies and capabilities to deliver composite applications called Office Business Applications (OBAs), and Chapter 3 provided a set of generic best practices to deploy OBAs in the enterprise. This chapter provides more specific guidance for building enterprise applications – with focus on enterprises that manufacture goods and get them to retail outlets through distribution networks.

Connected Systems for Manufacturing

Manufacturers today face the pressures of competition, government regulations, evolving technology, and high customer expectations. Increasingly though, new pressures and changes are intensifying and affecting how companies conduct business. With globalization, competitors are more numerous, more specialized, and respond faster to customers than ever before. This leads to shorter product lifecycles, rapid commoditization, and shrinking margins. In addition, the need for companies to play on the global scale often results in mismatches between the supply and demand for products and services. Globally networked supply chains also lead to longer order-through-to-settlement cycles, resulting in poor cash flows.

An increased focus on regulation is another destabilizing factor, as it requires companies have fine-grained control and visibility over organizational practices. These regulations could be governmental (Sarbanes-Oxley), exercised by trading partners (retailers requiring RFID tagging), or could be industry-specific (environmental regulations). Because technological advances continue to reshape entire industries, companies must remain at the forefront or face obsolescence.

Businesses tell us that these rapid changes and pressures present them with both their biggest challenge and their biggest opportunity. They know they must find a new, better way to navigate the business world. They need to have a clear view of their own resources, exchange information quickly and easily, and combining technologies in new ways.

These new imperatives place unprecedented pressure on business decision makers at manufacturing enterprises. First, businesses are expected to have greater focus on what they are doing. They need to focus on how they are going to grow. They need to focus on how to deliver more value and differentiate themselves from their customers. They need to have greater agility and greater responsiveness inside the organization. They need to have more visibility.

On one hand, companies are focusing more on the core of their business. They are focusing on what is happening inside the organization. At the same time, companies have greater and greater interdependence with the outside world. Every company exists within a value chain.

How do you establish trust with those other participating organizations? How do you establish visibility not just within your organization but across that entire value chain? From a customer perspective, if you have multiple participants in the value chain, how do you offer a seamless experience to the customer – even if there are multiple participants involved in pulling that thing together?

In short, remaining competitive in this environment requires companies to have a greater inward focus on their core business and processes, while simultaneously being more externally-focused on competitive differentiation, partner integration, and delivery of an agile value chain. AMR Research calls this the need for manufacturers to move to a Demand Driven Supply Network (DDSN), which they define as:

"A system of technologies and processes that senses and reacts to real-time demand signals across a supply network of customers, suppliers, and employees."

To do this, manufacturing companies need to synchronize demand (sales and marketing), supply (production), and product development (engineering). This requires linkages between the design, buy, make, and the sell side processes. Becoming demand-driven is a fundamental shift in how to do business. It is based on aligning processes – customer processes on one side, supplier partner processes on the other side, and production processes on the bottom. This allows companies to sense and shape demand and to respond profitably to demand fluctuations or disruptions in the supply chain. It is this shift that is creating the demand-driven supply network.

This need to synchronize has been recognized for years. For example, in his book *Men and Machines,* published in May 1929, Stuart Chase writes:

"Before the advent of mechanical power, handicraft met demand as it arose...Then came James Watt and his steam engine. The machine proceeded to develop in accord with its own laws regardless of the needs and conveniences of man. It continually increased its efficiency of rapid production with corresponding decrease of efficiency in elasticity and adaptability…still production continues to increase, not in response to any demand but under the compulsions of mechanical evolution…"

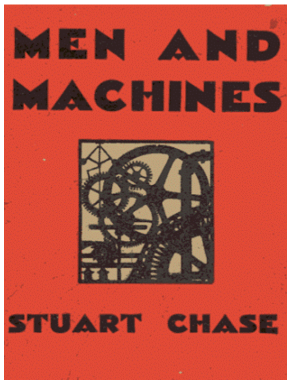


Figure 1: The need to synchronize was recognized by Stuart Chase in this book published in1929.

But how does a manufacturer become demand-driven? As Prof. Hau Lee points out in an article called "The Triple-A Supply Chain" in the October 2004 edition of the Harvard Business Review, "The best supply chains aren't just fast and cost-effective. They are also agile and adaptable, and they ensure that all their companies' interests stay aligned."

These three traits – agility, adaptability, and alignment (the Triple-A) – require manufacturers to put in business processes that help to manage change. Examples of such business processes are demand shaping and management, pull-based replenishment, and integrated product design.

These kinds of business processes require IT systems that detect change as early as possible and then respond to change at the right time and in the right manner. This requires IT assets that can be rapidly connected and configured, with the connections between assets also reconfigurable. This requires composition. It should be possible to not only connect systems internally, within the enterprise, but also externally, beyond organizational boundaries. Being connected internally implies connectivity from the edge of the network all the way to the enterprise systems – from plant instrumentation to enterprise LOB systems. Being connected externally requires connectivity with partners, suppliers, and customers.

What are the benefits of connecting systems and aligning processes in this manner? Manufacturing organizations will see improvements in all the metrics that are important to them, and this can bring significant financial benefits. For example, AMR Research has shown that achieving higher levels in the DDSN maturity model will lead to improvements in demand forecast accuracy and perfect order percentages. An AMR study with roughly 300 manufacturing companies showed that, on average, a 5% forecast accuracy improvement led to a 10% perfect order improvement. That translates to 50 cents of earnings per share (EPS) improvement, a 5% return on asset improvement, and 3.3 % margin improvement. This is huge for manufacturers.

There is, however, a major obstacle for manufacturers to implement the software and systems that can drive these kinds of cross-functional business transformations. Organizations will already have existing systems in place to handle their current business processes, and, while these systems may automate certain kinds of transactions and processes, in most cases they will not work well outside the boundaries of what they were designed to do. So it will not always be clear how to migrate systems from their “as-is” states to the desirable “to-be” state. In many cases, it may not even be clear what the desired to-be state should be.

Like any other journey, this transition requires a roadmap. AMR research suggests one such roadmap, shown in Figure 2. While this roadmap was devised to help organizations re-engineer their supply chains to be demand-driven, it is laid out in terms of generic capabilities that should enable agility, adaptability, and alignment for many kinds of business processes.

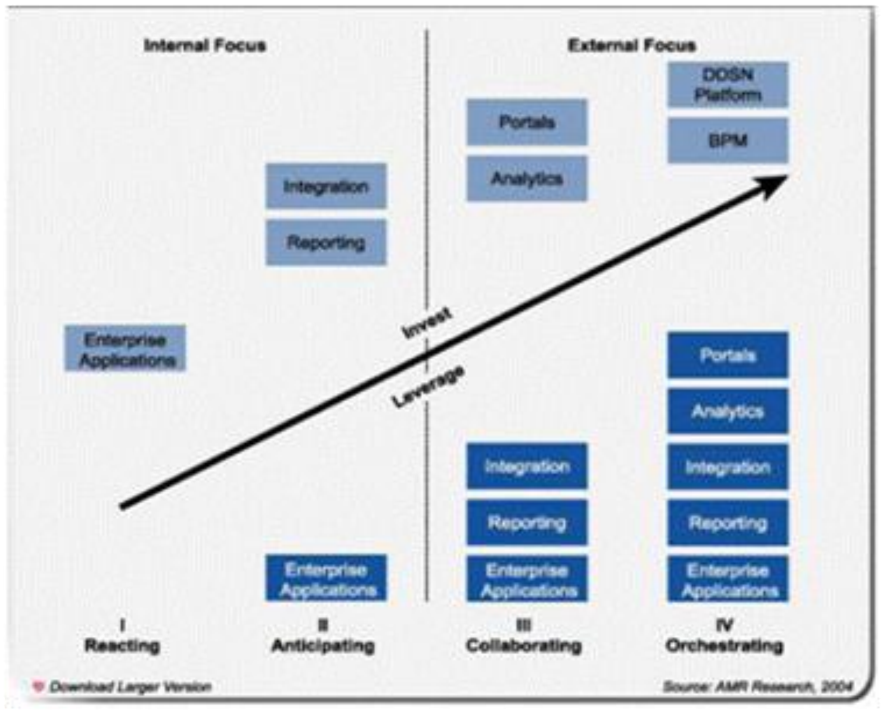


Figure 2: High-level roadmap proposed by AMR Research to align supply, demand, and product innovation-related processes into a demand-driven supply network

At a very high level, the benefits of such an approach are laid out in the table below, also from AMR Research.

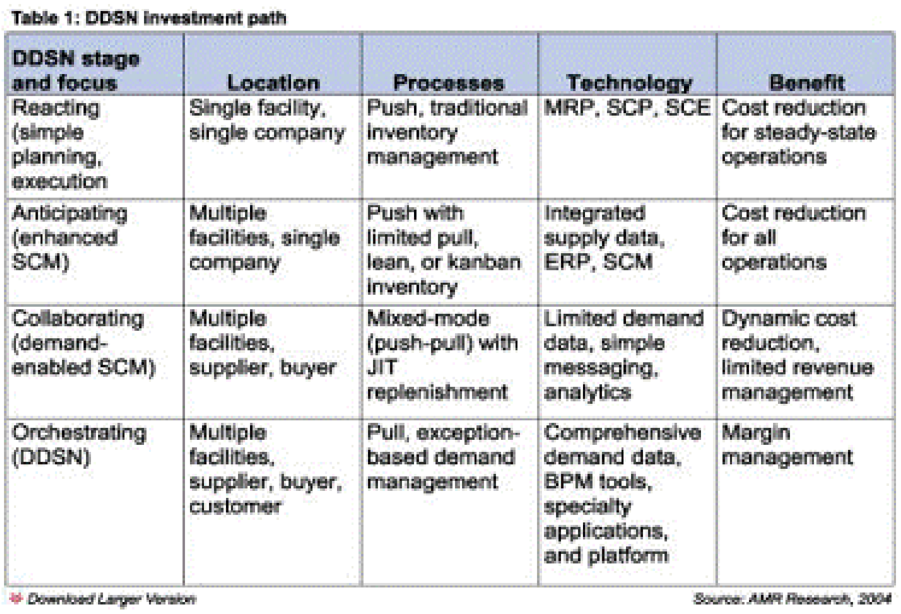


Table 1: DDSN investment path

DDSN stage and focus	Location	Processes	Technology	Benefit
Reacting (simple planning, execution	Single facility, single company	Push, traditional inventory management	MRP, SCP, SCE	Cost reduction for steady-state operations
Anticipating (enhanced SCM)	Multiple facilities, single company	Push with limited pull, lean, or kanban inventory	Integrated supply data, ERP, SCM	Cost reduction for all operations
Collaborating (demand-enabled SCM)	Multiple facilities, supplier, buyer	Mixed-mode (push-pull) with JIT replenishment	Limited demand data, simple messaging, analytics	Dynamic cost reduction, limited revenue management
Orchestrating (DDSN)	Multiple facilities, supplier, buyer, customer	Pull, exception-based demand management	Comprehensive demand data, BPM tools, specialty applications, and platform	Margin management

Download Larger Version	Source: AMR Research, 2004

Summarizing this guidance from AMR Research, organizations need to add capabilities for integration, reporting, portals, analytics, and business process management to sense and respond to real-time demand signals across a supply network of customers, suppliers, and employees. However this leaves IT departments grappling with the question of how to actually architect their internal systems to achieve these benefits. The remainder of this chapter will show how to leverage the platform capabilities from Chapter 2 and the architectural guidelines from Chapter 3 in a set of sample scenarios.

Connecting Business Processes Across Enterprises

The points made in this article will now be illustrated in the context of a solution template for supply chain collaboration. This solution was chosen because close collaboration with customers, suppliers, and partners has become essential for companies to be successful in the marketplace. The importance of this was highlighted in a comment made by Robert Handfield, Optimize, 2002: "…the future won't be about companies competing against each other, but rather about supply chains competing against other supply chains." Typically, companies collaborate with their suppliers so that procurement transactions can be automated, and also so timely information can be shared without too much manual intervention.

A functional overview of supplier collaboration is shown in Figure 3. Note that in this figure, workflows have been segmented into strategic, tactical and operational.

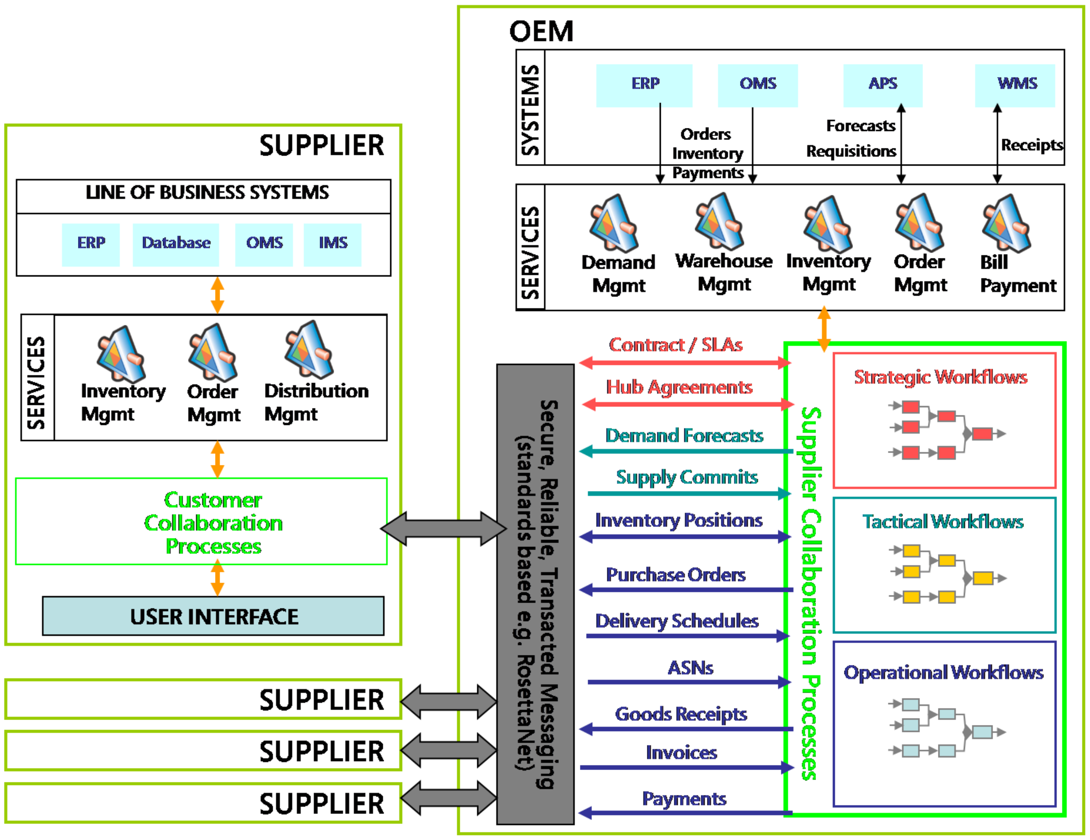


Figure 3: Functional overview of supplier collaboration

The goal is to make the supplier collaboration triple-A – thus reducing costs and inefficiencies, while ensuring a high level of service from suppliers. If this process is not set up to be triple-A , then the organization may experience the pain points listed in the table below.

Stakeholders	Collaboration constraints	Consequences
Customer Mfg	Poor understanding of supplier capacity to meet demand	Over-order of inventory Bullwhip effect
Logistics	Poor communication of capacities	Increased transportation and overtime labor costs
Sales	Poor communication of urgency	Increased quoted price and eroded margins
Manufacturing	Poor visibility of true needed date	Increased manufacturing cost due to expediting

To avoid these potential problems, we need to build composite solutions that enable cross-functional processes across the different stakeholders in the table.

Building a Composite Application for Trading Partner Collaboration

Figure 4 illustrates how a purchase order request is routed within a supplier organization. This is similar for all the other form-based operational processes in Figure 3. In the section on best practices for the workflow capability of the connected systems model in Chapter 2, we mentioned that workflow is everywhere, and that it does not just live in the data center on integration servers. This is illustrated in Figure 4, where workflow is shown serving the following functions:

- Processing inbound messages – For example, to transform from the multiple supplier formats into a single canonical format to be used in private processes.
- Setting up manual processes – For example, by pre-populating information on documents for review.
- Routing documents and managing their lifecycle for items – Such as task initiation, delegation, notification, completion, and expiration.
- Completing manual processes – Such as updating LOB applications.
- Processing outbound messages – For example, by converting back from a canonical document format into one of the multiple formats used by different suppliers.

These workflow instances could be running on central servers in the corporate data centers, or they could be running closer to information workers on departmental servers, or they could be running on client applications to manage document routing and lifecycles.

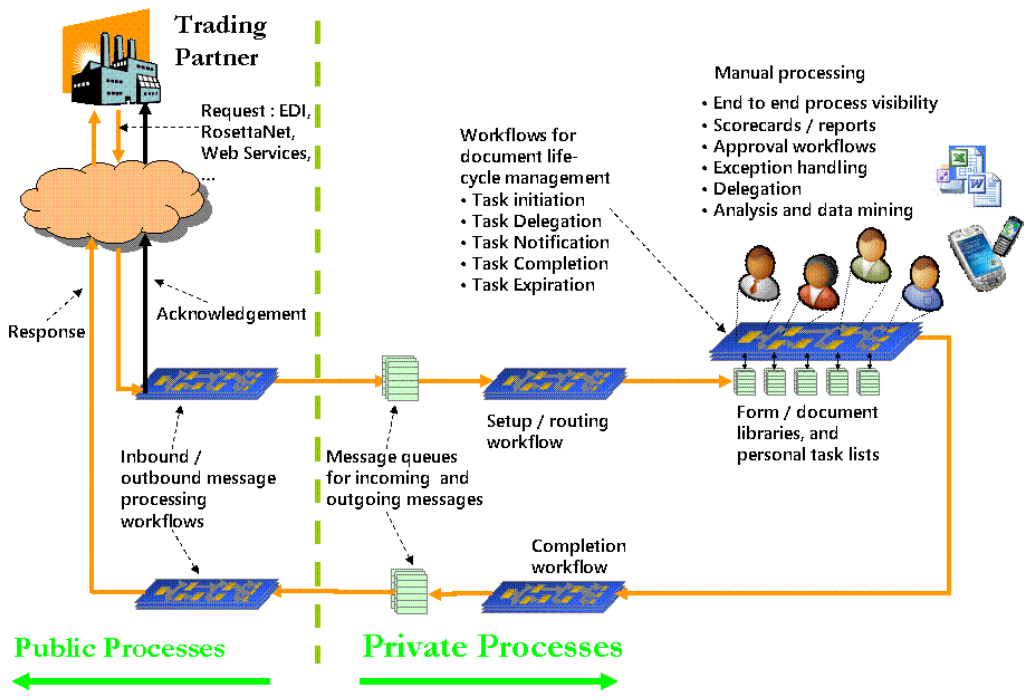


Figure 4: Building a composite application for trading partner collaboration

Logical View of the Architecture

Figure 5 shows a logical view of the architecture simplified from Figure 4. We will then break this down into smaller sub-systems for discussion.

Error! Reference source not found.: Logical view of the architecture

Handling Inbound Messages

An inbound message is a document received from a trading partner and needs to be acknowledged and routed to an appropriate person for processing, as shown in Figure 6. This set of activities requires the following capabilities, each of which are explained below.

- Message transformation.
- Message persistence.
- Asynchronous message processing.
- Master data management.
- Client access to application services exposed by LOB systems.

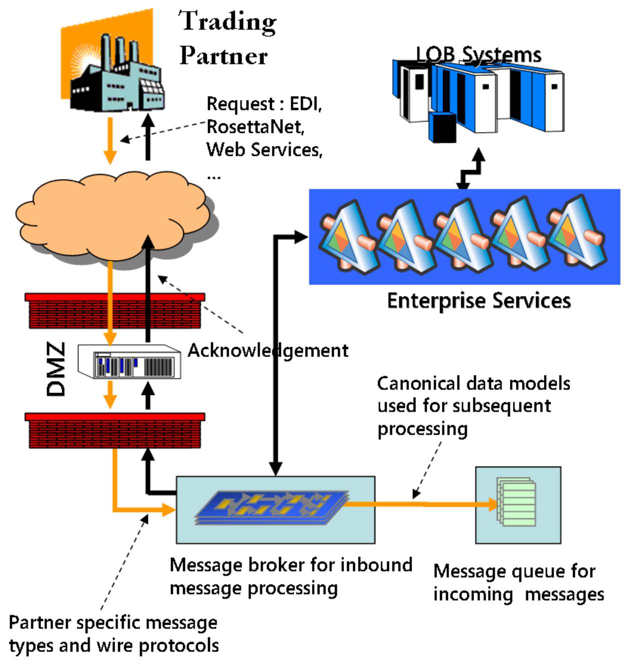

Trading Partner
LOB Systems
Request : EDI, RosettaNet, Web Services, ...
Enterprise Services
DMZ
Acknowledgement
Canonical data models used for subsequent processing
Message broker for inbound message processing
Message queue for incoming messages
Partner specific message types and wire protocols

Figure 6: Architecture for inbound message processing

Message transformation

Inbound messages from different partners may potentially be received in multiple message formats and delivered over multiple channels, such as Web services, EDI, e-mail, and RosettaNet. Messages may be exchanged in a variety of different patterns – one-way, asynchronous two-way, or synchronous two-way messaging. Therefore, the architecture needs to provide multiple messaging channels. These are entry points to handle each combination of these message interchange patterns and message formats. Once the message is received over one of these channels, it needs to be transformed into a canonical format that is required for downstream services. This does not need to be in the same format as the original request.

Message persistence

Once the message is received by the message broker, it gets processed, and a transformed message is persisted to a message queue. This enables loose-coupling between the message sender and the message receiver and promotes adaptability in the architecture. The message persisted in the message queue will be in the canonical format described above. The goal here is reliability – no message should be lost – and it is only after the message has been persisted that an acknowledgement should be sent back to the requesting party. Another benefit is that it is easier to build fault tolerance into the overall architecture in this fashion.

Asynchronous message processing

Once the incoming message has been transformed and persisted, an acknowledgment message can be returned to the sender. This acknowledgement is synchronous and is a response to the original request acknowledging receipt of the request. This is not likely to be an actual response – unless the message processing can be completely automated. The actual response will come later, after the message has been processed by an appropriate person or system – and this will be asynchronous to the original request requiring a capability to make a call back to the systems of the requesting party. This capability requires a shared understanding of the interfaces to make that call back. Both the sending of the acknowledgement and the final response will be governed by service-level agreements (SLAs) that have been established during the setup of a specific program between the trading partners.

Master Data Management

Asynchronous message processing implies that both trading partners will have the ability to refer to the original request (for example, by referring to an order by its ID) at later times. This requires a shared understanding of identifies for individual business entities. This can get complicated when the data schemas from the request are transformed into the canonical data models used within the organization. However as shown in Figure 7, the complexity does not end there – because different departments internally also refer to documents with different identifiers.

