

A Prescriptive Architecture for Electronic Commerce and Digital Marketing

Authors:

Javed Sikander
Microsoft® Corporation

Vinod Sarma
MindTree Ltd.

Published: March 2010

Version: Version 2.0

Abstract

This white paper provides a prescriptive architecture for e-commerce and digital marketing systems. It introduces the concept of an e-commerce *architecture frame* that identifies the key decisions you need to make and the tradeoffs you need to consider when developing reliable and scalable systems. It lists the key technical capabilities that are needed to develop these systems. Microsoft Commerce Server is the core component of the Microsoft platform for e-commerce and digital marketing. This paper recommends a methodology and provides best practices for developing solutions with Commerce Server.

Microsoft®

Copyright Information

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

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

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

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

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

© 2010 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, BizTalk, Dynamics, SharePoint, SilverLight, SQL Server, Visual Studio, Windows, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Microsoft®

Contents

Introduction	7
E-commerce 1.0 – Emergence of the Transactional Web	8
E-commerce 2.0 – Evolution of a Participatory Web	8
E-commerce and Digital Marketing	9
Technology Enablers for E-commerce and Digital Marketing	9
Business Intelligence and Data Mining	9
User Experience	9
Mobility	10
Search	10
Content Management	10
Web Services	10
Virtualization and Cloud Computing	10
Advertising	11
Instant Messaging	11
Payments	11
Prescriptive Architecture Fundamentals	11
Design Fundamentals	12
E-commerce Architecture Frame	13
Authentication and Authorization	14
Caching	18
Communication	19
Technical Capabilities for E-commerce and Digital Marketing Solutions	21
Content Management, Publishing, and Delivery	21
Rich Web and Client Presentation Capabilities	22
Stable Core Web Framework	22
Data Repository	22
Commerce Engine	22
Integration Framework	23
Analytics	23
Scalable Application Server Platform	24
Search	24
Campaign Management, Delivery, and Tracking	25

Microsoft Platform for E-commerce and Digital Marketing	25
.NET Framework 3.5	26
Windows Operating System	27
SQL Server.....	27
Microsoft Commerce Server.....	28
Microsoft BizTalk Server	29
ASP.NET.....	29
SharePoint Server	29
Bing	30
FAST	31
Search Server	32
Windows Presentation Foundation (WPF) and Silverlight	32
Atlas	33
Active Directory	33
Microsoft Dynamics AX.....	33
Architecting Digital Marketing and E-Commerce Systems with Microsoft Commerce Server 2009 34	
A Methodology for Architecting a Commerce Server 2009–based System.....	34
Scenarios.....	34
User Stories.....	35
Identifying Non-Functional Requirements	35
Product Feature Alignment	35
Identifying Required and Appropriate Customizations	37
Identifying Design Patterns for Customizations	37
Identifying the Integration Points.....	38
Identifying Additional Products and Features	38
Deployment Architecture and Capacity Planning.....	39
System Architecture	41
Reference Implementation	42
System Architecture – Conceptual View	42
Content Management and Delivery	42
Multiple Channel Interfaces	43
Commerce Engine.....	43
Business Management Applications.....	43
Integration Middleware.....	43

Data Warehouse	44
Data Analytics	44
Web Analytics	44
Search	44
Back-end Systems	44
System Architecture – Logical View	45
Consumer Store Front	45
Campaigns.....	51
Integration	53
Admin Site.....	53
Catalog Management	54
Order Management	55
Marketing Management.....	55
Customer Management.....	55
Schema and Master Management	55
Analytics and Reporting Architecture.....	56
Back-end Integration Architecture	57
System Architecture – Development View	60
System Architecture – Deployment View	63
Common Trades-offs	66
Designing Commerce Server 2009 Systems	66
System Design	68
Design Template – Commerce Server 2009 Full Framework Solution	68
Design Template – Commerce Server 2009 with SharePoint Commerce Services on Windows SharePoint Services	70
Design Template – Commerce Server 2009 with an ASP.NET Front End	73
Design Template – Commerce Server 2009 Solution that Extends SharePoint Commerce Services	75
Development Best Practices for Digital Marketing and E-Commerce Systems Using Commerce Server 2009	77
Solution Structure	77
ASP.NET Web Parts and Web Pages	77
.NET Components (Business Logic, Data Access).....	78
Office SharePoint Services Web Parts	79
Office SharePoint Services UI Elements	79
BizTalk-Specific .NET Components	79