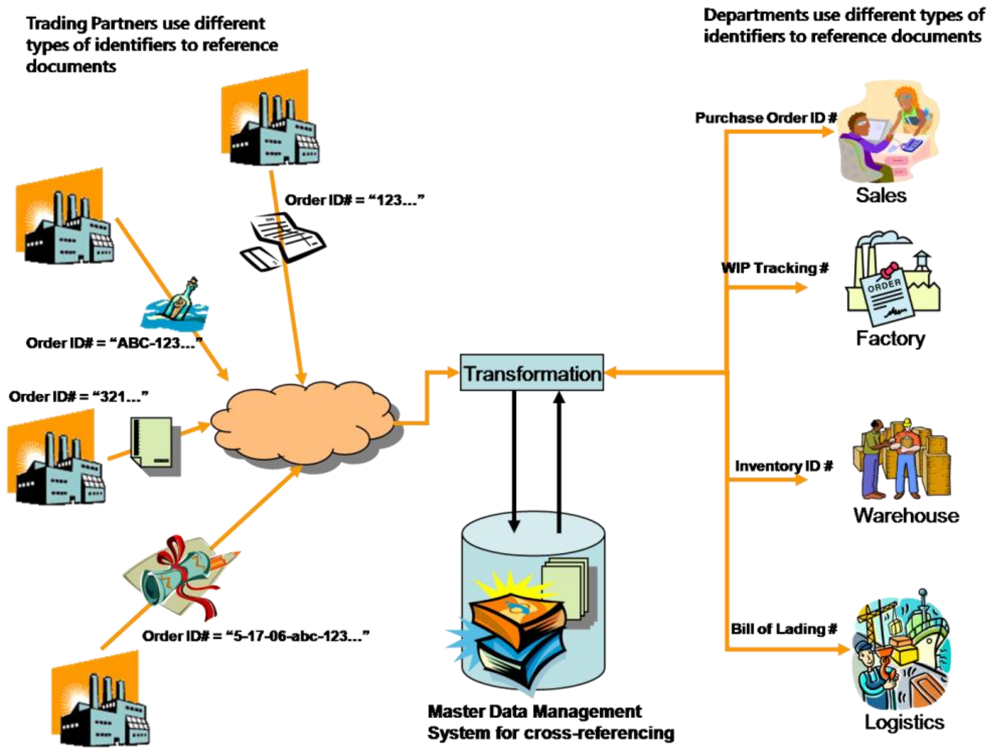

Trading Partners use different types of identifiers to reference documents
Departments use different types of identifiers to reference documents
Order ID# = "123..."
Order ID# = "ABC-123..."
Order ID# = "321..."
Order ID# = "5-17-06-abc-123..."
Transformation
Master Data Management System for cross-referencing
Purchase Order ID #
Sales
WIP Tracking #
ORDER
Factory
Inventory ID #
Warehouse
Bill of Lading #
Logistics

Figure 7: Master data management system needed to cross-reference documents and manage master data elements

Client access to application services exposed by line-of-business systems

As part of the inbound message transformation process, it is very likely that application services exposed by existing LOB applications will need to be accessed. Typically the inbound message will need to be extended with information from these applications to create the canonical data models used in subsequent processing. This extra information might be for internal document tracking, for aggregate level information that is missing in the request, or for routing purposes. For example, one of the application services likely to be invoked during the message transformation process is the master data management system described above.

There are multiple ways to instantiate the architecture for processing inbound messages. One way is to use the BizTalk Server as the message broker and to use accelerators to make it easier to synchronize with trading partners (such as the RosettaNet accelerator). BizTalk can also be used to transform messages and then push them into SQL Service Broker – which queues them up prior to subsequent processing.

Workflows to Setup Manual Activities

Documents need to be routed from their message queues to the appropriate person that will be handling that document. Before the document reaches its intended recipient, it may need to be pre-processed into a form that the end user can use for a quick decision. For example, a purchase order request from the customer may need to be run through an order promising or fulfillment engine to check the request against inventory that is available to promise (ATP). Figure 8 demonstrates these capabilities.

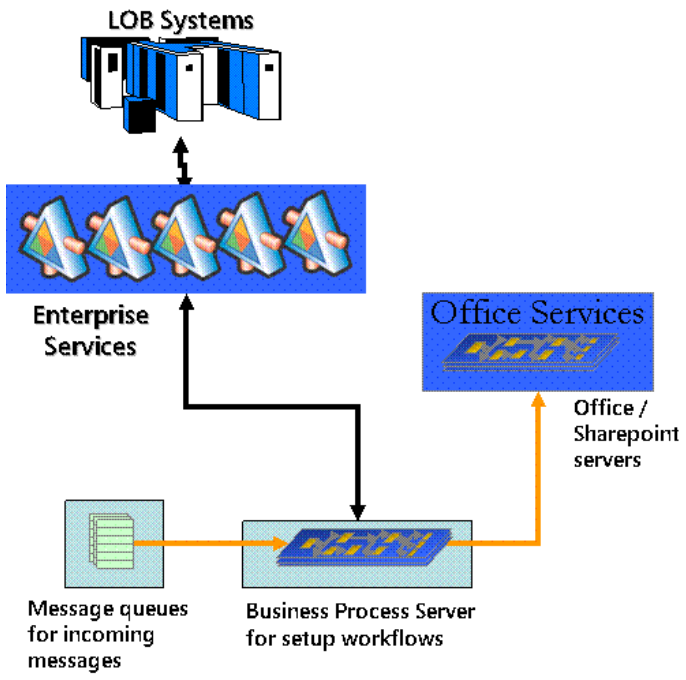

LOB Systems
Enterprise Services
Office Services
Office / Sharepoint servers
Message queues for incoming messages
Business Process Server for setup workflows

Figure 8: Workflows to set up manual processes

These workflows could be triggered as documents arrive or could be run in batch mode on a scheduled basis. Again, there are multiple ways to instantiate this architecture. For example, one way is to use the BizTalk Server. This is discussed in more detail in the section on implementation below.

Smart Documents – InfoPath for PO Request and Confirmation Forms

In the document routing step above, data and content for smart documents is generated by a business service that has pre-processed the incoming message. Typically this content is XML that complies with specified schemas for canonical document formats. Documents that reach the end users for manual processing can be of different types, but it is expected that these will be InfoPath forms for operational workflows and spreadsheets for tactical and strategic workflows.

It is straightforward for information workers to use InfoPath and Excel to create forms and spreadsheets that can generate XML conformant to a specified schema. Coding is not required, and XML schematization of these documents does not detract from flexibility to customize look-and-feel. Both Excel and InfoPath provide users with the ability to import XML schemas pertaining to business processes and to then use graphical tools to associate schema elements with graphical controls (InfoPath) or spreadsheet cells (Excel). For many business processes, there are industry standards available which provide XML schemas you can use to create the forms and spreadsheets. This is depicted in Figure 9.

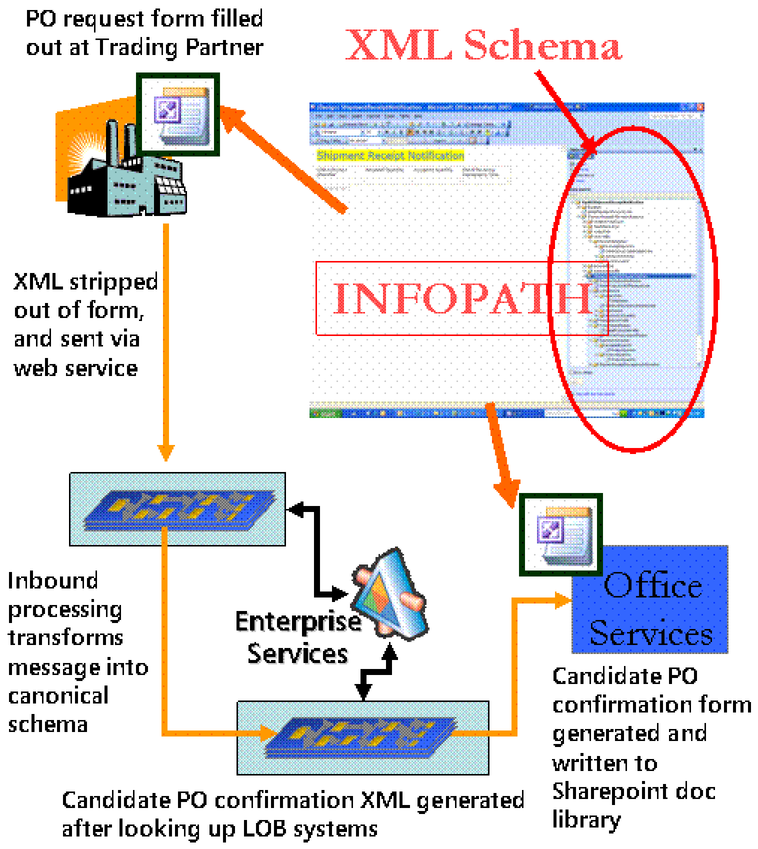

PO request form filled out at Trading Partner
XML Schema
INFOPATH
XML stripped out of form, and sent via web service
Inbound processing transforms message into canonical schema
Enterprise Services
Office Services
Candidate PO confirmation form generated and written to Sharepoint doc library
Candidate PO confirmation XML generated after looking up LOB systems

Figure 9: PO request and PO confirmation forms created from industry standards based XML schemas

Once the smart document is generated from within the setup workflow, it is submitted to the Office services responsible for managing its lifecycle. Typically this means that the document is submitted to a SharePoint document library or form library. Once this is done, an entry can be made to a user's task list and an e-mail sent to that user, including a notification of the task and a link to the Smart Document for download.

Office Services – Document Lifecycle Management

Smart documents are submitted to Office services for manual processing, for example to a document library or a forms library in SharePoint. The SharePoint Server can be configured to start workflows whenever a document is added to a library or is modified. In addition to out-of-the-box workflows for things like document approval, the server can be extended with custom workflows and activities. Users can then configure a document library to associate a set of workflows with a set of actions (such as document added, or document modified).

In addition to this support for workflow, Office 12 Server provides great out-of-the-box support for a number of Office services such as collaboration, enterprise content management, portal, search, workflow, business intelligence, and project management. In this way SharePoint forms a great platform for hosting composite applications as shown in Figure 10.

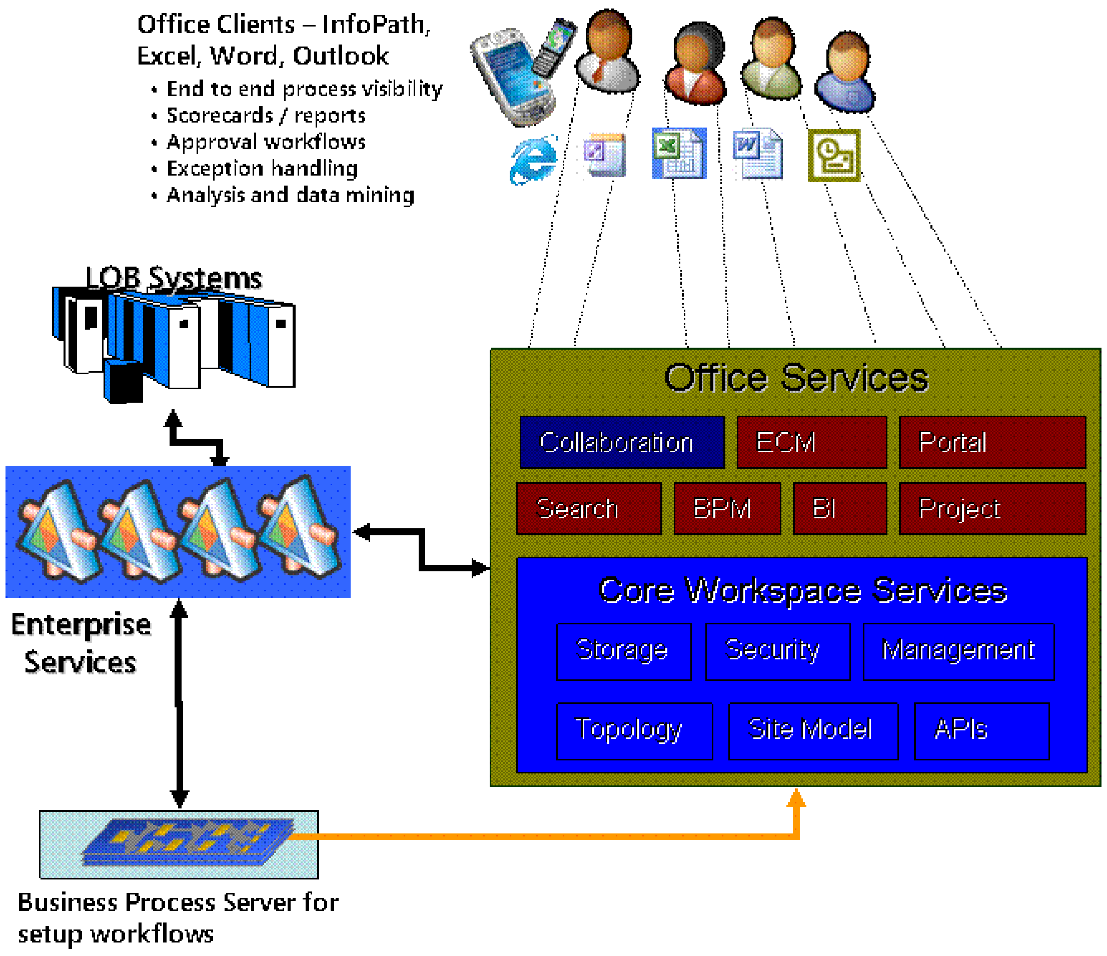


Figure 10: Office 12 Server provides platform support for a number of different types of Office services

For example, consider document lifecycle management (Figure 11). Documents that reach users for manual processing go from authoring and collaboration through management and publication to archiving or destruction. This refers to all kinds of documents--whether they are InfoPath forms or Excel spreadsheets. The stages that a document might go through are:

- **Authoring –** Process of creating or editing the document.
- **Collaboration –** Process by which multiple users engage upon a single document.
- **Management –** Process by which documents are stored, versioned, audited, and so forth.
- **Publication –** Process by which finished documents are uploaded back into the business process for further processing.
- **Archiving –** Process by which completed documents are stored for future reference.
- **Destruction –** Process by which old documents are deleted.

Whenever a document reaches one of these stages, a new workflow could be triggered. As an example, when a document expires, an expiration workflow would manage the archiving process.

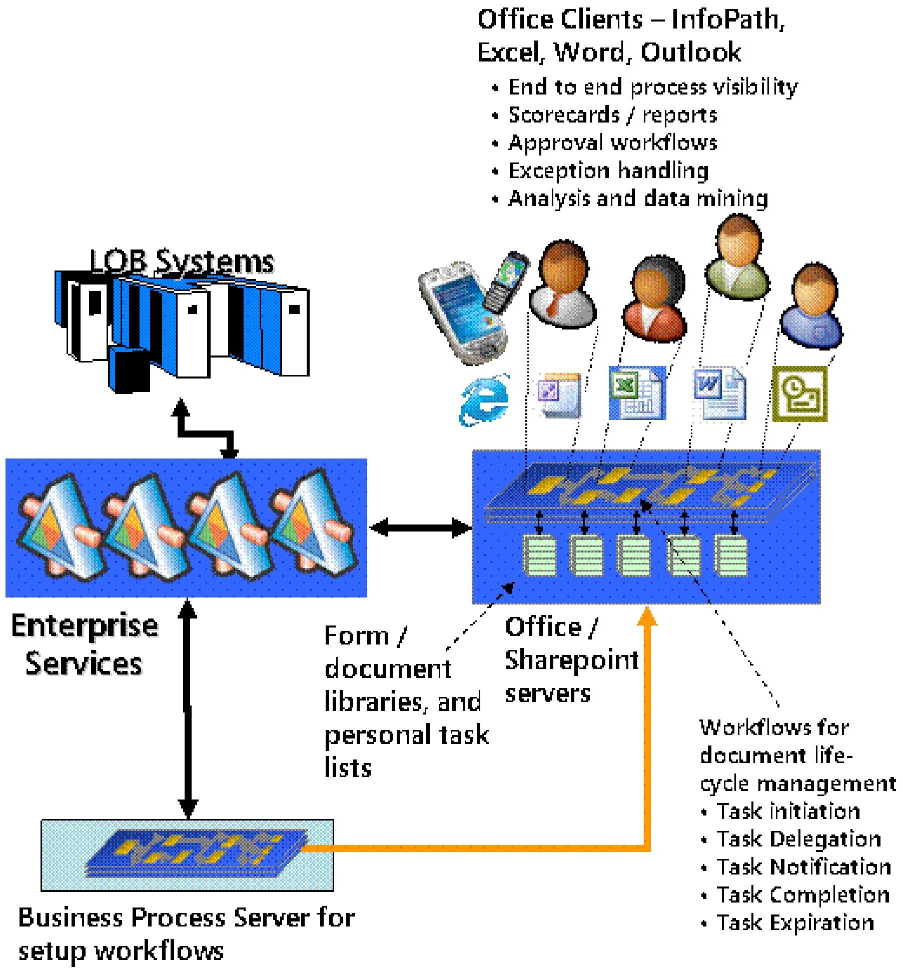

Office Clients – InfoPath, Excel, Word, Outlook
• End to end process visibility
• Scorecards / reports
• Approval workflows
• Exception handling
• Analysis and data mining
LOB Systems
Enterprise Services
Form / document libraries, and personal task lists
Office / Sharepoint servers
Workflows for document life-cycle management
• Task initiation
• Task Delegation
• Task Notification
• Task Completion
• Task Expiration
Business Process Server for setup workflows

Figure 11: Managing the life cycle of documents during manual processing steps

Workflows to Complete Manual Activities

Users can submit Smart Documents by editing the document and saving it back to the document library. This would then trigger a completion workflow to process the document. The activities in this completion workflow would then invoke endpoints on enterprise services to update the appropriate LOB systems. Once this is done, the completion workflow could post any reply back to the trading partner through message queues that have been set up for outgoing messages.

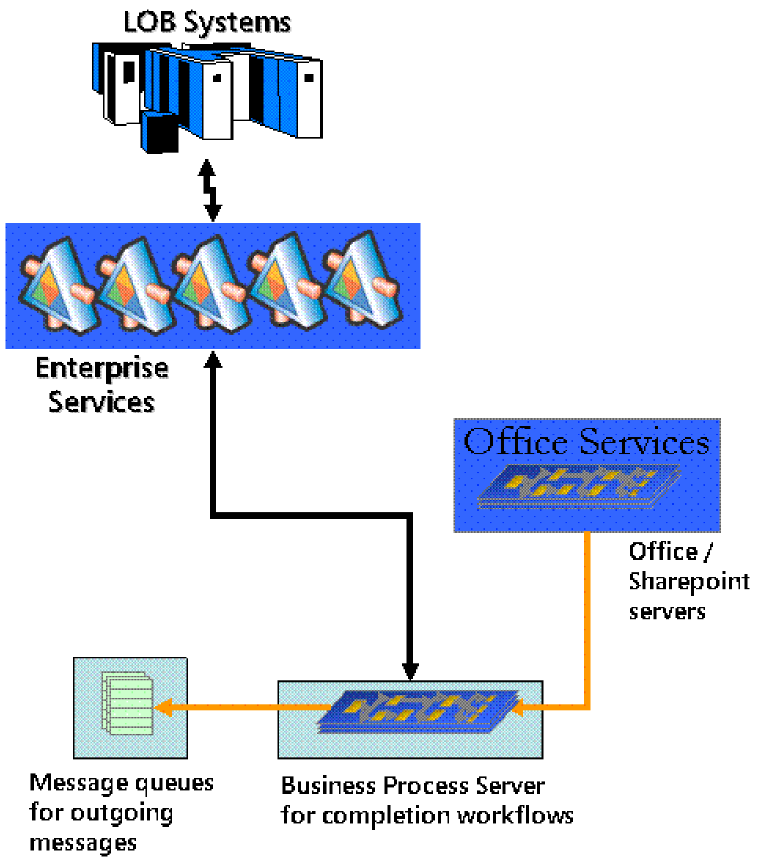

LOB Systems
Enterprise Services
Office Services
Office / Sharepoint servers
Message queues for outgoing messages
Business Process Server for completion workflows

Figure 12: Workflows to complete manual activities, for example by updating Line Of Business Systems

Note from Figure 12 that business process workflows can be located centrally, or closer to the user. For example, these workflows could be running within central servers running BizTalk for enterprise-wide process automation or within departmental servers running applications like Office Server that have embedded Workflow Foundation.

Handling Outbound Messages

Once messages have been posted to queues for outbound messages, they need to be delivered as responses to the original request (Figure 13). This task needs to be handled by a message broker, as outbound messages need to be converted from canonical formats into the formats used by each of the different trading partners. As discussed above, outbound messages to trading partners may potentially be sent in multiple message formats and delivered over multiple channels. Therefore, the architecture needs to provide for a set of exit points from the system that can handle each combination of message formats and distribution channels. This is the job that the message broker needs to perform.

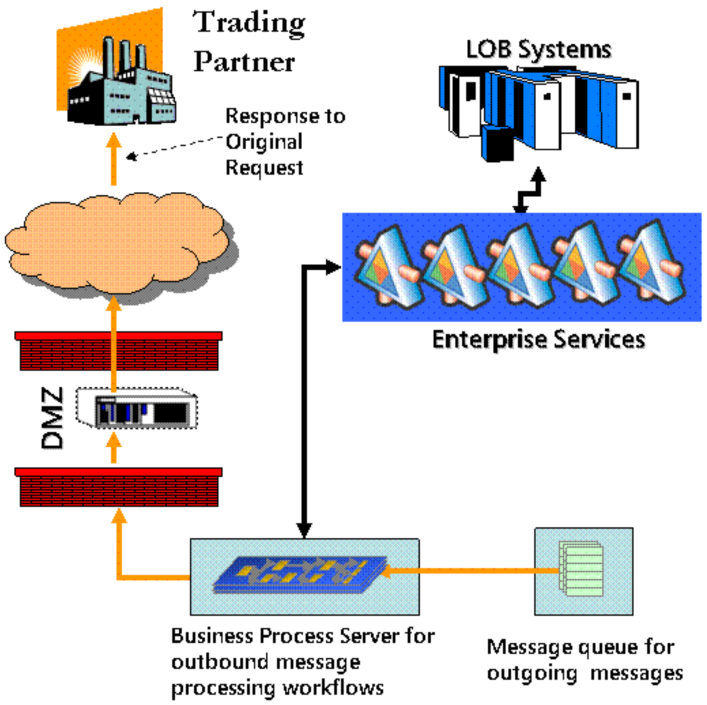

Trading Partner
Response to Original Request
LOB Systems
Enterprise Services
DMZ
Business Process Server for outbound message processing workflows
Message queue for outgoing messages

Figure 13: Workflows to process outbound messages

Note that clearly this is asynchronous from the initial request and that this processing of outbound messages could be achieved in multiple ways. For example, this processing could be happening in outbound pipeline processing in a BizTalk server or could be done in a server hosting Workflow Foundation.

Implementing this Solution as a Composite Application

To demonstrate the use of composition to build this solution, a reference implementation was built using two scenarios involving documents exchange between a retailer and a manufacturer:

- Purchase order (PO) request sent by the retailer, followed by PO confirmation returned by the manufacturer.
- Strategic forecasts sent by the retailer, followed by response to strategic forecasts returned by the manufacturer.

To demonstrate composition, the implementation leveraged data and process models from the RosettaNet industry standard for B2B collaboration. This allowed us to build a collection of software assets that could be deployed into the 2007 Microsoft Office System and other Microsoft platform technologies. The approach followed was to first generate XML schemas (XSDs) from the DTDs publicly available and then to use the schema definitions to design various artifacts--like Web service interface definitions, InfoPath forms, and Excel spreadsheets. This is shown in Figure 14, and the implementation approach is described below that, followed by a list of the software assets that made up the composite solution. For more information on the RosettaNet spec, visit their Web site at http://www.rosettanet.org.

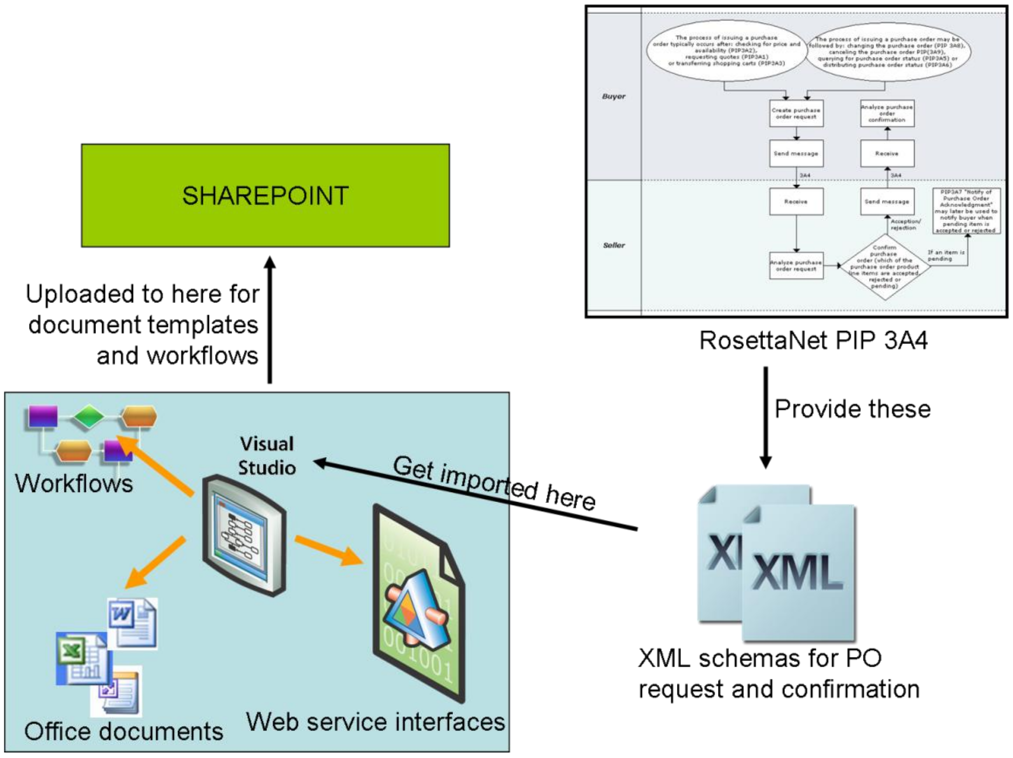


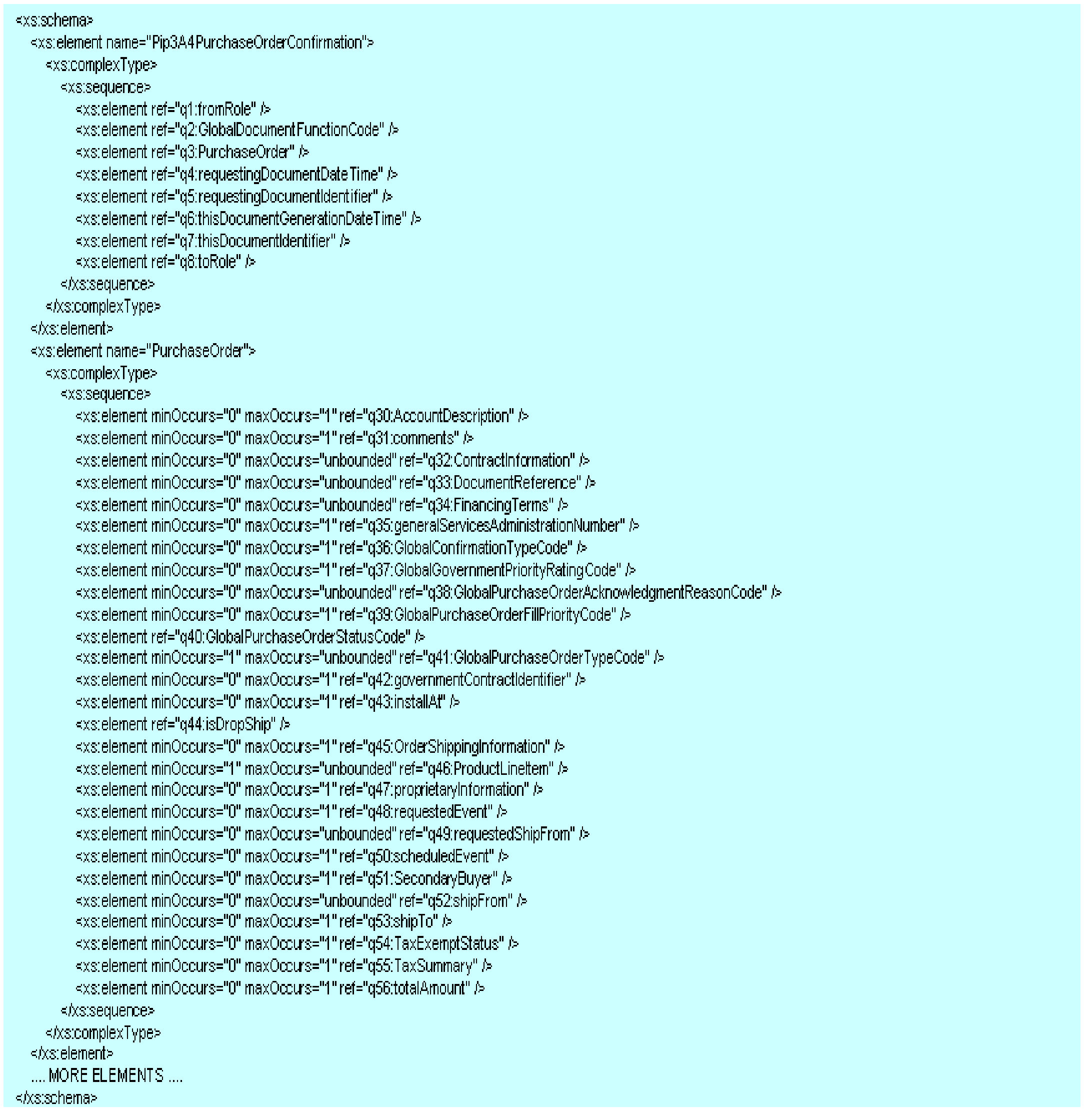
Figure 14: Data and process models from the RosettaNet standard were leveraged to build a collection of software assets, which could be assembled into composite applications

Generating the Industry Models

In the past, data issues have been the bane of many an enterprise solution, and one of the major technical hurdles is the choice of data models – as these form the foundation of the solution. It is hard to come up with good models without a lot of iteration, and usually the best approach is to leverage past industry learning and experience for the particular composite application being built. The best way to do this is to seek out an industry standard with pre-packaged data models. For the reference implementation, the RosettaNet specification was used, which comes with various process and data models for trading partner collaboration. As the data models are in the form of DTDs, the first step was to convert these into a form that is more amenable to tooling, and so the DTDs were converted into XML Schema (XSD) documents using Visual Studio. An example of the PO Confirmation schema is shown in Figure 15.

The next step was to use the xsd.exe tool to generate C# classes that corresponded to the data models:

```
xsd 3A4_MS_V02_03_PurchaseOrderConfirmation1.xsd /l:CS /classes
/outputdir:classes /namespace:RosettaNet_3A4_MS_V02_03_Confirmation
```



```
<xs:schema>
  <xs:element name="Pip3A4PurchaseOrderConfirmation">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="q1:fromRole" />
        <xs:element ref="q2:GlobalDocumentFunctionCode" />
        <xs:element ref="q3:PurchaseOrder" />
        <xs:element ref="q4:requestingDocumentDateTime" />
        <xs:element ref="q5:requestingDocumentIdentifier" />
        <xs:element ref="q6:thisDocumentGenerationDateTime" />
        <xs:element ref="q7:thisDocumentIdentifier" />
        <xs:element ref="q8:toRole" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="PurchaseOrder">
    <xs:complexType>
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" ref="q30:AccountDescription" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q31:comments" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="q32:ContractInformation" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="q33:DocumentReference" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="q34:FinancingTerms" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q35:generalServicesAdministrationNumber" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q36:GlobalConfirmationTypeCode" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q37:GlobalGovernmentPriorityRatingCode" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="q38:GlobalPurchaseOrderAcknowledgmentReasonCode" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q39:GlobalPurchaseOrderFillPriorityCode" />
        <xs:element ref="q40:GlobalPurchaseOrderStatusCode" />
        <xs:element minOccurs="1" maxOccurs="unbounded" ref="q41:GlobalPurchaseOrderTypeCode" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q42:governmentContractIdentifier" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q43:installAt" />
        <xs:element ref="q44:isDropShip" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q45:OrderShippingInformation" />
        <xs:element minOccurs="1" maxOccurs="unbounded" ref="q46:ProductLineItem" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q47:proprietaryInformation" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q48:requestedEvent" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="q49:requestedShipFrom" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q50:scheduledEvent" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q51:SecondaryBuyer" />
        <xs:element minOccurs="0" maxOccurs="unbounded" ref="q52:shipFrom" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q53:shipTo" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q54:TaxExemptStatus" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q55:TaxSummary" />
        <xs:element minOccurs="0" maxOccurs="1" ref="q56:totalAmount" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  .... MORE ELEMENTS ....
</xs:schema>
```

Figure 15: An abbreviated view of the XML schema document for the PO Confirmation model in the RosettaNet schema

Building Client Applications

User interfaces were designed using Office client applications. Purchase Order (PO) processing forms (request and confirmation) were built using the InfoPath graphical forms designer. Strategic forecasts were built into Excel. Visual Studio Tools for Office (VSTO) was used to extend Outlook with an add-in, for users to search for and edit POs in the event of e-mail notifications.

For example, Figure 16 shows how the RosettaNet models for POrequest and confirmation were imported into the InfoPath forms design tool--to create forms which could be hosted by 2007 Microsoft Office System on the server and rendered thin in client browsers. The form is just an XML presentation view layered on top of the RosettaNet XML data model, and all the server-side application logic is in terms of these same data models. An advantage of this approach is that it leads to lose coupling between the view (the form) and the model (the server-side application logic). The layout of the form could change and different sets of schema elements could be bound to form UI controls, but this would not change the data models themselves or the syntax of server-side interfaces. Naturally the view and the model are not completely de-coupled, because the semantics of the application logic might need to change if additional schema elements are added to the view.

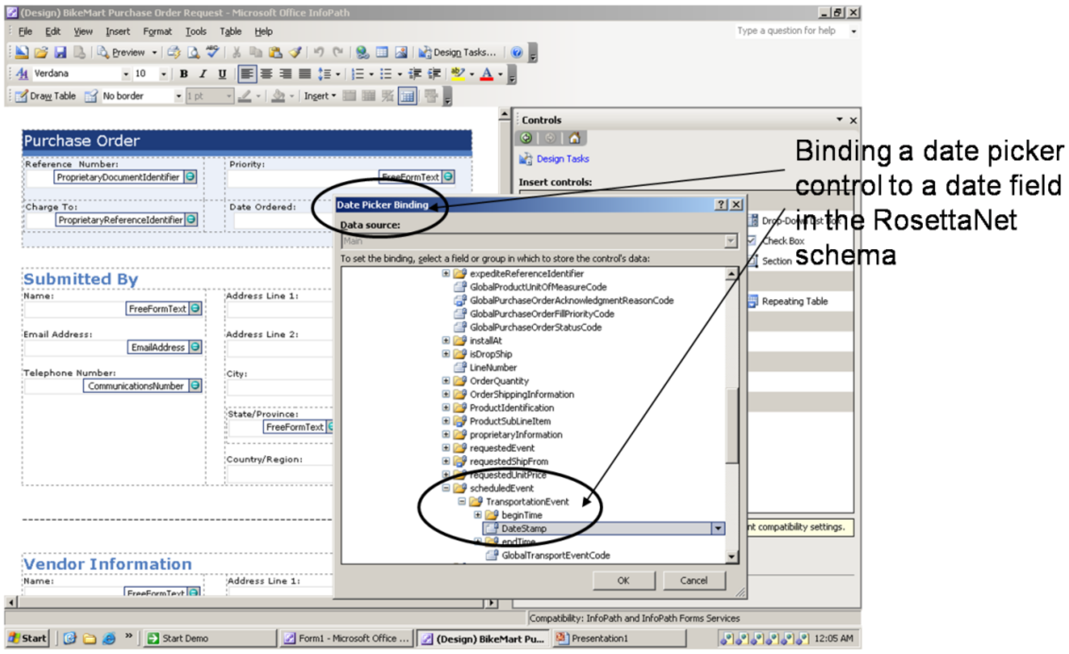


Figure 16: PO request and confirmation forms were built using the graphical InfoPath forms designer, after importing XML schemas from RosettaNet

A similar approach was taken to create documents for the strategic forecasts exchanged between the trading partners. XML schemas from the RosettaNet specification were used for this purpose and bound to Excel spreadsheets using XML maps. This is a great way to associate spreadsheet cells with XML schema elements, so that XML data can be written-to and read-from spreadsheets. This binding between XML data and spreadsheet cells is loosely-coupled, as users can rearrange layout of the cells. Nonetheless, Excel will still track the association to schema elements.

However, there are a couple of restrictions to be kept in mind here. First, XML maps cannot handle certain kinds of complex schemas, such as where there are collections of collections. Second, spreadsheets registered with Excel services in 2007 Office System cannot contain XML maps. There are a few ways to work around these issues.

When the schema for the data model is too complex for XML maps, a simplified (flattened) schema can be used and server-side processing can transform from one format to another. As

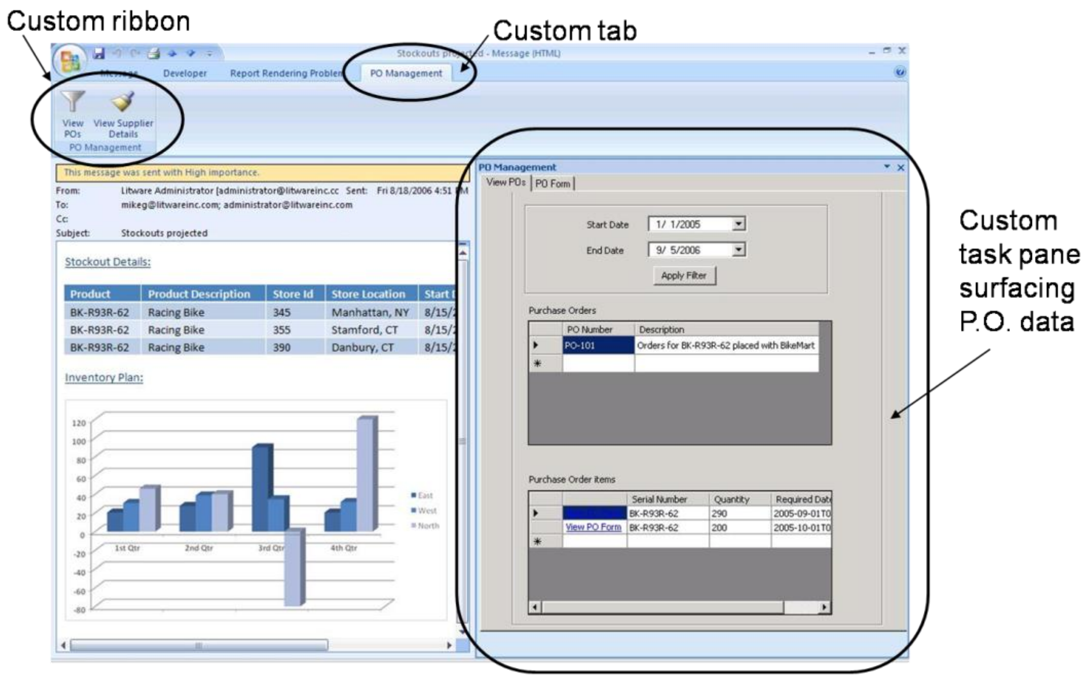
2007 Microsoft Office System documents are saved as XML, using the Open XML schema, this opens the door for server-side XML processing that was not possible with earlier versions of Office. These kinds of server-side logic can be built into packaged Workflow Foundation activities that are assembled into workflows in SharePoint and can transform documents as they arrive. Also, these packaged activities might be code or even an XSL style sheet.

The restriction on using XML maps with spreadsheets registered with Excel services was not a major issue for the scenarios in the reference implementation. These spreadsheets represent working documents that are in-process and that are used by Information Workers in the strategic forecasting business processes. These documents are generated by workflows, edited by people, and then de-constructed by other workflows which turn them from documents into messages and data. This usage pattern does not fit the three key scenarios that are mainly targeted by Excel services, which are:

- Sharing thin (browser) views of spreadsheets.
- Building BI (business intelligence) dashboards.
- Turning Excel spreadsheets into backend application services.

Therefore, those Excel spreadsheets with XML maps were stored in a document library that was not registered with Excel services, and server-side processing was built into Workflow Foundation activity libraries to strip out the XML maps before those spreadsheets were consumed through Excel services.

As shown in Figure 17, an add-in was added to Outlook using VSTO. This added PO management capabilities through a custom ribbon, tab, and task pane. An implication of this is that this add-in is a component that needs to be deployed on the client machine, instead of on the server.



This message was sent with High importance.

From: Litware Administrator [administrator@litwareinc.cc Sent: Fri 8/18/2006 4:51 PM
To: mikeg@litwareinc.com; administrator@litwareinc.com
Cc:
Subject: Stockouts projected

Stockout Details:

Product	Product Description	Store Id	Store Location	Start
BK-R93R-62	Racing Bike	345	Manhattan, NY	8/15/
BK-R93R-62	Racing Bike	355	Stamford, CT	8/15/
BK-R93R-62	Racing Bike	390	Danbury, CT	8/15/

Inventory Plan:

Start Date 1/ 1/2005
End Date 9/ 5/2006

Purchase Orders

	PO Number	Description
▶	PO-101	Orders for BK-R93R-62 placed with BikeMart
*		

Purchase Order items

		Serial Number	Quantity	Required Date
▶		BK-R93R-62	290	2005-09-01T0
	View PO Form	BK-R93R-62	200	2005-10-01T0
*				

Figure 17: Support for managing POs was added to Outlook, through an add-in

Adding Server-Side Storage

SharePoint sites were set up for various departments in the reference implementation: sales, marketing, and production. In real life these could either be departmental sites, or team collaboration sites. Setting up sites implies assembly and configuration using the SharePoint administration tools, rather than development.

Once the sites were set up, server-side storage of in-process documents was provisioned on these sites by creating document libraries to hold strategic forecast spreadsheets and PO forms.

Creating Workflows and Orchestrations

For the reference implementation, business process representations were created in two places:,

- Workflows were deployed into SharePoint, to wire together the document libraries that had been set up in the previous step. These workflows performed server-side processing of PO forms and strategic forecast spreadsheets after they had been saved to the document libraries. In the case of POs, the workflows get triggered whenever a form is saved to the library – either as a new form or as a change to an existing form. However, the forecasting process is much more iterative in nature, and so, in the case of strategic forecasts, the workflows need to get initiated only when the forecast is complete and ready to be published. Therefore these workflows get initiated explicitly by the information worker, by starting them directly from the client menu – although the workflows do still run on the server.
- Business processes were also modeled outside the SharePoint sites. These are long-running processes that manage the lifecycle of individual business entities (such as POS and forecasts) or the lifecycle of individual business processes (such as, procure-to-pay). Although the right technology to use here – to get better manageability – was BizTalk, for the purposes of the scenario, these business processes were actually implemented using state machines from Workflow Foundation, and run as Web services.

Before these sets of workflows could be assembled, a domain-specific workflow library was created with activities for processing PO and forecasts. These workflow activities were then packaged into an assembly, and then workflows were assembled from these building blocks.

Setting Up Services

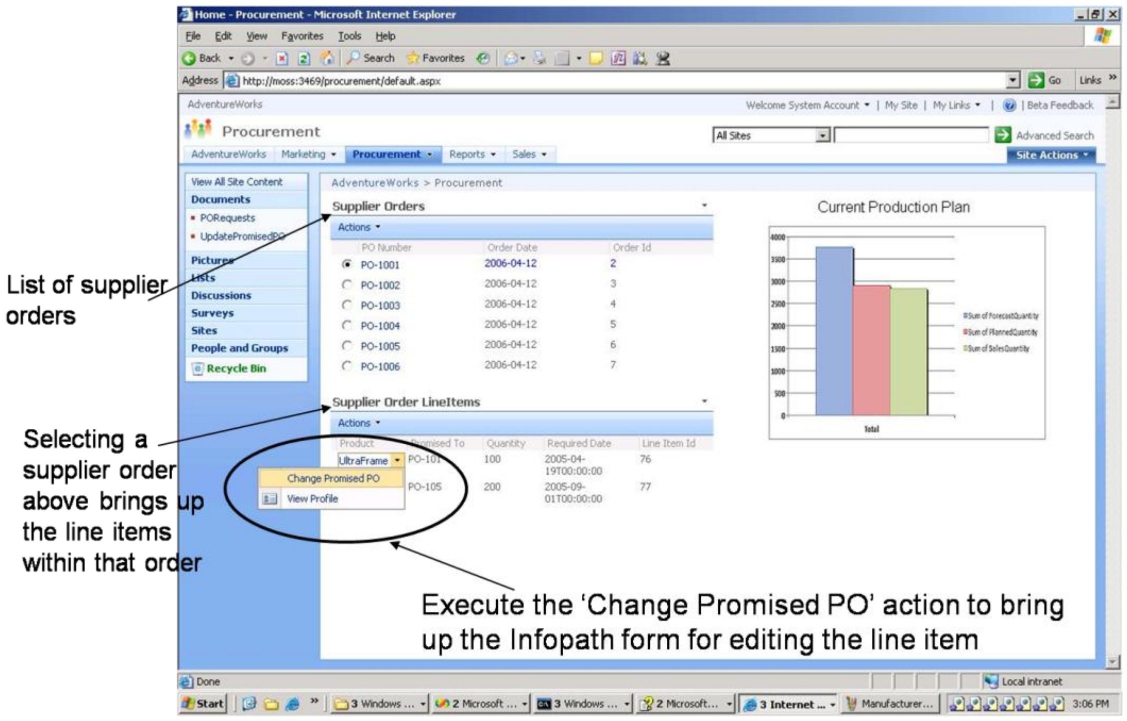
In the section above on generating industry models, we described how XML schemas and C# classes were generated from RosettaNet schemas. These were then used to build Web service interfaces to represent LOB systems. These interfaces form part of the assets that make up the composite application, because it should be possible to deploy the packaged process against any kind of IT landscape. So these Web services interfaces represent the touch points to backend IT systems needed to implement to integrate through Web services.

These Web services are invoked from the business activities that make up the workflows above.

Making Connections for Data

The reference implementation also showed how to create composite applications for cross-functional processes right into SharePoint by defining data entities in the Business Data Catalog (BDC) and consuming these entities from within SharePoint lists. The BDC is a shared service built into the 2007 Microsoft Office System, and it can be used to define the entities themselves, light weight relationships between entities, and actions that can be taken upon them.

For example, Figure 18 shows the user experience for one of these cross-functional processes. There are two lists on this SharePoint page – one for supplier order header information and the second list for details information. Selecting an order, from the list above, causes all the line items for that order to be displayed in the list below. There is a drop-down menu on the lower list that brings up selected information about the supplier order line item and displays it in a form for editing.


List of supplier orders
Selecting a supplier order above brings up the line items within that order
Execute the 'Change Promised PO' action to bring up the Infopath form for editing the line item
Home - Procurement - Microsoft Internet Explorer
http://moss:3469/procurement/default.aspx
Procurement
AdventureWorks > Procurement
Change Promised PO
View Profile
Current Production Plan
Sum of ForecastQuantity
Sum of PlannedQuantity
Sum of SalesQuantity
Total

Supplier Orders

	PO Number	Order Date	Order Id
●	PO-1001	2006-04-12	2
○	PO-1002	2006-04-12	3
○	PO-1003	2006-04-12	4
○	PO-1004	2006-04-12	5
○	PO-1005	2006-04-12	6
○	PO-1006	2006-04-12	7

Supplier Order LineItems

Product	Promised To	Quantity	Required Date	Line Item Id
UltraFrame	PO-101	100	2005-04-19T00:00:00	76
	PO-105	200	2005-09-01T00:00:00	77

Figure 18: Cross-functional application for editing supplier orders built into SharePoint

To set this up, BDC Web Parts are set up for the two tables. The first table is mapped to a BDC entity for Supplier Order, and the second table is mapped to a BDC entity for Supplier Order line item. The parent-child relationship between the two lists is modeled from within SharePoint. In addition, there is an action defined on the BDC entity for Supplier Order line item called "Change Promised PO" that brings up the InfoPath form for editing some details of the line item. An action is defined by a name (which shows up in the drop-down menu), a URL (which is where the selected row gets posted back to), and a set of attributes from the selected entity that get posted back to the URL (Figure 19). This leads to two possibilities – the URL might lead to a Web service that can process the data being sent back, or it could point back to an InfoPath form being stored on the server – which is happening here. The InfoPath form has a code-behind that takes the parameters being passed back to it and uses it to pre-populate the form.