WCF Services.....	80
Build Setup	80
Performance Benchmarking and Verification	81
Site Structure.....	81
Invoking Foundation Services.....	82
Integration Interfaces.....	82
Security.....	83
Threat Modeling	83
Encryption.....	83
Authentication and Authorization	84
Secure Page Access.....	84
Estimating Commerce Server Development Efforts	84
Extending Commerce Server and Office SharePoint Server Web Parts.....	86
Dependencies	86
Customizing the Product Details Web Part	86
Extending the Product Provider Web Part	87
Extending the Product Query Web Part	88
Extending the Order History Web Part.....	88
Commerce Server Configuration File Settings	89
Search Integration	93
Commerce Server Adapters	94
Sending Data from Commerce Server	94
Sending Data to Commerce Server.....	95
Data Processing by the Commerce Server Adapters.....	95

Introduction

Electronic commerce, or *e-commerce*, refers to the purchase and sale of goods and services over the Internet. Fundamentally, e-commerce is about the people, process, and technology involved in allowing a consumer or business to purchase goods or services from another business or individual. For centuries, traditional commerce has involved physical brick and mortar businesses, stores, shopping malls, catalog sales, and so on. In the last hundred years, other channels for commerce, such as telephone and television sales were established. With the growth and widespread availability of the Internet in the 1990s, a sizeable commerce activity moved to the World Wide Web. Today, consumers go to their favorite e-commerce sites to not only to buy and sell, but to conduct research, review, or comment on products and services. A recent comScore presentation¹ reports that nearly 70 percent of customers consider the Internet to be an important factor in making buying decisions, and 60 percent have gone online to do research before purchasing items in a store.

The share of e-commerce sales as compared to overall retail sales has been increasing continuously, and now represents approximately 6 percent of the total US retail sales in 2009. For industries such as travel and hospitality, it is becoming the preferred transaction channel. In the future, this trend is expected to grow even more for traditional retail offerings.

Conversion rates—that is, the percentage of customers who visit an e-commerce site and actually buy something—remained stable, even during the recent challenging economic times. The conversion rate remains at 3 to 3.5 percent, as it has for the past few years. This is a clear indication that Web-based commerce has not been as adversely affected as other channels during the recent economic slowdown. Table 1 summarizes the conversion rate statistics.²

Table 1: Internet Commerce Conversion Rate

	Average	Sample size
Conversion rate	3.4%	85
Percentage of sale driven by repeat customer	28%	70
Repeat rate (percentage of customers who made a repeat purchase within 12 months)	24%	59
Marketing cost per order	\$ 14.30	68

¹ comScore, Inc., “State of the U.S. Online Retail Economy Through Q1 2009.”

² Source: Forrester Research, Inc., “The State Of Retailing Online 2009: Marketing.”

E-commerce 1.0 – Emergence of the Transactional Web

The earliest e-commerce simply exploited the reach and ubiquity of the Internet with simple and fairly standard storefronts on the World Wide Web. Retailers around the world rushed to create an online presence. Some simply published their product catalogs and information on the Web, and allowed consumers to find and choose products. However, the customers still had to visit a physical store or use the telephone to purchase. Other retailers provided order capture capabilities on the Web. Few retailers provided the end-to-end experience to research and pick products from catalogs, add them to a virtual shopping cart, choose a shipping option and select from easy payment options and checkout. New business models emerged where large online retailers provided hubs for smaller retailers and individuals to create store fronts and sell their goods and services. These business models leveraged the power of the Internet and provided capabilities to search through large product catalogs, create auctions, and track product shipments. Figure 1 illustrates the emerging e-commerce–based transactional Web.