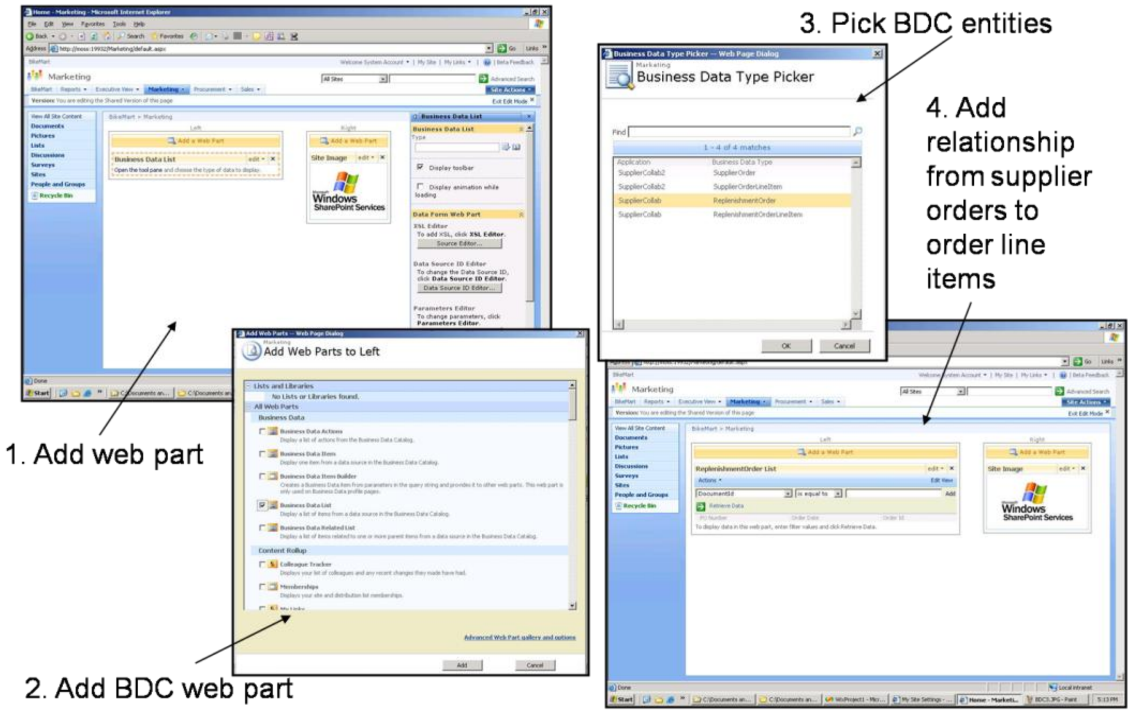

3. Pick BDC entities
Business Data Type Picker
4. Add relationship from supplier orders to order line items
1. Add web part
Add Web Parts to Left
2. Add BDC web part

Figure 19: Setting up the cross-functional process from Figure 18

The composite application also had assets for Business Intelligence (BI). There were SQL Services analysis services cubes for order fulfillment and a production schedule and inventory plan that are populated from the transactional data using SQL Services Integration Services. This information is plugged into BI dashboards in SharePoint. Dashboards were also set up from spreadsheets stored in Excel services and SharePoint lists. Finally, the Business Scorecard Manager tool was used to create reports that plugged into SQL Server Reporting Services on top of the transactional data.

Connecting Business Processes Within the Enterprise

A typical systems landscape for a manufacturer is shown in Figure 20. Some of the important terms are explained here:

1. Enterprise LOB Applications are corporate LOB applications such as ERP, SCM, and CRM systems.
2. Factory LOB Applications are LOB applications at the level of a manufacturing facility.
3. MES: Manufacturing Execution System>
4. SCADA: Supervisory Control And Data Acquisition. A distributed instrumentation and control system.
5. HMI: Human Machine Interface. The user interface (typically graphical) for operating and monitoring the control system.
6. DCS: Distributed Control System. A networked system that allows remote monitoring and control of industrial equipment through connected:

   Sensors – Typically a measuring device for operational data.

   Controllers – Such as PLCs (Programmable Logic Controllers).

   Actuators – Devices that transform electronic signals into motions.

   Operator Terminals – These could be desktop computers distributed throughout the shop floor or even mobile devices which can connect wirelessly to sources of information.

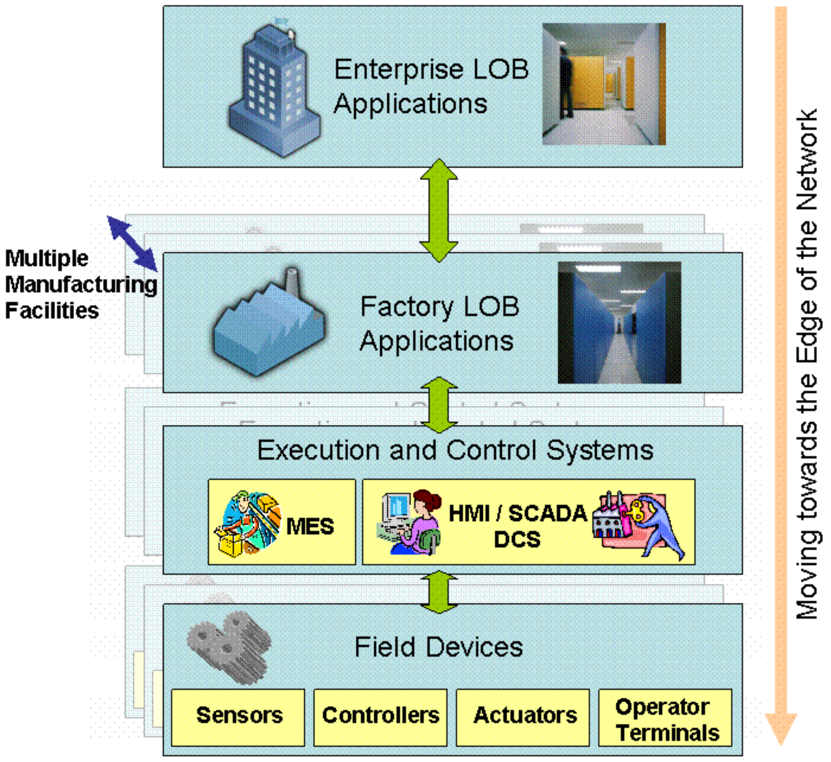

Enterprise LOB Applications
Multiple Manufacturing Facilities
Factory LOB Applications
Execution and Control Systems
MES
HMI / SCADA DCS
Field Devices
Sensors
Controllers
Actuators
Operator Terminals
Moving towards the Edge of the Network

Figure 20: Typical high-level systems landscape for a manufacturer

The Coming Move to Processing at the Edge

Technology advances are enabling a new generation of field devices, and soon there will be a lot more computing power distributed close to the edge. These changes will be driven by the rapid adoption of radio frequency identification (RFID), the emergence of new and powerful mobile devices, and in the future by sensor networks. As more applications and devices are deployed at the edge, there will be a greater need to manage these devices and the data streams they generate. Thus the proliferation of edge devices can be expected to push the deployment of edge servers to accomplish these tasks. In general, the architecture for these edge devices and edge servers will look like Figure 21.

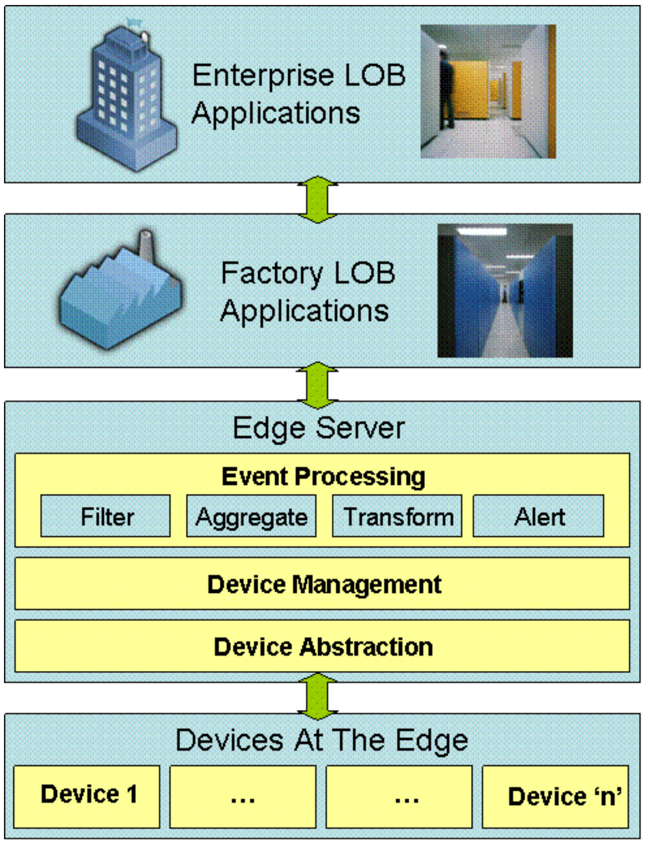


Figure 21: Proliferation of new types of edge devices will push deployment of edge servers into the IT systems landscape

Take for example the case of RFID technology, which is making it possible to gather greater quantities of data at the edge. Microsoft is extending its platform to provide the services in Figure for RFID. Here, the devices corresponding to the lowest layer in the stack will primarily be RFID readers. These readers will be generating events as and when it detects RFID tags, and then these events will get passed along to the edge server for processing. Within the edge server, an RFID service will deliver the device abstraction, device management, and event processing capabilities.

The device abstraction layer in the RFID service will ensure that devices from different vendors look the same to the device management and event processing layers and also to the business applications being fed by the edge server. The device management layer will provide a single, consistent way to deploy, configure, and monitor RFID devices. The edge processing layer will make it possible to filter, transform, and aggregate events raised by RFID devices to ensure that business applications see events in the proper context. A further complication is that as devices get smarter, some of the processing from the edge server will get offloaded to the devices (things like filtering); capabilities of devices will vary across vendors--and yet the device management layer of the edge server would need to expose these varying capabilities through a consistent interface to manage devices collectively as a group.

Another disruptive change expected soon, is the broad industry support emerging for Web services. This will start a trend to introduce Web services at each of the different layers of manufacturing systems in Figure 21. For example, “smart” field devices (sensors, actuators, controllers, and so forth) might expose Web services interfaces. These could be functional interfaces for setting and retrieving information or could be management interfaces to configure and deploy the device. Client applications – such a Human Machine Interface (HMI) system or a management console - could then connect directly to the device. Some standards bodies such as the OPC Foundation are releasing Web services specifications to ensure interoperability among vendors.

Managing Information Flows Across the Enterprise

Enterprises want to integrate their production systems with their business systems, to achieve significant increases in productivity and to drive decision making into the factory floor. Plant personnel should be able to monitor the health of their systems and to be alerted to exceptions and potential problems. They should also be able to gain insight into the business metrics that will be affected by problems in production. However, this requires sophisticated, yet easy to use, tools, which can scale across complex systems landscapes.

Also, as sensors and other devices become more sophisticated and pervasive, the potential to monitor hard-to-reach manufacturing equipment increases, which gives manufacturers the potential to improve their industrial automation and drive down costs. For example, the Air Force Research Laboratory was awarded a patent to monitor conduits such as electrical wires and hydraulic lines. This allows the detection of stresses to the conduit, along with alerts to potential leaks, while avoiding time-consuming manual inspections. To make this even better, these ‘smart’ devices at the edge could be connected into enterprise systems ,through edge servers, with unified management consoles and the ability to distribute reports. One potential high level architecture to do this is shown in Figure 22.

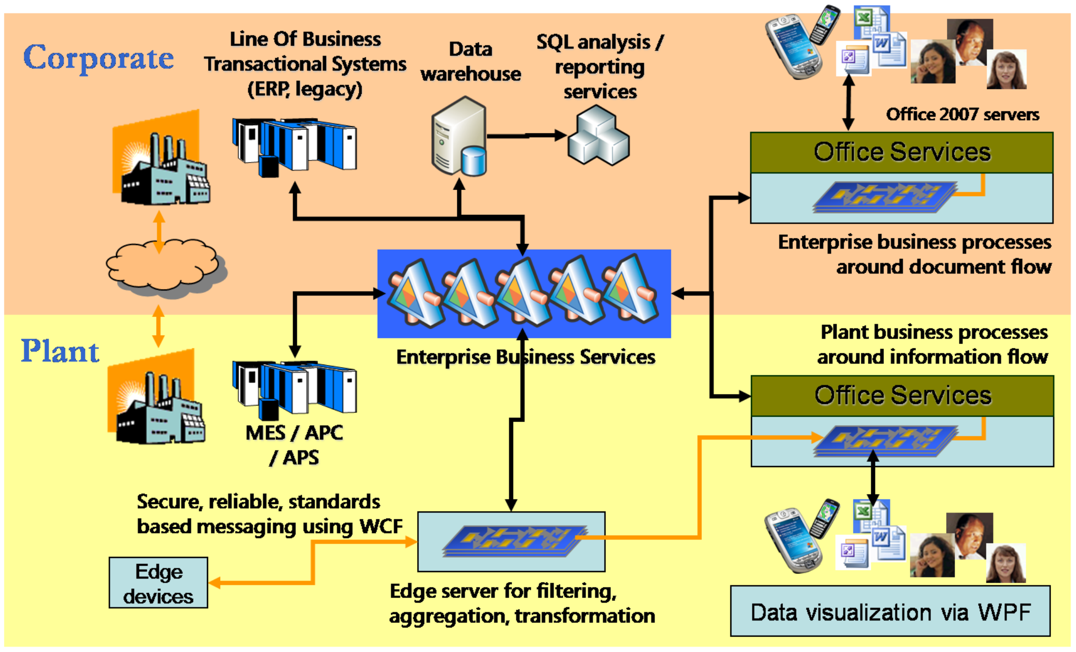

Corporate
Line Of Business Transactional Systems (ERP, legacy)
Data warehouse
SQL analysis / reporting services
Office 2007 servers
Office Services
Enterprise business processes around document flow
Enterprise Business Services
Plant
Plant business processes around information flow
Office Services
MES / APC / APS
Secure, reliable, standards based messaging using WCF
Edge devices
Edge server for filtering, aggregation, transformation
Data visualization via WPF

Figure 22: Integrating business processes across the enterprise, from plant floor systems to enterprise applications

Role-Specific Guidance for Building and Deploying Composite Applications

What are some of the cross-functional processes for which composite applications might be built and deployed? Some examples of potential areas within the manufacturing vertical are as follows:

- Managing the value chain such as solutions for supply chain management or distribution network management.
- Managing customer centricity as in CRM solutions.
- Managing production or manufacturing solutions for plant operations or lean manufacturing.
- Managing product innovation, as in solutions for collaborative product design or new product introduction.

Roles in Product Design, Engineering, R&D

An increase in product complexity, coupled with greater customer expectations, has put pressure on research and development. From an article titled “New Product Facts” in the *Economist*, “With the pace of innovation heating up, any enterprise that fails to replace 10% of its revenue stream annually is likely to be out of business within five years.”

Pain points

- Pressures to reduce new product time-to-market by optimizing R&D processes.
- Need to identify successful products faster in the research cycle to better manage the development effort.

- Greater need for secure real-time inter-departmental and external partner collaboration throughout product lifecycle process.

Implications for composition

There is a need for solutions that integrate product design and engineering with other business processes for production and demand generation. This requires greater collaboration among teams, often distributed globally when design teams are not co-located with manufacturing teams. Platform capabilities required are real-time communications, document collaboration, and data synchronization between Product Data Management and Product Information Management systems and other enterprise applications.

Roles in Sales, Customer Service

Mounting pressures for collaboration, coupled with increased market demands, have created new challenges for sales and marketing. From the Gartner article, "Predicts 2004: Customer Service, From Function to Process," comes this quote: "Customer service will continue to be the 'litmus test' of an enterprise's commitment to the customer, and as such will be a key indicator of the integrity of the enterprise."

Pain points

- Shrinking product life cycles demand changes to marketing strategy with a greater focus on customer issues and complaints.
- Greater need for visibility and participation in product development to provide early feedback on customer experience that may impact warranty and service cost.
- Effective customer response through reduced errors and shorter service times is becoming a competitive advantage.
- Ability to service warranty issues impacts bottom-line profits.

Implications for composition

For better customer service levels, enterprises need to synchronize demand (sales and marketing), with supply (production), and product development (engineering). This requires solutions beyond those offered by traditional CRM systems. Some of these industry solutions are:

- **Demand shaping/management** – Ability to influence / shape demand through multiple channels (such as customers, markets, partners, or retail).
- **Pull-based replenishment** – Transform from push (pushing products to market) to pull (market pulls what it needs and supply chain responds appropriately) as in Kanab, lean manufacturing, and other frameworks.
- **Demand fulfillment solutions** – Improve service levels through better order promising allocation planning, and inventory planning solutions.
- **Integrated innovation** – Product design and engineering integrated to supply (production) and demand. That is, design only what you can build and sell, and engineer products in a way that production can be scaled up or down in response to market needs without requiring excess inventory.

Roles in Manufacturing, Operations

Increasing pressures on operations from networked supply chain, R&D, product development, and customer service initiatives have changed the playing field for business decision makers in manufacturing and operations. This is highlighted by a report from IDC research which states, "With increased product complexity, increased customer demands around product performance, and correspondingly decreased product lifecycles, manufacturers are being pressured to deliver better product cheaper and faster."

Pain points

- Increased emphasis on lower inventories, higher inventory turns, and shorter cycle times demand "lean" production techniques.
- Lack of visibility into supply chain and plant floor operations leads to unreliable production schedules, higher inventory buffers, and erratic order fulfillment.
- Pressure to reduce costs, increase throughput, and respond to changing customer needs requires better and more secure information sharing while protecting confidential information.

Implications for composition

Improving operations usually requires active monitoring of events and information to get more pro-active decision making. This usually requires real-time, or near real-time, flows from enterprise resources (sources of data) to business decision makers (consumers of information). This is complicated because data exists in multiple locations; is being generated at multiple frequencies and formats; and often needs to be cleansed, aggregated, transformed, and correlated before it can become meaningful information. Also, the role of the business decision maker should determine access rights and presentation of information. Furthermore, when exceptions or un-planned events occur (and they will), resolution of those events typically impact people and activities involved in multiple business processes. Usually such kinds of cross-functional applications do not exist in the enterprise, but with the right kinds of IT assets in place, a platform for composition will enable rapid assembly and configuration of composite applications to meet these needs.

What will these composite applications look like? They will provide role-based views of operational data, with real-time (or near real-time) visibility into distribution, procurement, inventory, and operations. For example, some of this information might include demand forecasts, supply commitments, inventory positions, purchase orders, delivery schedules, advance shipment notifications, and goods receipts.

Chapter 5
Financial Services OBA

Introduction

This chapter will walk you through a loan origination scenario. This scenario is tailored for the banking industry. It will provide guidance for making architecture decisions around real-world business and technology issues that plague the banking industry. It will also help you identify opportunities in your business for using Microsoft Office SharePoint Server (MOSS), BizTalk, and SQL Server. Using these, we will show:

- How 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 (OBA) through these reusable services layers.

The MOSS 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)** – That can be leveraged to publish dynamic and user-manageable content.
- **Workflow** – Using 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 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 line-of-business (LOB) data to the presentation layer.

Building feature-rich composite-style applications is the goal of OBAs. Using this technology, financial services organizations can migrate to an SOA with more ease than ever before. This paper will discuss both integration and composablity issues between applications, and how these can be accomplished in several different ways.

The first and most mature way to integrate is by focusing on the integration of data. This approach, however, 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. However, since 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 due to a lack of consistency.

We will use a 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.

Enabling Technologies

- BizTalk – By using BizTalk, banks can now orchestrate their business processes in a dynamic and fluid way.
- .Net Framework 3.0 (WF, WCF, WPF)
    - 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 Framework (WCF) will reduce the increased complexities of the integration needs. For example it will cover integration of: 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 documents such as Excel, Word and PowerPoint.
- 2007 Microsoft Office System (SharePoint, Excel Server, documents)
    - SharePoint 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.
    - Excel Server can host your Excel data in a central store with robust relational database features that will provide protection of data, centralized backup procedures, a single version of the truth, and security around sensitive data. This will provide the Excel user experience for multiple services without cross-training on other applications.
    - Document formats have been changed to an open document format: Open XML. This will ensure 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 work force is becoming increasingly important for banks.
    - Windows Mobile – Mobile devices can be leveraged in the field for appraisal visits and zoning.

Financial Services Business and Technical Challenges

In the context of composite applications, there are quite a few challenges in financial services. Banks and insurance companies are starting to push more and more services to the edge. Things like ordering stamps at an ATM, from a consumer perspective.

For this paper, the focus will 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
- Correspondent
- Telesales

- Retail

There are some commonalities among these, but there is also exclusive behavior as well.

Loan Origination 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. As well, compiling and staying on top of regulatory laws is always a challenge given the turbulent changes happening in the industry today. Laws such as 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 application demonstrates alignment with key business processes within the loan origination. Since we are only taking 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 (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 (30-year fixed) or their portfolio products (products that the bank funds and usually doesn’t sell).
- **Application or Registration** – When the customer enters the proper qualification information for a loan product.
- **Processing or Locking Loan** – When a customer agrees to a specific product and rate and “locks into a commitment.” This stage kicks off many sub-processes 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.

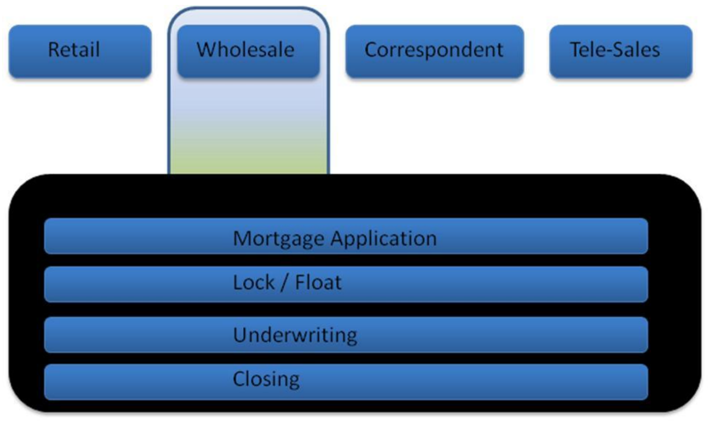


Figure 1: Business channels

With this solution, it is important to understand the amount of complexity both in the business processes and the current state technology landscape. The technology landscape will be defined in the technical architecture section below.

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.

Keep in mind that when architecting solutions, the technology components are just one of the pieces. There are human elements when architecting solutions. So not only do we take the system-level processes into account but also the human or manual processes to give the architect the complete perspective.

Understanding loan origination is not easy due to its complexities. However, comprehending some core concepts of loan origination will make things easier.

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

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

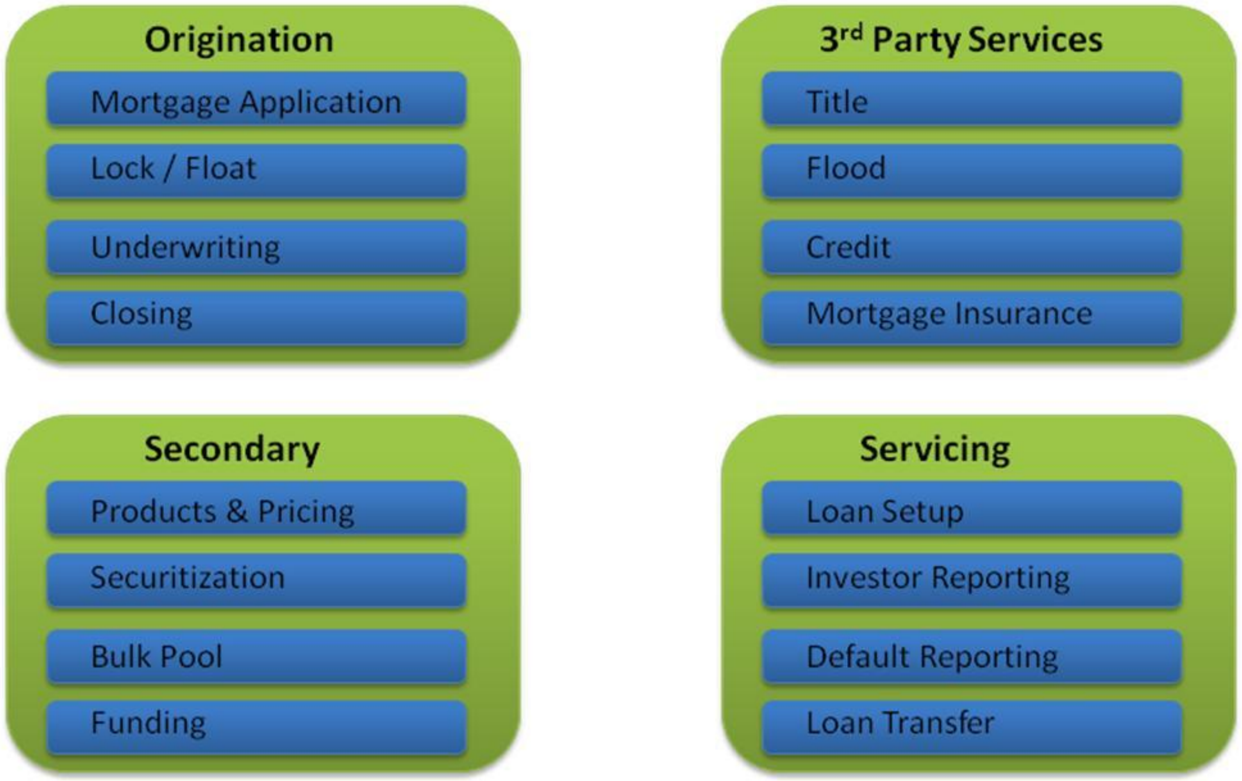


Figure 2: Business capabilities

Briefly, these capabilities are described as follows:

- **Origination** – The process of acquiring the customer, 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 may 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. Listed below are some of the challenges the loan industry faces. The goal is for this architecture to aid in the simplification or elimination of these issues.

- Workflows are complex and difficult to manage. These are often based on exception processes, particularly portfolio loans, region and typically will have 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, below.

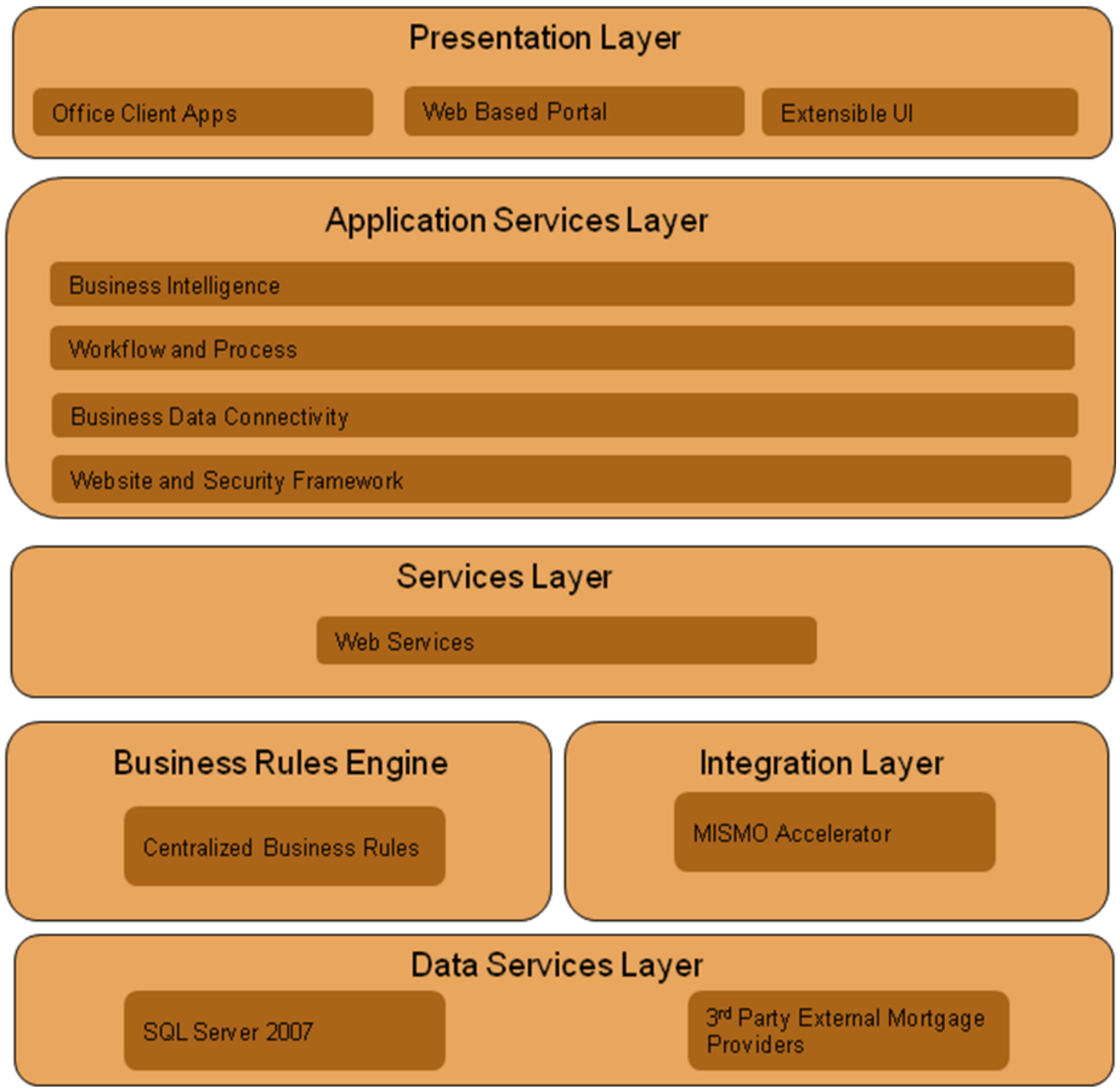


Figure 3: Logical Architecture

Microsoft Office SharePoint Services Architecture

The MOSS architecture is really several logical layers (Figure 4). This consists of the:

- **Presentation Layer** – Serves as the user interface. ASP.Net 3.0 Web forms hosted on the Windows SharePoint Portal Server. 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 MOSS layer will use Windows Communication Foundation. The Integration layer will use BizTalk.

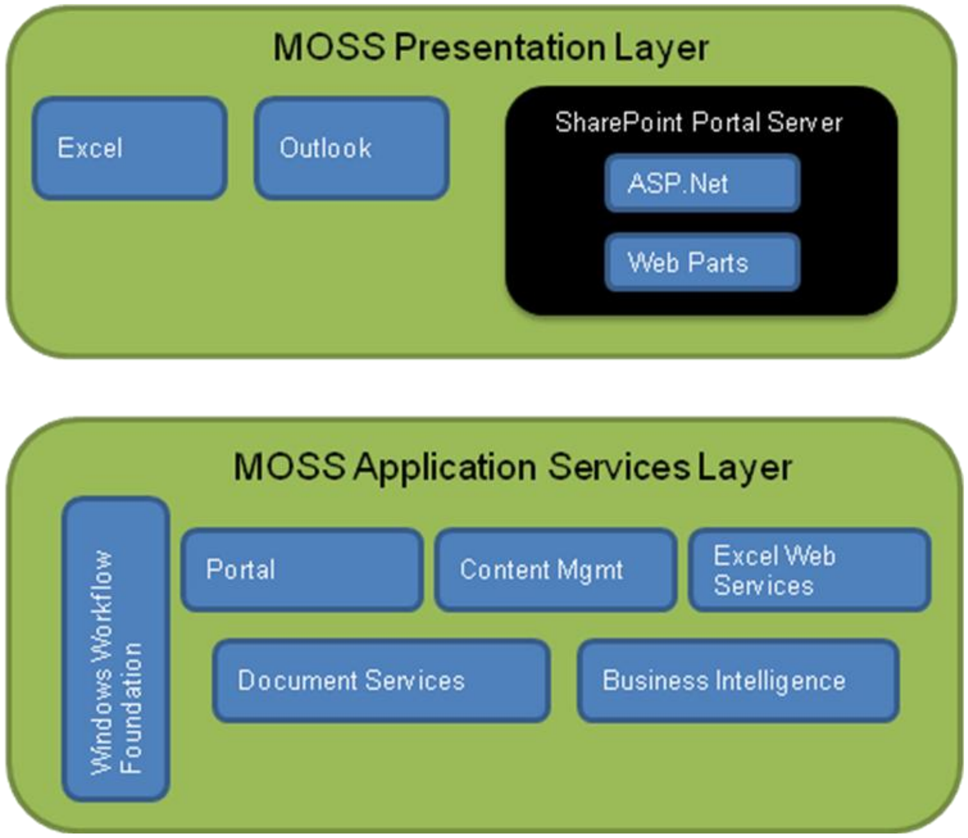


Figure 4: The MOSS layers

Presentation Layer

There were several options when considering the user interface architecture. The Presentation Layer will be a Web- based application. It will expose MOSS Web pages from here. For security and scalability reasons the presentation and business logic layers are separated.

- **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 rich of a presentation was not required by the solution. However, it could be a fit if organizations want that rich interface.
- **ASP.Net Web Forms** – The .Net form of Web pages also referred to as “aspx pages.” Web Forms were chosen as the UI of choice for this solution for their ease of customization and more application-centric presentation.
- **InfoPath Forms** – Form-based services used when data entry forms are required. These were not chosen because the application was process-oriented and not data-oriented.

Below is a comparison chart of the technologies we used.

Criteria	Web Forms	InfoPath Forms
User Experience	Application-centric	User entry/ form-centric
Content Control	Allows customized or fine-grained control over page elements	Dynamic UI is possible by creating multiple views of the same form
Page Flow and Navigation	Need custom code needs to handle navigation	Need custom code needs to handle navigation
Development environment	SharePoint Designer/Visual Studio	InfoPath Client or VSTO
Integration with SharePoint	Need to add pages need to SharePoint sites. Need custom code to persist data from the page into a document library.	Need to deploy forms need o a SharePoint form library. After filling a form, the data can be automatically stored in the form library as XML.
Validation of page data when user enters values.	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 map XML schema to Web page fields	Native support for binding XML schema to page fields
Reusability	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 ASP.Net 2.0	Built using ASP.Net 2.0 and Forms Services which is part of MOSS

Table 1: Comparison of technologies

Using the Web Forms, you can integrate SharePoint Web Parts in a more appealing way. In addition, you gain content management. Examples on 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 MOSS portal platform.
- User interfaces to manage and version content within the application.

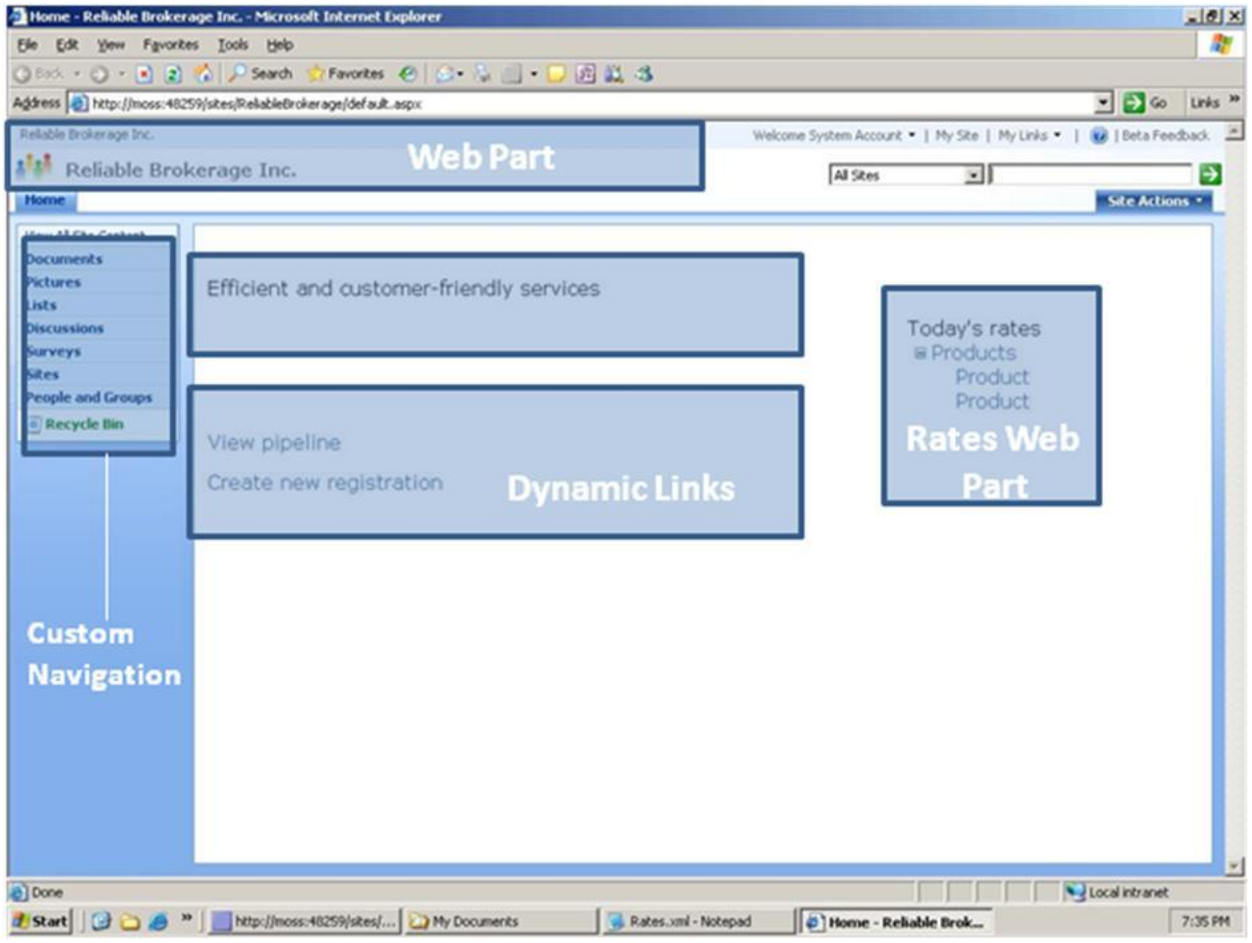
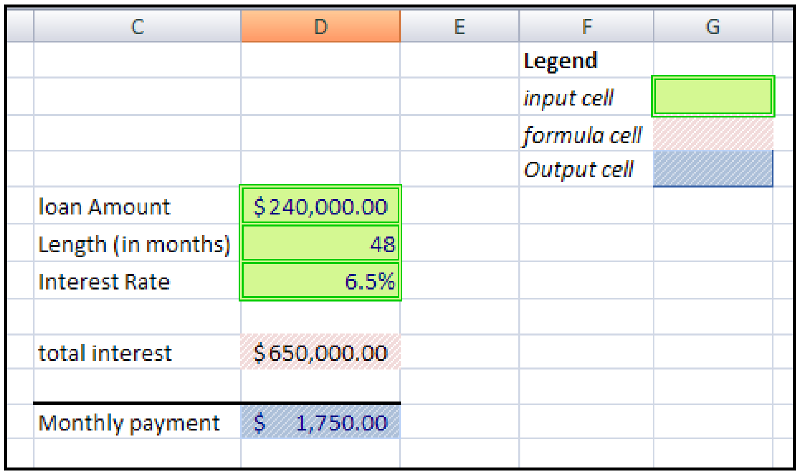


Figure 5: SharePoint user interface

The greater portion of this page design comes from Web Parts, befitting the composite architecture of the application.

Office Clients

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



C	D	E	F	G
			Legend	
			input cell	🟩
			formula cell	▨
			Output cell	▨
loan Amount	$240,000.00			
Length (in months)	48			
Interest Rate	6.5%			
total interest	$650,000.00			
Monthly payment	$ 1,750.00			

Figure 6: Using Excel as a client

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

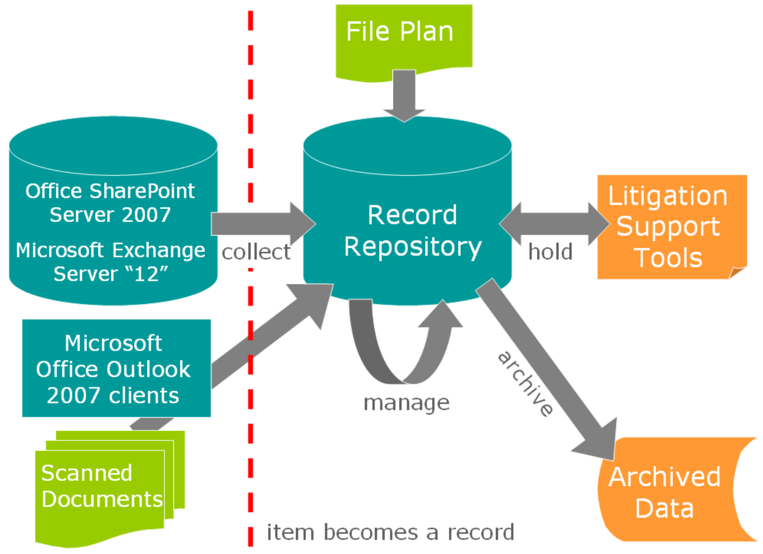


Figure7: Solving regulatory issues with MOSS technologies

These capabilities will also aid in auditing and regulatory issues around records management requirements as shown in Figure 7.

The records repository has several features that help ensure the integrity of files stored there. First is the ability to ensure that records are never automatically modified by the system. This means that records uploaded to a records repository and then downloaded later will be identical--byte for byte. Second are the default version and the audit settings that monitor changes to content to prevent direct tampering with records. Third, records managers can add and maintain metadata on items separately from the records' metadata. This allows information such as who manages the item to be changed without modifying the underlying record. Changes to this metadata are versioned as well.

Application Services

The following built-in MOSS services (Figure 8) 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.
- **Excel Web Services –** For all calculations.
- **Document Libraries** – Provides management functionality for auditing and versioning.
- **Windows Workflow Foundation (WF)** – For orchestrating the business process.

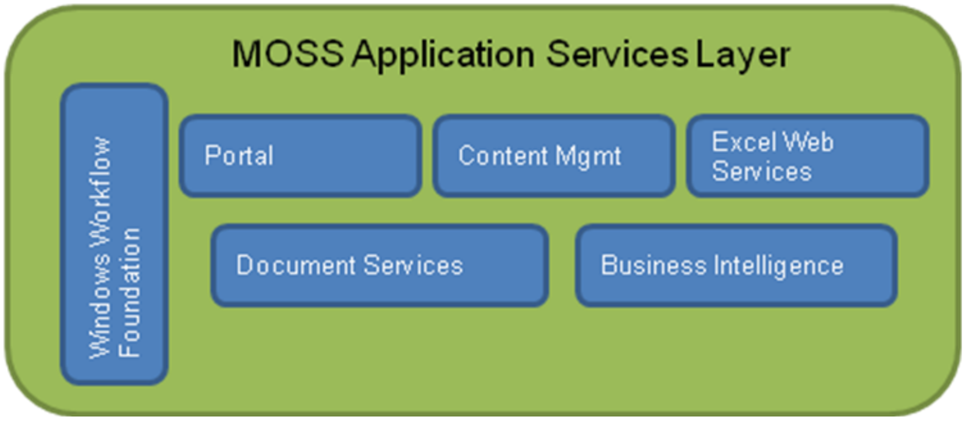


Figure 8: Application services implementation reference

WF and WCF Design Considerations

In this solution, you will notice that there are two workflows used. One is contained within MOSS and the other in BizTalk. The built-in workflow used in MOSS is Windows Workflow Foundation (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 is used to manage workflow for integration as shown in Figure 9. We will cover this in more detail in below.

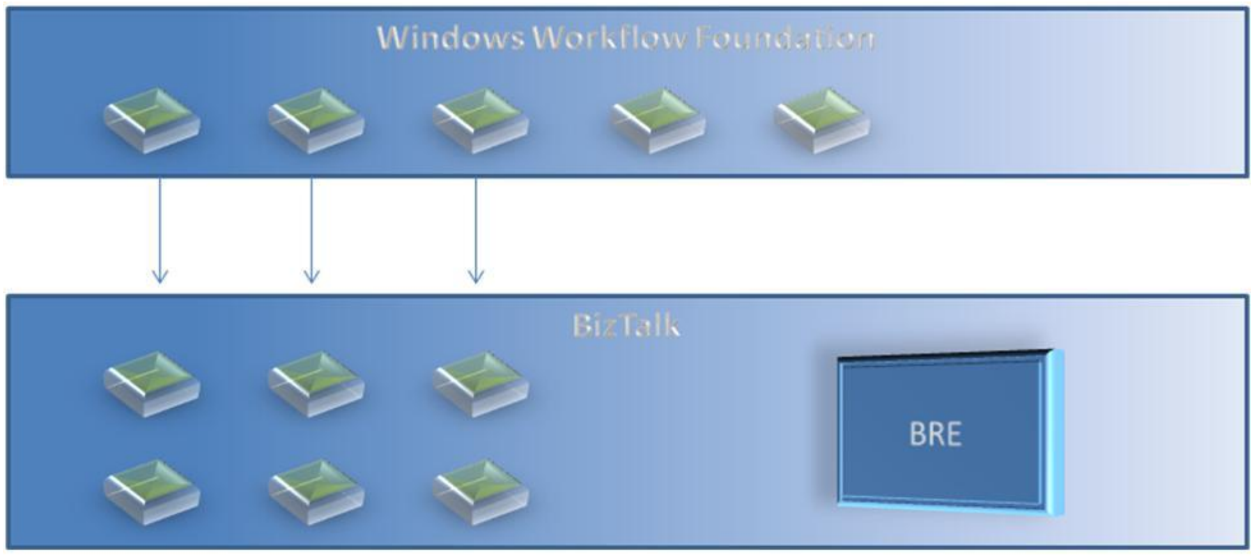


Figure9: WF interaction with BizTalk

Increasingly, there is a need to coordinate the interactions between real human actors in software. Humans are, of course, a key participant in almost every software system, especially in collaborative processes, and 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, a participant is out sick or on vacation, and provisioning a user.

Within human workflow (Figure 10) people communicate with various systems and other people in a business process implemented in software using a workflow model. In this model, pre-built 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.

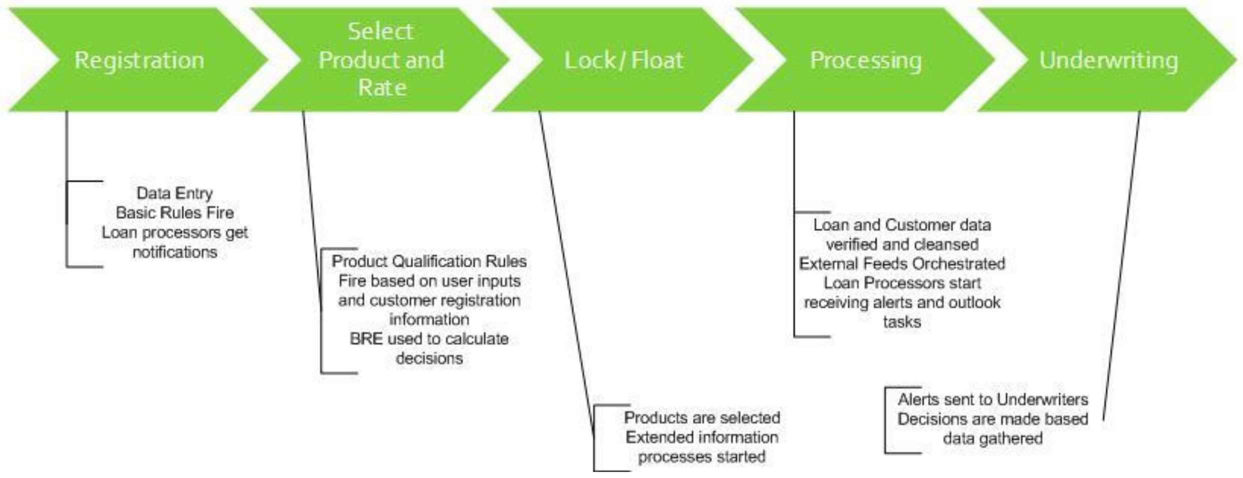

Registration
Select Product and Rate
Lock / Float
Processing
Underwriting
Data Entry
Basic Rules Fire
Loan processors get notifications
Product Qualification Rules Fire based on user inputs and customer registration information
BRE used to calculate decisions
Products are selected
Extended information processes started
Loan and Customer data verified and cleansed
External Feeds Orchestrated
Loan Processors start receiving alerts and outlook tasks
Alerts sent to Underwriters
Decisions are made based data gathered

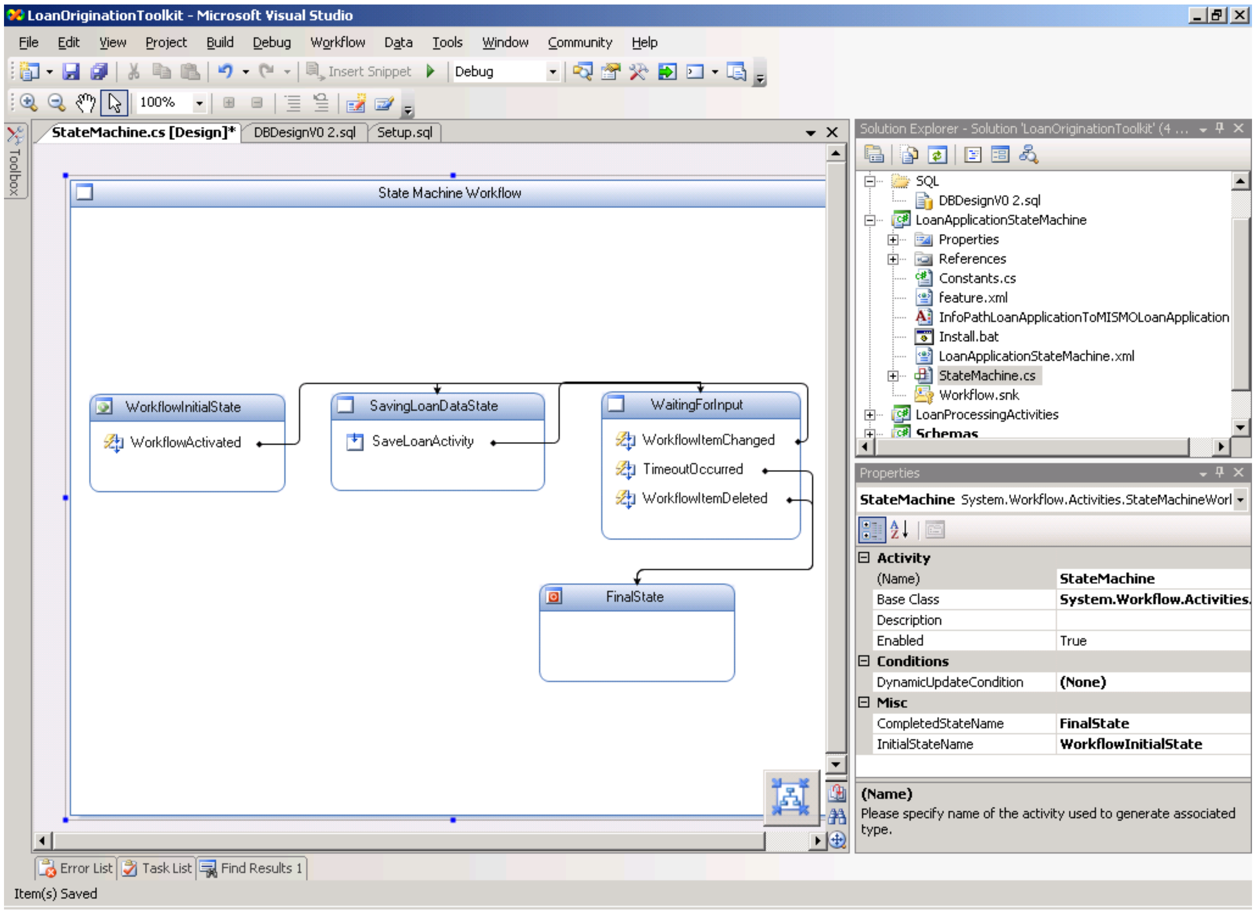
Figure 10: Business process modeled

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

- Products and pricing.
- Loan registration.

Here, we will discuss the over-arching 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 (Figure 11).
- **Pricing Workflow** – Used by the secondary marketing employees responsible for importing and applying daily rates on products.



StateMachine System.Workflow.Activities.StateMachineWorl

Activity	
(Name)	StateMachine
Base Class	System.Workflow.Activities.
Description	
Enabled	True
Conditions	
DynamicUpdateCondition	(None)
Misc	
CompletedStateName	FinalState
InitialStateName	WorkflowInitialState

(Name)
Please specify name of the activity used to generate associated type.

Figure 11: Master Loan Workflow

Integration 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, like 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 rather than just simply 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). By doing so, banks can slowly migrate off of redundant solutions or platforms.

The integration layer consists of Web services stubs exposed from MOSS in conjunction with the BizTalk Web services endpoints.

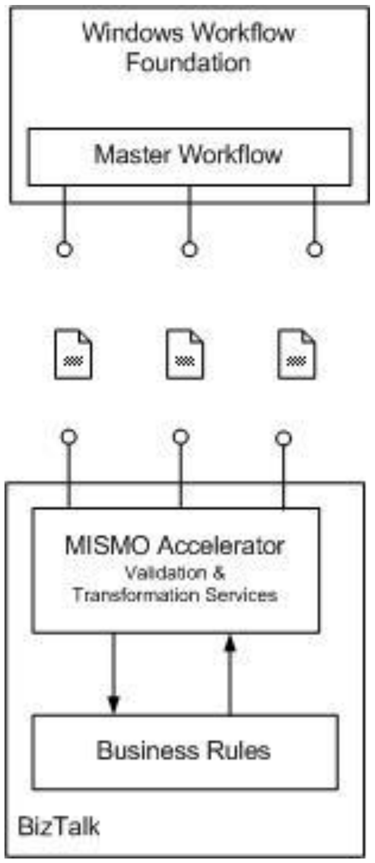

Windows Workflow Foundation
Master Workflow
MISMO Accelerator
Validation & Transformation Services
Business Rules
BizTalk

Figure 12:WF and BizTalk interaction

As shown in Figure 12 above, we used Web services to interoperate with MOSS. 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–complaint, 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 complaint at both WF and BizTalk layers to ensure interoperability (Figure 13).

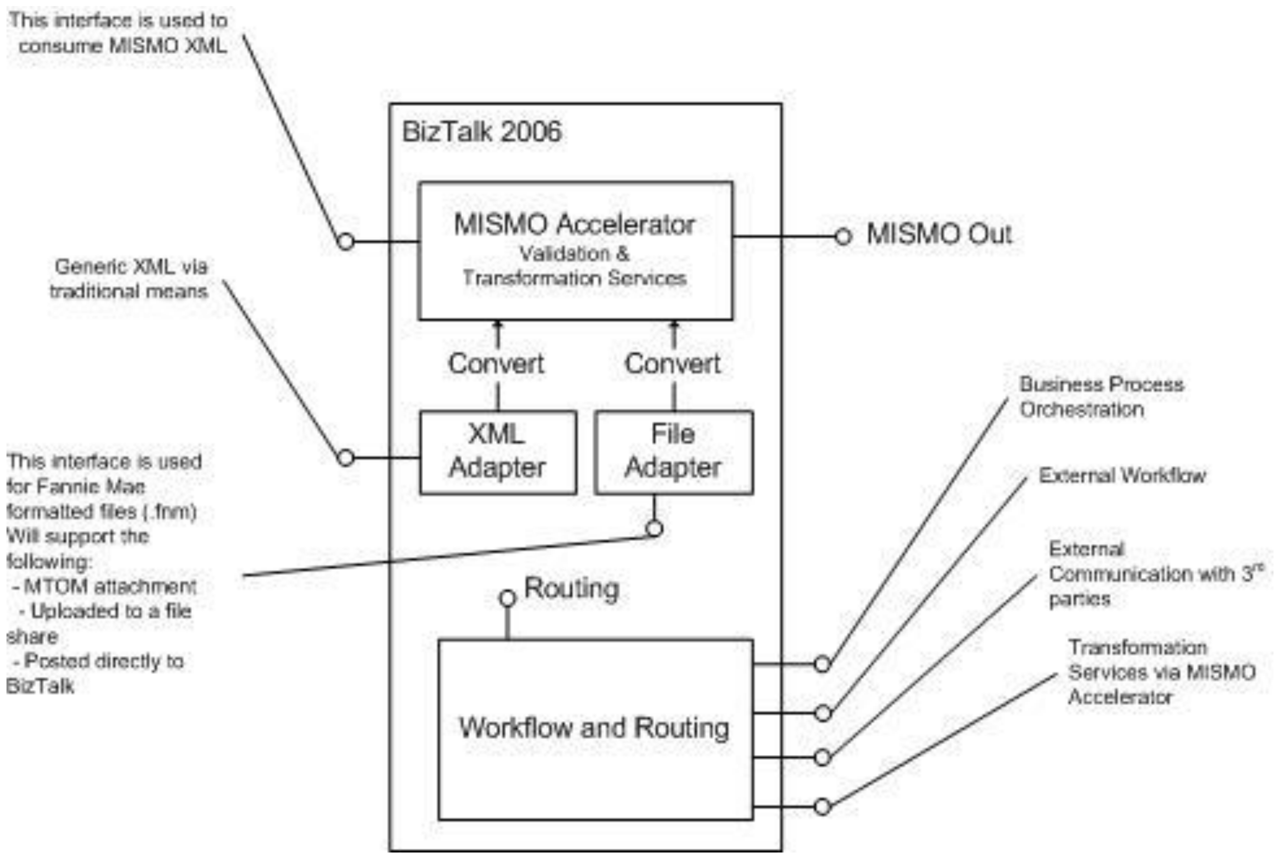

This interface is used to consume MISMO XML
BizTalk 2006
MISMO Accelerator
Validation & Transformation Services
MISMO Out
Generic XML via traditional means
Convert
Convert
XML Adapter
File Adapter
This interface is used for Fannie Mae formatted files (.fnm) Will support the following:
- MTOM attachment
- Uploaded to a file share
- Posted directly to BizTalk
Routing
Workflow and Routing
Business Process Orchestration
External Workflow
External Communication with 3rd parties
Transformation Services via MISMO Accelerator

Figure 13: MISMO Accelerator in BizTalk

Message Schemas

Throughout the solution it was critical to use standardized XML (Figure 14) to ensure standardized messaging and protocols for both compositability and extensibility. In addition to standardized XML, the MISMO schemas--which provide the proper business context where used – 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 may 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 need to be integrated. Most times these are used for specific business channels (for example, correspondent baking versus wholesale banking). In this architecture, BizTalk will coordinate the Web services using out-of-the-box Business Process Management (BPM) facilities.

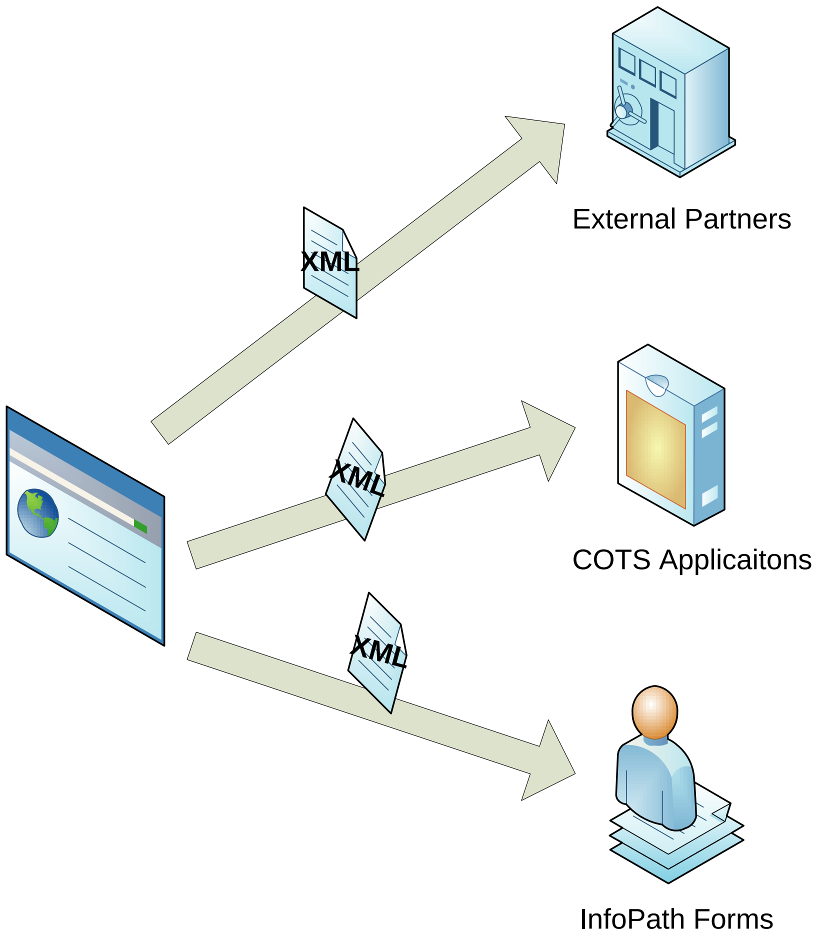
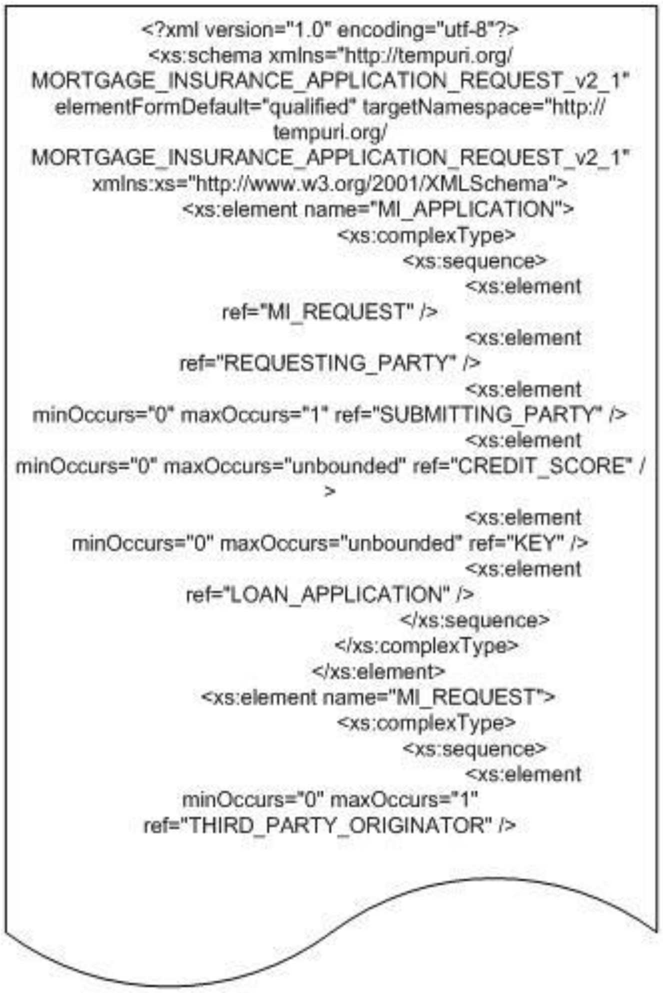


Figure 14: Extensibility of the platform

When using the MISMO schemas Figure 15 there were some gaps because the last update of the MISMO DTDs were a few years old. Listed below are the missing components from both a technical and business perspective:

- **Out dated 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" />
```

Figure 15: MISMO schemas used

Message Name	Sender	Receiver	Purpose
MortgageApplication	InfoPath	BizTalk orchestration and business rules engine	Register in loan system
RequestFlood	BizTalk Orchestration	Stub 3rd party provider	Retrieving the flood data about a property.
RequestCredit	BizTalk Orchestration	Stub 3rd party provider	Retrieving borrower's credit data.
RequestMI	BizTalk Orchestration	Stub 3rd party provider	Retrieving the property's

Message Name	Sender	Receiver	Purpose
			mortgage insurance
RequestTitle	BizTalk Orchestration	Stub 3rd party provider	Retrieving a property's title data.

Table 2: MISMO Messages Used

Message Name	Sender	Receiver	Purpose
Request3rdPartyServices	SharePoint Loan Workflow	BizTalk Orchestration	Retrieving the 3rd party data like Flood, Title etc.
Store3rdPartyResponse	BizTalk orchestration	Web service within SharePoint	Sending 3rd party data back to SharePoint
GetEligibleRates	SharePoint Loan workflow	BizTalk Orchestration	Retrieving the products that a borrower is eligible for
GetProviders	SharePoint workflow	BizTalk	Retrieving the list of providers for a specific loan

Table 3: Additional schemas created

Message Flow

Messages flow in and out of the centralized message hub using standardized MISMO messages. As discussed above. Additional messages needed to be developed for this solution.

Points to note:

- 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.
- Windows Workflow Foundation (WF) is the over arching control that coordinates the entire business process.
- Web services will be used primarily as the integration technology.

Orchestration Architecture

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

BizTalk 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 rational for this is the intensive business rules in loan origination, where 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 3.0 technologies. It is intended that these components will be called from BizTalk or exposed as Web services at an extensibility point. Since 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 and 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 or industry-specific nomenclature.

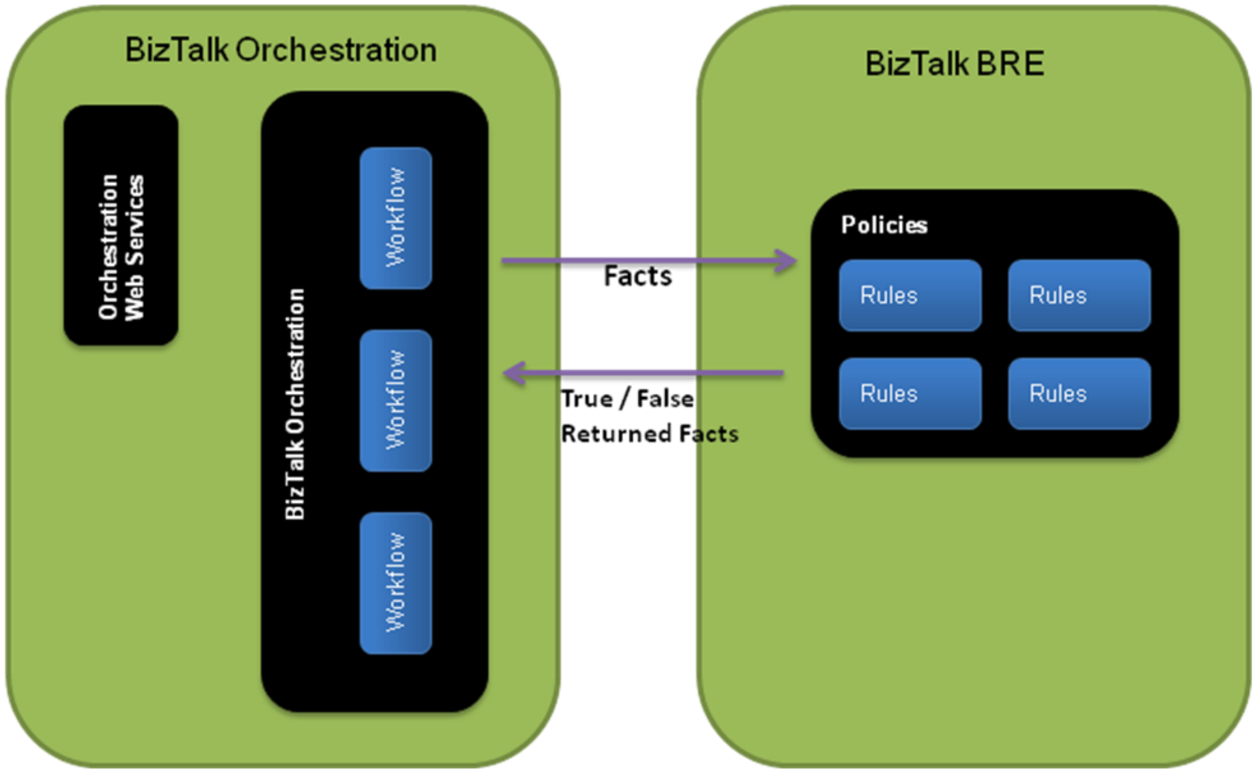


Figure 16: Integrating with the BizTalk Business Rules Engine

The process shown in Figure 16, above, is managed through a common set of highly-integrated tooling. The Orchestration Designer, together with the BizTalk Editor and the BizTalk Mapper, provide an effective way to define a business process and the rules it uses. It can be useful, however, 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, business-oriented users can create and modify sets of business rules using a tool called the Business Rule Composer, as shown in Table 4.

Business Challenge	BRE Case
Loan Origination has complex rules based on a specific workflow.	BizTalk has an open and extensible way of creating orchestrations with shapes and linked to specific business rules implementations.
Majority of Rules and Calculations don't change often. However invocation, order and parameters of those rules change.	Generic or reusable business rules can be created very easily on the BizTalk platform. There is clear separation of the Orchestration and Business Rule IDE's. Developers or Business Analysts can modify rules in a WYSWIG manner.
Loan systems have many business channels with varying applications accessing them.	With BizTalk 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 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 cumbersome and large development effort. BizTalk can manage internal and external orchestrations in a short 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 DDA, CRM, or Marketing and external providers, such as credit or mortgage insurance.	Integration is BizTalk's strong suit. With an extensive adapter framework and message bus architecture, integration with disparate systems has never been this easy. Relevant adapters: include: • MISMO (reference implementation) • IFX • SWIFT • 3270 emulation

Table 4: The BRE case

This implementation of the BizTalk BRE is useful is 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 sync with several business process 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 request from third party services. Expressing all of these rules as conditional statements using, say, an orchestration's Decide shape might be possible--but would be fairly complex to implement. For rule-intensive processes like these, the BRE is the best design choice.

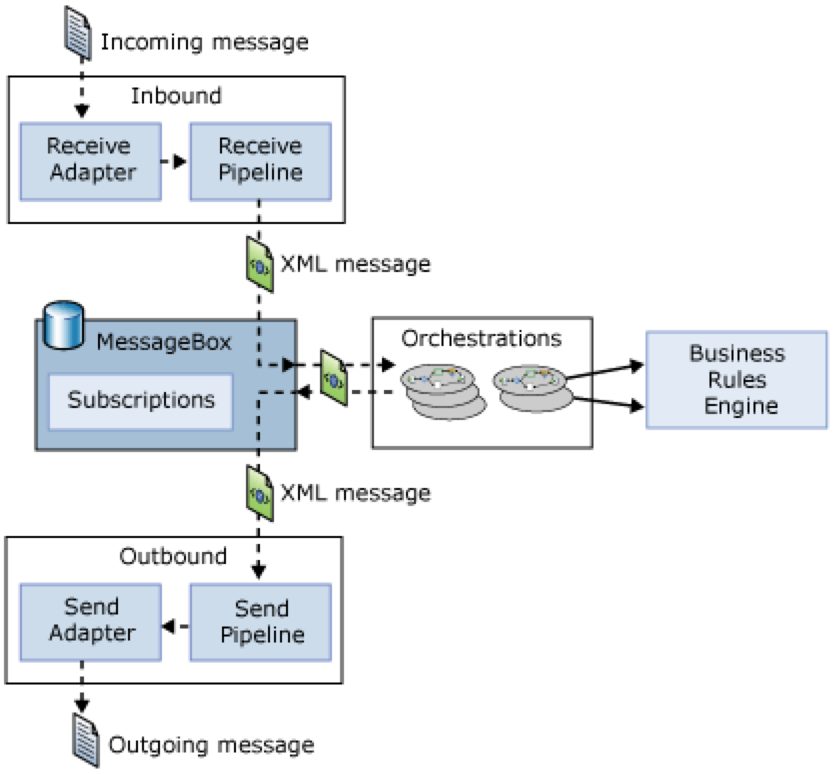


Figure 17: BizTalk BRE message flow

As Figure 17 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 things such 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 called 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. These orchestrations are not created by writing code in a language such as C#, however. Instead, a business analyst or (more likely) a developer uses an appropriate tool to graphically organize a defined group of shapes to express conditions, loops, and other behavior. Orchestrations can optionally 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 may 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. A business analyst, for example, 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 has defined this process, a developer can create a BizTalk application that implements it. This includes things such 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.

Architecture Case for BRE in Loan Origination:

- 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.
- No need of duplication of calculations or business rules, these can be invoked based on a specific workflow.
- Scalable infrastructure for rules and orchestrations.
- Regulatory issues like separation of duties is more manageable.

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 MOSS.
- **SQL Analytical Services** will be used to expose Web Parts in MOSS.

Enabling Scenarios

Now that the architecture has been 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, Excel's 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 Excel files can be difficult to manage. We want to keep this data on the server to keep the lender complaint with data retention regulations.
- **Auditablity** – In many cases data is difficult to audit when there are many users manipulating data in client-side tools. With this solution the Excel files can reside on the 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** – Excel is the client for our solution because: business users want to have familiar tooling, business owners do not want to pay for re-training, developers do not want to create redundant tools, and architects want to reduce complexity and cost.

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

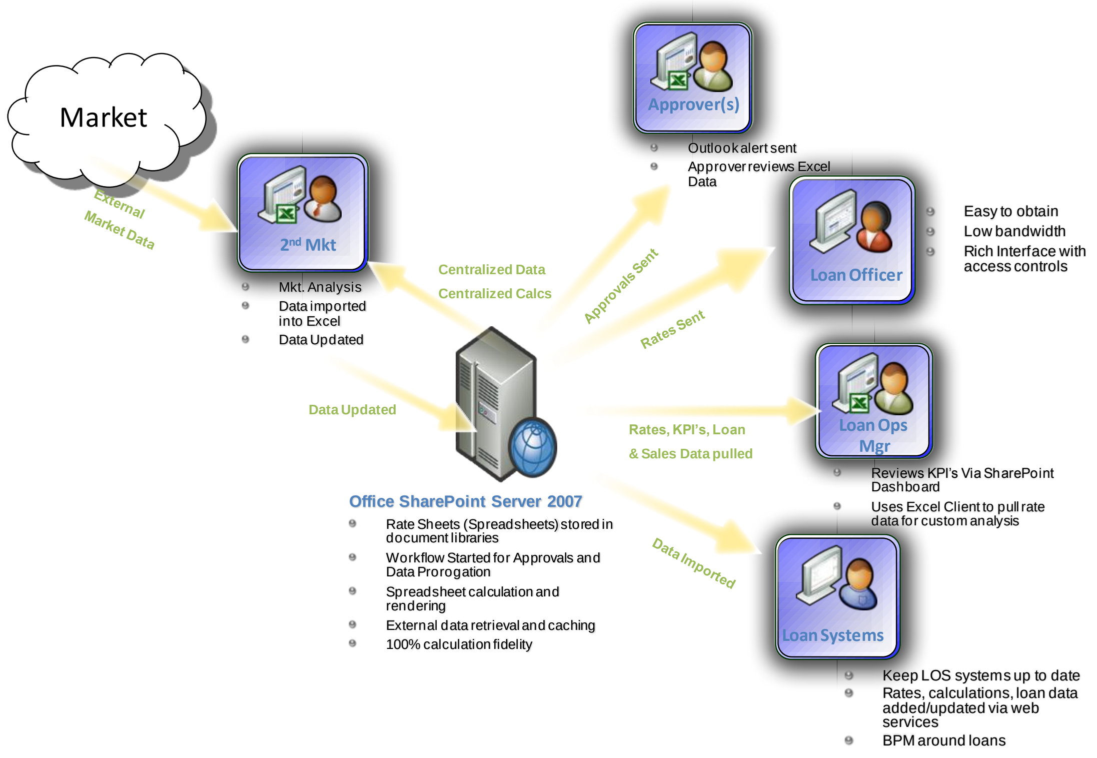

Market
External Market Data
2nd Mkt
Mkt. Analysis
Data imported into Excel
Data Updated
Centralized Data
Centralized Calcs
Data Updated
Office SharePoint Server 2007
Rate Sheets (Spreadsheets) stored in document libraries
Workflow Started for Approvals and Data Prorogation
Spreadsheet calculation and rendering
External data retrieval and caching
100% calculation fidelity
Approvals Sent
Approver(s)
Outlook alert sent
Approver reviews Excel Data
Rates Sent
Loan Officer
Easy to obtain
Low bandwidth
Rich Interface with access controls
Rates, KPI's, Loan & Sales Data pulled
Loan Ops Mgr
Reviews KPI's Via SharePoint Dashboard
Uses Excel Client to pull rate data for custom analysis
Data Imported
Loan Systems
Keep LOS systems up to date
Rates, calculations, loan data added/updated via web services
BPM around loans

Figure 18: Product and pricing technical overview

Shown in Figure 18above is the overview of the technical implementation of the scenario. The process exposes the following capabilities.

- The secondary marketing analyst starts his/her day by opening a standard 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.
- The secondary marketing analyst reviews feeds and market data and puts it into Excel.
- Manipulations are performed on the data, and the document is saved to the document library.
- The workflow is then kicked off. The appropriate approver is sent an e-mail alert to review and approve.
- The manager approves the rates.
- Workflow persists product and rate data to the LOS SQL Server.
- Web site blackout period ends and registration can now continue.
- The workflow then starts the distribution process:
  - Data is transformed and sent to disparate LOB systems
  - E-mail alerts are sent to appropriate lender staff (processors).
  - E-mail alerts are sent to brokers with a link to the current rates.
  - Rates can now be used by the application.

Loan Registration

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

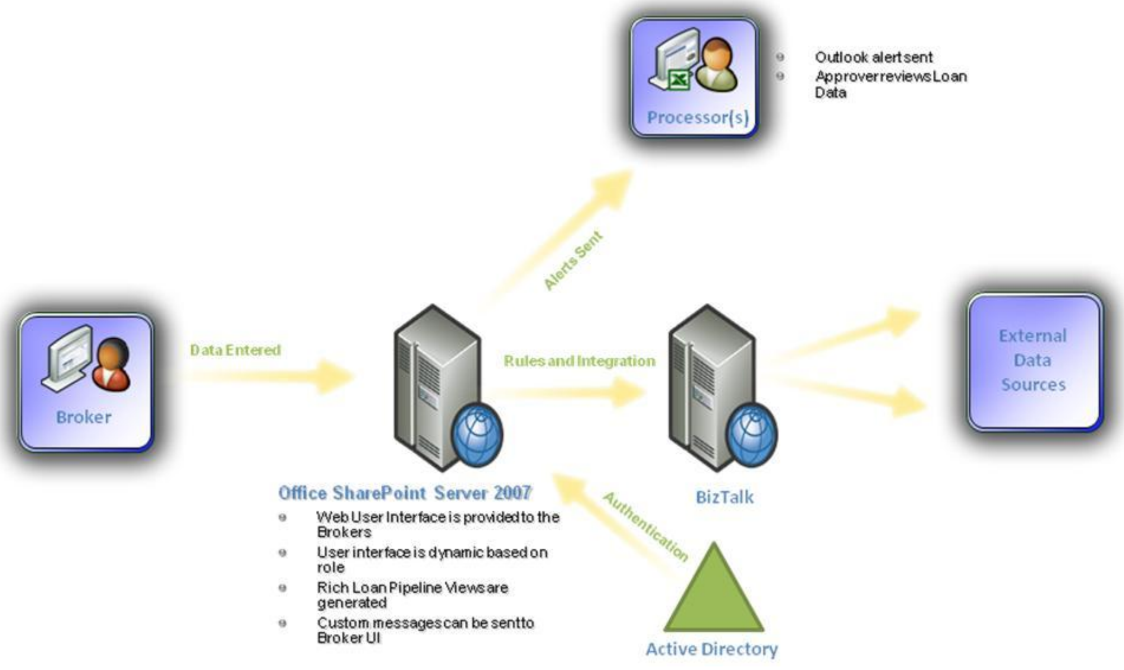

Processor(s)
Outlook alert sent
Approver reviews Loan Data
Alerts Sent
Broker
Data Entered
Rules and Integration
External Data Sources
Office SharePoint Server 2007
Web User Interface is provided to the Brokers
User interface is dynamic based on role
Rich Loan Pipeline Views are generated
Custom messages can be sent to Broker UI
BizTalk
Authentication
Active Directory

Figure 19: Registration technical overview

Shown in Figure 19 above is the overview of the technical implementation of the scenario. The process exposes the following capabilities.

- Broker comes to lender portal and authenticates.
- The UI is dynamically generated based on broker's role.
- The broker enters in the customer's information into the registration pages.
  - Alternatively the broker can choose to send a MISMO compliant XML feed or standard Fannie Mae files to BizTalk to reduce the duplication of data entry.

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.

Chapter 6

Retail Industry OBAs

Retailers are faced with the enormous challenges of globalization, regulation, growing costs, and demanding customers. In addition to these changing market conditions, there are now multiple channels to reach customers. All of this increases the need for flexibility and agility in the business applications. However, at a time when competition is keenly interested in buzzwords like agility and adaptability, businesses are confronted with systems seemingly built of steel and cement. Businesses are stuck with applications which were developed as though things would never change. These applications become more expensive and cumbersome each time organizations tweak them to fit new business imperatives. Unfortunately, this leads to less materialization of automation because systems require constant human intervention and IT development. All too often, the alignment of technology with business objectives gets bogged down by integration problems and progresses no further than the white board.

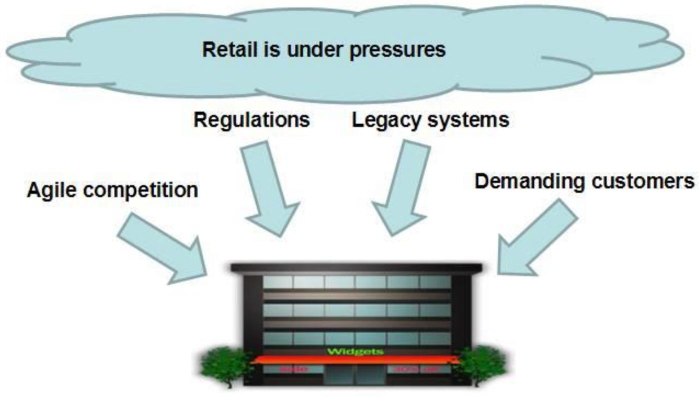


Figure 1: Retailers under pressure from different angles

Figure 1 also shows another form of pressure – embracing emerging technologies to enhance the overall experience of the customers and improve efficiency within the organization. These pressures are forcing retailers to rethink their overall strategies. Current legacy systems – either in store or at the corporate headquarters – were designed and implemented many years back, and retailing has moved on since then. These legacy solutions tend to be uni-functional. For example check-out systems remain single-purpose devices, and even though new point-of-service (POS) solutions offer inventory management, price optimization, and customer management some retailers are still only using POS devices to check-out the customer.

In addition, current retail solutions are built as separate siloed systems. Brick and mortar retail channel runs independently from the eCommerce channel, which runs independently of the catalog base order management system, and so on. However, customers are increasingly using the stores to touch and feel the products but ordering the products from eCommerce sites. As broadband availability becomes pervasive, the distinction between different channels is blurring. Customers are expecting the same experience irrespective of how they shop. They expect their preferences and customer information to be carried between the channels and expect to accrue loyalty points across all channels.

Current retail solutions pose many challenges in providing a seamless customer experience across multiple channels and providing information where it is needed. Some of the most common challenges are:

- Lack of applications that can pull together the information gleaned from stores and other channels.
- Heterogeneous systems built using different platforms.
- Silo-ed applications with minimal or no integration between them.
- Lack of real-time business intelligence software that provides real-time KPI indicators for decision making.
- Tightly-coupled point-to-point integration.
- Duplicate customer data from each channel maintaining its own customer data.
- Lack of product inventory visibility across channels.

So what is the solution? SOA is leading the way to help in this area. However, the challenge is not just limited to integration of different applications in a loosely-coupled fashion, it goes beyond that. It is about providing the right information at the right time and at the right place, based on the role of the owner and context of the task. In addition, it is about providing this information in a format that workers are already used to. For example, it means getting sales information in near real time to the store managers, regional managers, and merchandising managers in the corporate headquarters. Another area is the availability of product information to the sales agents.

Making information available in near real–time requires systems that can generate near real time data, transmitting the data in near real–time to the right place, and building applications that can consume data in near real time.

Providing Realtime Information

A typical retail enterprise has many moving parts that keep the enterprise running. These different systems are housed and run in different parts of the organization. For example, in the retailer's enterprise headquarters, there are systems such as supply chain management, store planning, warehouse management, CRM, ERP, HR, and so forth. These different applications are built using technology ranging from DOS, to mainframe, to proprietary platforms. For the enterprise to function smoothly, these disparate systems need to work together to produce and consume messages. Today this is accomplished through tightly-coupled integration between these applications. This prevents the enterprise from plugging in new applications without significant investment in yet another tightly-coupled solution. Slowly, but steadily, enterprises are realizing the benefits of SOA and turning to it so that they can replace their legacy applications in an incremental fashion without having to rip and replace all the systems.

With service orientation, one of the key issues to address is integration of the disparate systems in a loosely-coupled fashion. This helps in getting the data in real time to the right application. However, this does not address the problem of what happens when the data is received, how best to consume it, who can consume it, and so on. These aspects are left open and enterprises are looking for a solution to deal with it. One way is to build custom user interfaces where data from different applications is brought together to provide the functionality required by the user. However, this requires building custom applications from the ground up; training the users; and huge maintenance costs as every time a new functionality is required. It demands custom work. Whereas, if the most familiar application to the user, such as Office, is used as the interface where data from different applications, systems, and data sources is brought together for presentation, then it reduces the cost of training and reduces the maintenance expense as well.

As explained in the beginning of this book, Office 2007 platform comes with capabilities to build custom applications which provide the relevant information within the most familiar interfaces. It provides the Open XML file format, extensible UI, the Business Data Catalog (BDC), business intelligence capabilities, and communication and collaboration through WF. These standard capabilities of the office 2007 platform help in assembling what is described as *composite applications*. Different tiers of composite applications were described earlier, most critical of which is the presentation tier. Traditionally, there was little to no focus on this important tier. The presentation tier is where the end user interacts with the applications, and if this tier is not well thought out – no matter how stable, scalable, and reliable the backend application may be – it does not enhance productivity of the end user.

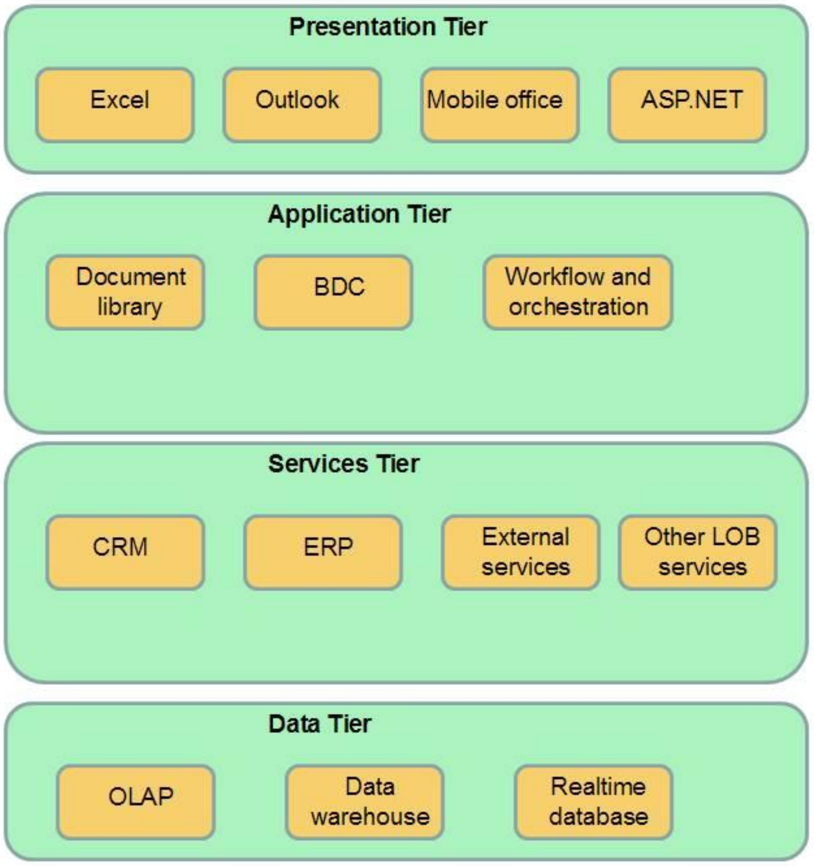


Figure 2: Retail application tiers

Composite applications are great way to surface information out of service-oriented-architecture. Having different LOB applications such as inventory management, replenishment, sales information, and so on exposed as services makes it easier to build support for cross-functional processes into a composite application. This ability to surface data from LOB applications helps in providing the right data to the right user at the right time at the right place. This is also based on the context and user privileges. For example, information can be either hidden or exposed based on context such as user credentials, scheduled tasks, benefits information (HR), and training information (which training they have completed and which they have not). This helps in improving overall efficiency and productivity at a retailer.

Retail Composite Applications

Providing a compelling user experience depends not only on service delivery and flexible workflows, but also on the ability to consume services and interact with the workflows in a rich meaningful way. Service consumption needs to be contextual, mapping to the natural workflow of employees and customers. There are, of course, some traditional ways to provide this which span from smart clients to Web applications. However, providing an experience to the information worker which is based on familiar tools makes a huge difference. Also, a technology which can support both offline and online scenarios is a great addition.

Enterprise Collaboration

In retail, enterprise collaboration between stores and corporate headquarters plays a critical role. Current collaboration tools range from snail mail, phone, and fax to e-mail. Some of these tools are good for unstructured communication. However, for structured collaboration these tools fall short. So what ends up happening is the structured collaboration is accomplished pretty much manually. Figure 3, below shows the some of the manual collaboration tools.

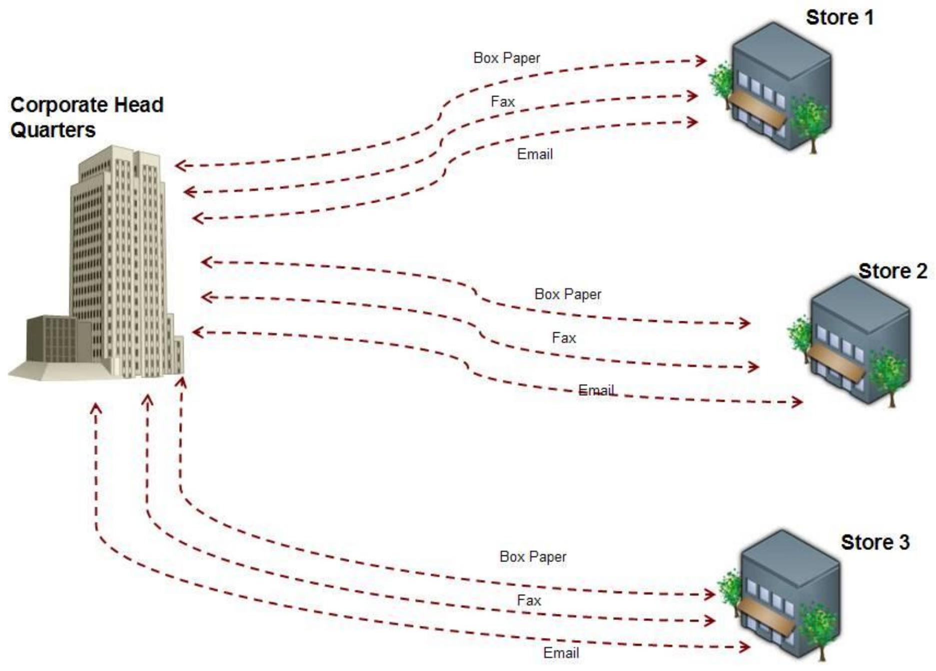


Figure 3: Current collaboration tools

OBA with its powerful component capabilities can help in capturing and automating the collaboration steps and process. For example the new office client applications come with customizable containers that can easily surface information and functionality from LOB applications through custom tasks, ribbons, and activities that present users with data and actions in the context of their current activity. Microsoft Office SharePoint Services (MOSS) comes with many key components to enhance overall productivity in the collaboration area. The workflow (WF) engine of the 2007 Microsoft Office System helps greatly in managing the processes of collaboration. When a promotion is planned, the category manager or marketing manager creates the promotion and associated document libraries. As shown in Figure4, below. The store manager then associates the library with a particular workflow which triggers actions for the store managers. These actions are to review the promotions and provide feedback. Currently, this feedback is collected via e-mail, fax, phone, or other forms of ad hoc tools. However, with the 2007 Microsoft Office System, the store manager receives the promotion details in the Office client with custom ribbons and panes based on the context of the work. Updates are made to the promotion and once the document is resubmitted, it then triggers an action for the marketing manager, or others based on the workflows.

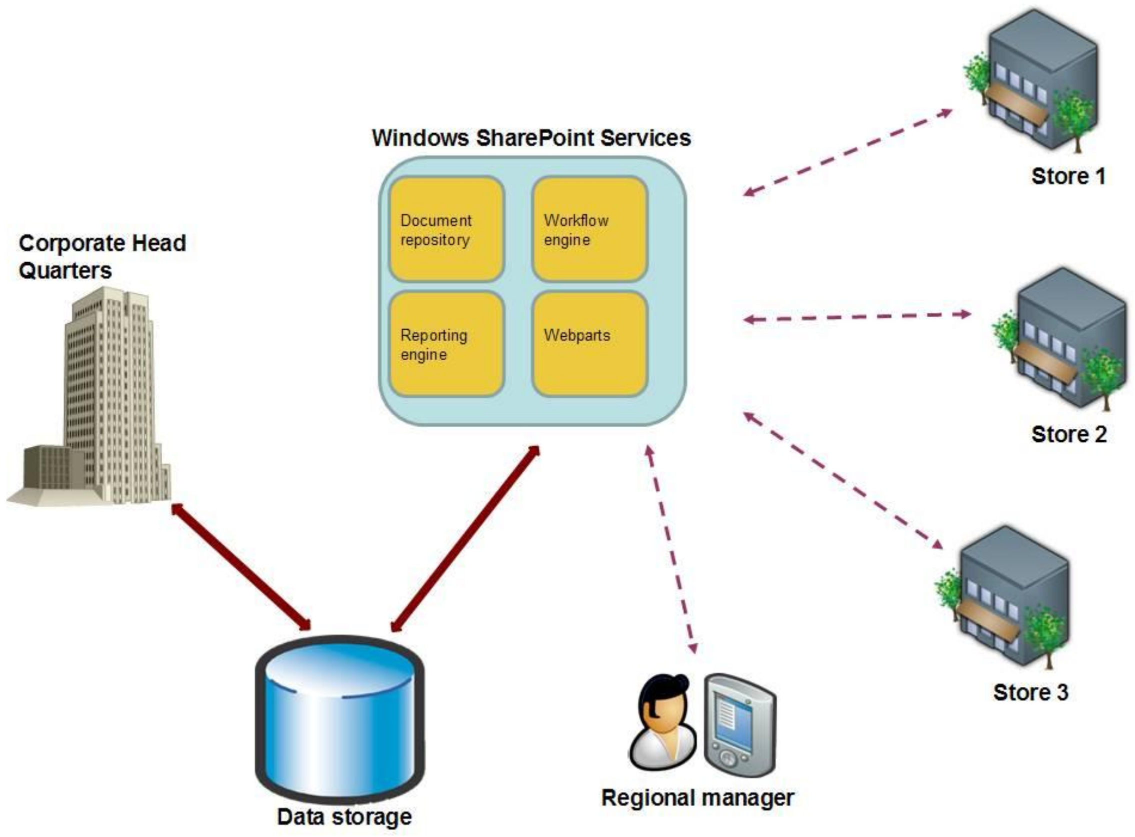


Figure4: Collaboration with Microsoft Office SharePoint Services

This same information can also be surfaced using 2007 Microsoft Office System Web Parts in a store manager dashboard. Figure 5, below, shows some examples of Web Parts in a retail enterprise.

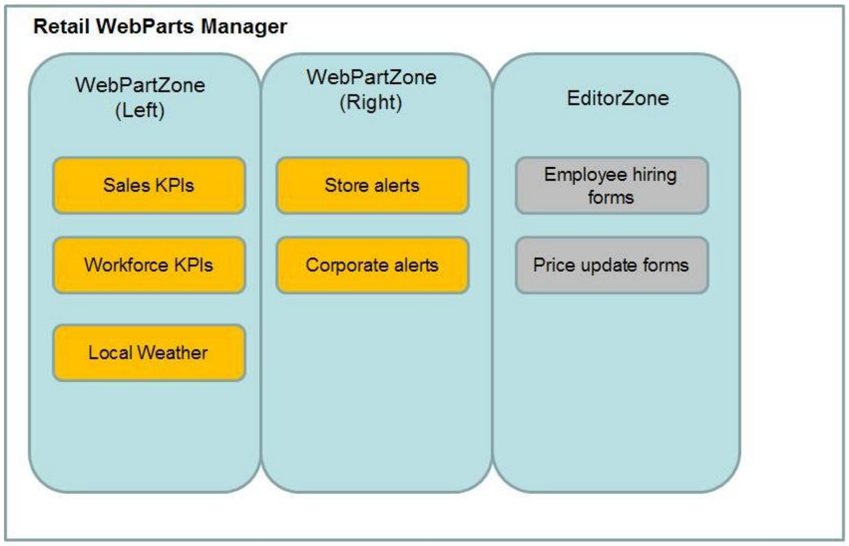


Figure 5: Examples of some Web Parts in retail

Collaboration between corporate headquarters and the store happens for promotion management. Typically the promotion is created by the marketing manager in Excel, then a document library is created, and the promotion document is associated with a workflow. SharePoint services triggers the workflow which creates tasks for all the store managers where the promotion will be offered. All the store managers get alerted about the new task. Store managers can get to the task of reviewing the promotion and providing feedback based on the availability of time and their own schedule. SharePoint services offers complete version control of the document to track the changes made by the store managers.

The managers review the promotion, update it, and save it to the SharePoint server. This triggers the workflow to alert the marketing manager to review and accept or reject the changes selectively as made by the store managers. As a result, the marketing manager completes the promotion by collaborating with the store managers much more quickly and without losing information. Store managers get the flexibility to review the promotion in their free time as opposed to being forced into a conference call or fit into someone else's schedule. This improves overall productivity of the manager and efficient use of his/her time.

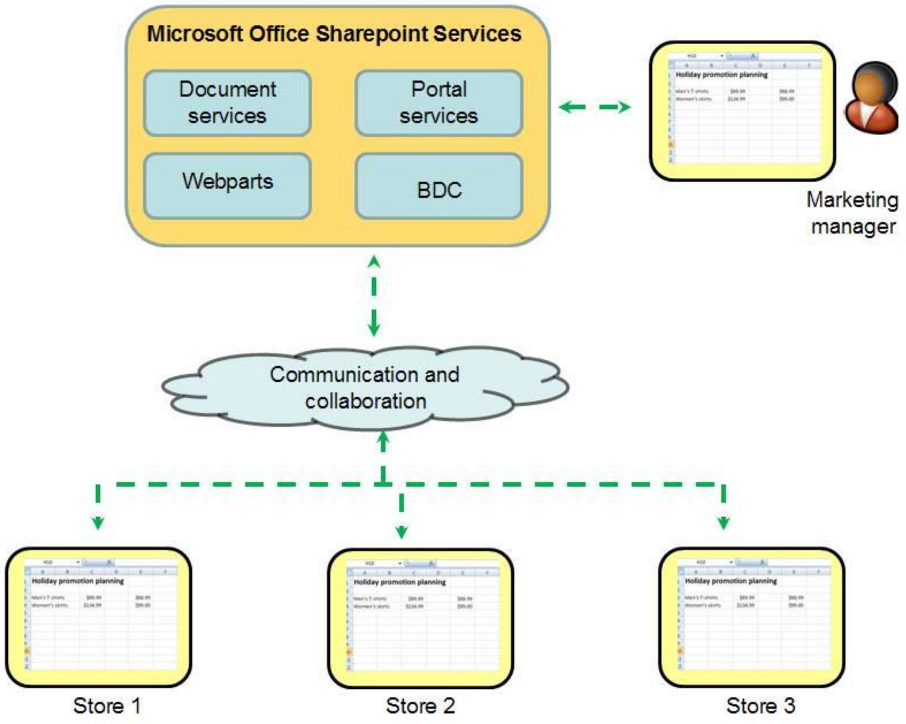

Microsoft Office Sharepoint Services
Document services
Portal services
Webparts
BDC
Holiday promotion planning
Marketing manager
Communication and collaboration
Store 1
Store 2
Store 3

Figure 6: Collaboration using MOSS

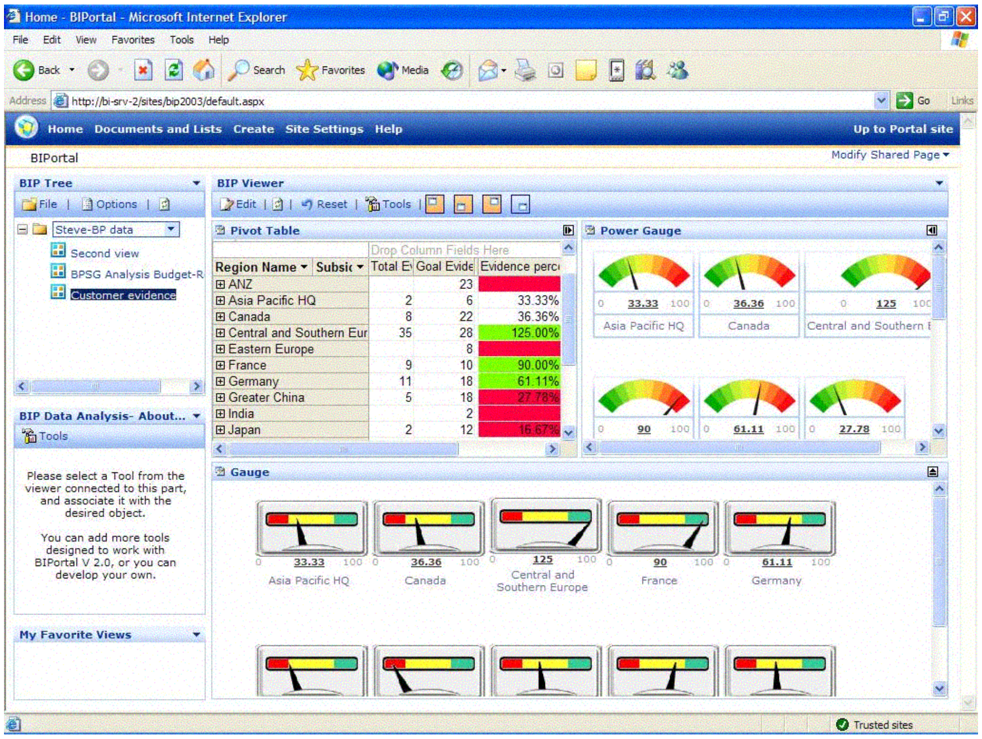
As shown in the above figure, this architecture enhances the overall collaboration between the store and the corporate office. This also automates the process of collaboration where all the steps of the process are captured by the SharePoint server. Documents are managed and workflows ensure the proper process is followed. In absence of this automated collaboration, the same process would usually require many e-mails or faxes. In other words, a frustrating and very inefficient way to collaborate!

Above we took one example of collaboration to showcase the capabilities of MOSS. However, in retail there are many scenarios where collaboration is needed between stores and the corporate office. Labor scheduling, training, category management, and so on are other areas where collaboration is very critical for the overall operation of the retail enterprise.

Retail dashboards

Dashboards in the retail space are becoming a norm. Many retailers present everything employees need directly on their dashboards, whether they need realtime stock levels, price lookups, merchandizing information, or online training and HR profile updates. For example, a store manager prefers a dashboard that is custom-built for the store manager's duties. Similarly, others such as the regional manager, store employees, and corporate employees all need their own custom dashboards which make it easy for them to perform their job. These custom dashboards can be from the built ground-up as a custom application. However, they will not be flexible enough and will require huge maintenance when information changes. In comparison, dashboards built by assembling Web Parts are easy to build and change.

There are typically many LOB applications in an enterprise such as merchandising applications, ERP, HR, and so on. Surfacing appropriate information from these applications is very much required by the manager and employees based on their role. 2007 Microsoft Office System comes with the (BDC) that runs within the server as a shared service. It can read data from multiple types of data sources – databases, analysis services, cubes, and Web services – and then surface this back into the portal through SharePoint tables and lists.



Region Name	Subsic	Total E	Goal Evide	Evidence perc
⊞ ANZ			23	🟥
⊞ Asia Pacific HQ		2	6	33.33%
⊞ Canada		8	22	36.36%
⊞ Central and Southern Eur		35	28	125.00%
⊞ Eastern Europe			8	🟥
⊞ France		9	10	90.00%
⊞ Germany		11	18	61.11%
⊞ Greater China		5	18	27.78%
⊞ India			2	🟥
⊞ Japan		2	12	16.67%

Please select a Tool from the viewer connected to this part, and associate it with the desired object.

You can add more tools designed to work with BIPortal V 2.0, or you can develop your own.

Figure 7: Example store manager dashboard

As the above figure shows, a rich store manager dashboard helps the manager keep a close eye on the KPIs in real time and make decisions based on the store performance. This rich user experience for the manager improves overall productivity and improves efficiency in the store. Dashboards are important for store managers to get a quick understanding of the overall store operation. They provide the critical KPI information at a glance, alerts from the suppliers, manufacturers, or from the corporate office. They provide workforce utilization information, overall schedule information, tasks assigned to the store, and their status. Figure 8, below, shows an example of a dashboard that is built as a composite application, where information from different services is surfaced.

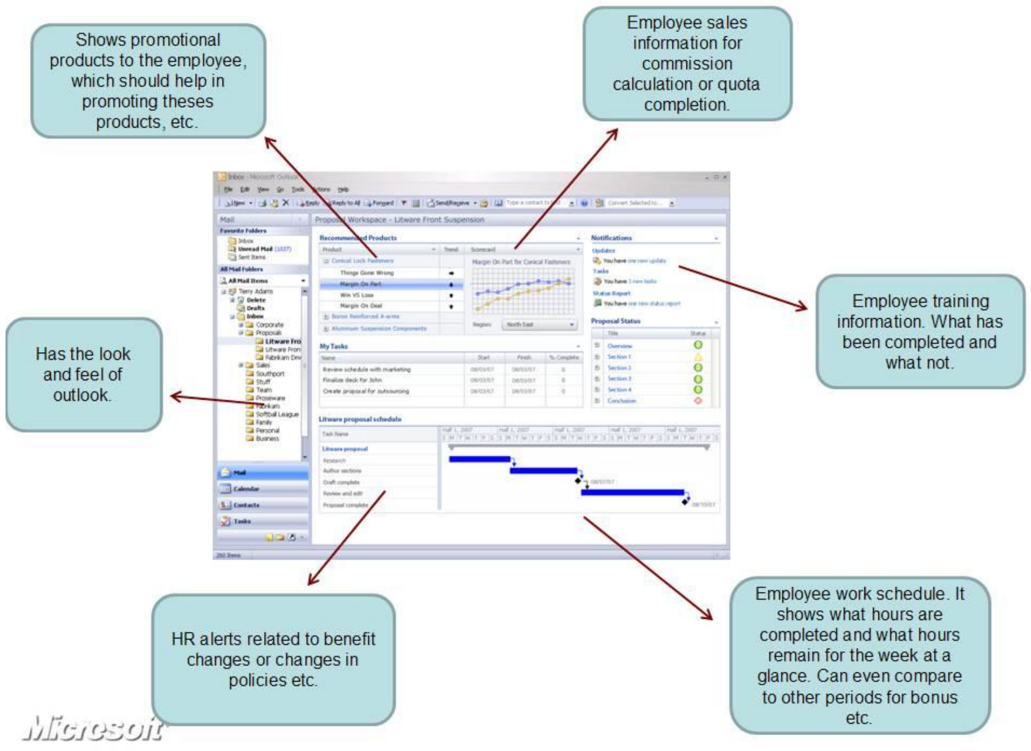

Shows promotional products to the employee, which should help in promoting theses products, etc.
Employee sales information for commission calculation or quota completion.
Employee training information. What has been completed and what not.
Has the look and feel of outlook.
HR alerts related to benefit changes or changes in policies etc.
Employee work schedule. It shows what hours are completed and what hours remain for the week at a glance. Can even compare to other periods for bonus etc.
Microsoft

Figure 8: Store manager dashboard assembled using different services

Similarly, the corporate office needs dashboards to get an understanding of the overall business at a glance. Corporate teams need quick views into the overall KPIs, alerts about hot issues, promotions, employee turn-over, and training. Along the same lines, employees need their own mini dashboards where they can quickly view their schedule, training information, HR information, and so forth.

MOSS provides the key capabilities to build these dashboards quickly. As explained in previous sections, MOSS comes with a Web Part gallery that contains a rich set of Web Parts out of the box, for example, to surface Office Excel spreadsheets and charts, and to view lists and tables. Solution providers such as the merchandising application or inventory management application can also provide their own custom Web Parts for application-specific or domain-specific functionality, which can then be uploaded into MOSS. Information workers can then personalize pages, by adding Web Parts from the gallery, removing Web Parts from a page, or rearranging the layout.

The 2007 Microsoft Office System BDC runs within the server as a shared service. Since it can read data from multiple types of data sources, such as databases, analysis services cubes, and Web services and then surface this back into the portal through SharePoint tables and lists, it is ideal for building customer dashboards in retail. So data from the business analytics cubes can feed into the KPIs Web Parts. For example, these feeds from external services can enable alerts or weather reports for the dashboards. Since MOSS can read data from Web services it opens up huge opportunities in retail to consume data from external entities and corporate systems.

For example, internal Web services can provide information about customer returns, operator voids, or fraud in the store. These Web services are consumed by the BDC to update the dashboard for the manager to quickly look at the information and react to it. When a manager is moving around, Microsoft Office for Windows Mobile can be used to react to any alerts and update any information.

In building the store manager dashboards, one of the key elements is aggregating real-time data which then is manipulated to create the KPIs. Typically in retail, real-time data availability is a rarity. Only few top retailers have achieved what is known as the "trickle feed." However, it is critical to get this data to generate KPIs which are relevant to the managers. Figure 9, below, shows the architecture for aggregating real-time sales information from the stores. The store service captures the sales information from the POS terminals, aggregates it, optimizes it, ,and transforms it to a standard format (potentially IXRetail format) and then transfers it to the corporate systems in real time.

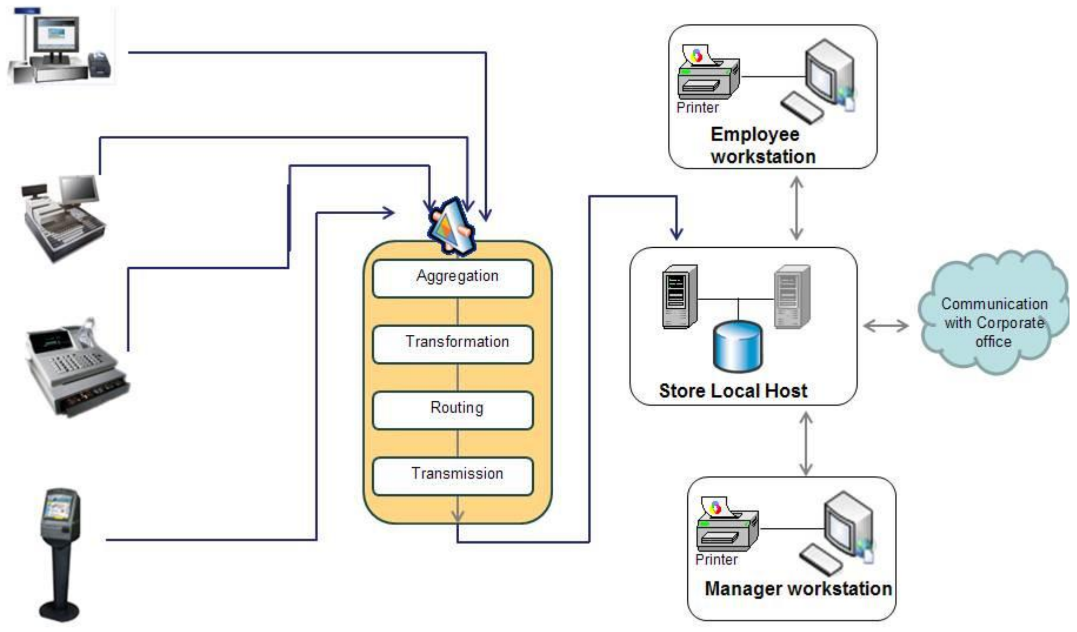


Figure9: Aggregation and transfer of data from store to the corporate systems

Corporate services can be built as a BizTalk service which aggregates data from different stores, transforms the data if need be, and then transfers it to all the services that are interested in consuming the data. For example, an inventory solution is interested in the sales information to keep track of the inventory levels at the stores, which help in creating new orders for products falling below a certain threshold. A CRM application is interested in the customer order information to track customer loyalty and for creating targeted promotions on products and services. Other LOB applications may be interested in the data for many other purposes. For all these LOB applications real-time data helps in ensuring right the decisions are taken at the right time. Figure 10, below, shows this process of data aggregation at the corporate headquarters.

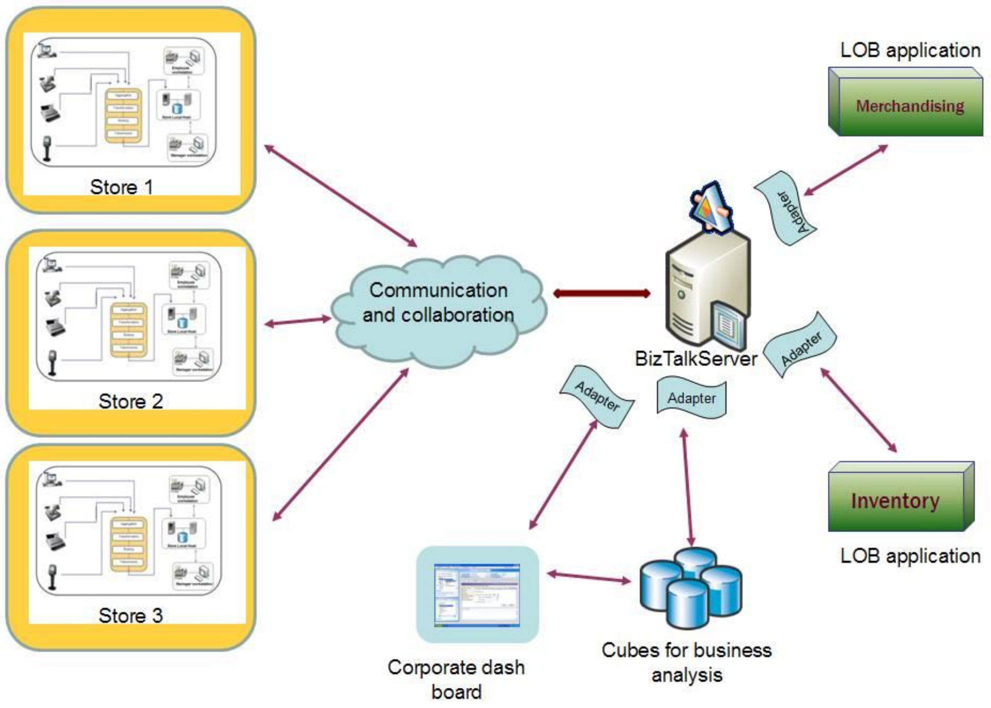


Fig 10: Transformation and orchestration of data at the corporate head quarters

BizTalk also sends the data to the data warehouse application for creating cubes which drive the business analytics piece. Here the BDC helps in reading the analytics information from the cubes to update the KPIs in real time. Another key functionality is the integration with external entities. For example, real time collaboration with suppliers can be accomplished through BizTalk using the BizTalk adapters. This information is surfaced into an Office client for review and update. Figure 11, below, shows how data from many external and internal services is surfaced using BDC to create the dashboard.

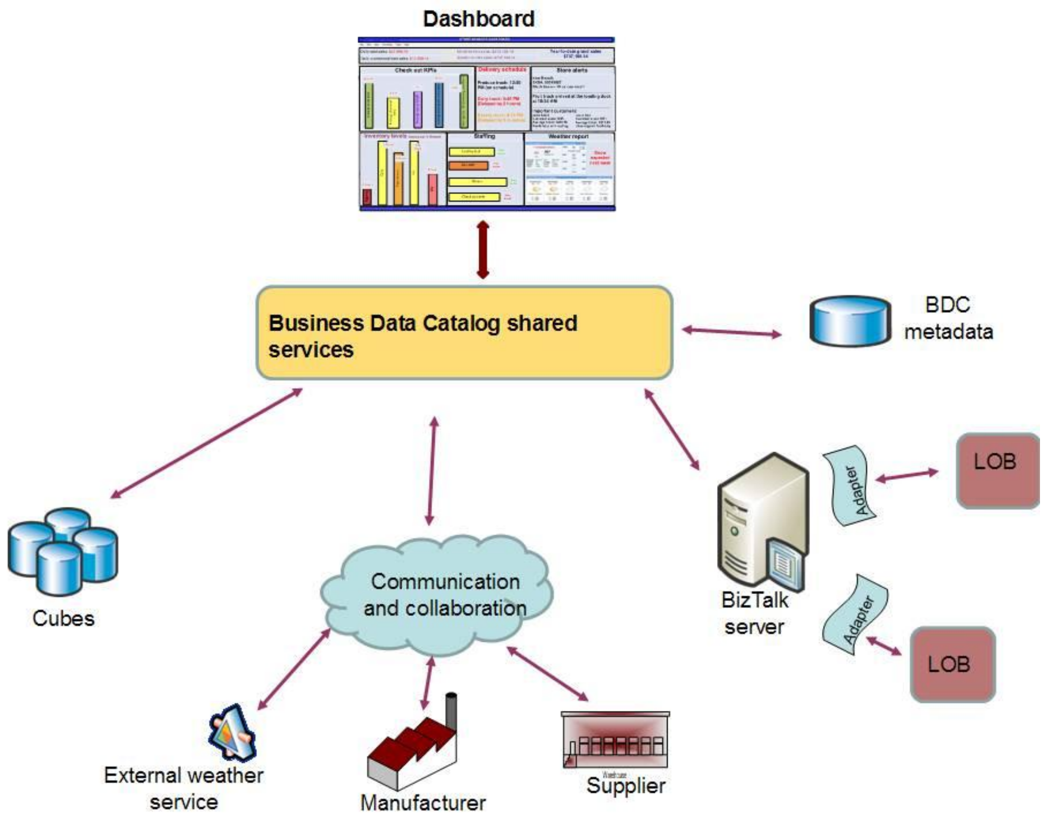


Figure 11: Assembling of custom dashboards using Microsoft Office SharePoint Services

Conclusion

2007 Microsoft Office System has evolved into an application platform. It comes with many capabilities that are critical in building applications and specifically in building composite applications. This helps in custom assembling applications for different users as opposed to fitting the user's needs into existing applications.