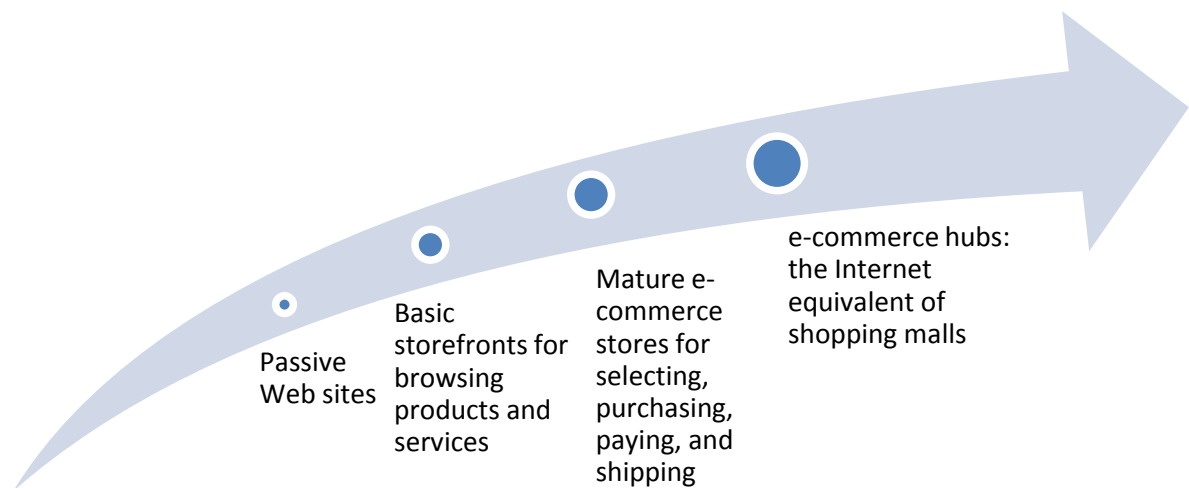


Figure 1. Growth of the transactional Web

E-commerce 2.0 – Evolution of a Participatory Web

Over time, the e-commerce function has evolved from a component of the retail strategy to a core strategic enabler for businesses. For most retailers, e-commerce represents their fastest growing revenue channel and a key asset from which they can build and maintain relationships with their customers. Retailers today are focused on delivering consistent experiences for customers across channels to enable revenue growth, brand loyalty, and innovation.

The Web itself has grown in terms of its value to businesses and consumers alike. The reach and ubiquity of the Web makes it the ideal medium for communication, collaboration, business, social exchanges, and fun and entertainment. However, in the past five years, the Web has gone through another significant transformation. It has evolved from a transactional, read-only Web to a participatory, read-write Web

The evolution of e-commerce powered by this new participatory Web is what we call *e-commerce 2.0*. Retailers have found creative ways to exploit this participatory social Web by deploying new ways to reach, understand, influence, and service their customers. In a McKinsey survey from 2007, three quarters of the executives surveyed said that they plan to maintain or increase their

investments in technology trends that encourage user collaboration, such as peer-to-peer networking, social networks, and Web services.

E-commerce and Digital Marketing

The function of marketing is to create brand awareness and position products or services in a way that excites the target users and compels them to purchase and use the products or services. As the Web transforms the way we live and work, many business functions are being altered. Marketing is one such function that is undergoing a massive change. In the new Internet-based marketplace, consumer perception of a brand is influenced not only by price, placement, and packaging in the store, but also by its presence online. This experience is not limited to the retailer/manufacturer brand Web site, but also includes social networks, blogs, user-generated audio/video sites, and instant messaging. It includes user experiences on the desktop, mobile phones, gaming consoles, and kiosks. Creating and maintaining a consistent brand experience across such digital media channels (referred to as *digital marketing*) has become a high priority for many chief marketing officers. The need for digital marketing is driven by digital commerce; therefore, marketing needs to reach and touch consumers not just in the physical world but also in the digital world.

Technology Enablers for E-commerce and Digital Marketing

This evolution of e-commerce could not have happened without significant advances and paradigm shifts in the technology landscape. The following subsections describe some of the enabling technologies for e-commerce 2.0. Subsequent sections delve into these technologies in greater detail.

Business Intelligence and Data Mining

Multi-channel retailing and the ability to consistently identify and serve customers across e-commerce sites and brick and mortar stores is of growing interest to many businesses. Identifying online customers in physical stores and vice-versa, allowing customers to purchase online and pick up from a store nearby or purchasing in stores and having the goods delivered home are all capabilities that are improving customer loyalty. With powerful data mining, retailers are now able to track what customers are buying online, suggest related products and services, and offer instant rebates. Using business intelligence capabilities, retailers can evaluate product usage and service data to design better, customer-focused products and services. Collaborating with suppliers on product design and development is helping retailers to bring products to market quickly and cost effectively.

User Experience

The user experience on e-commerce Web sites has significantly improved over the past decade. Retailers include rich media, such as audio and video, with product information for topics such as product assembly and use. This has been possible because of technologies like Adobe Flash and Microsoft Silverlight, which allow seamless integration of animation and video with capabilities for streaming and zooming. These technologies also help create an engaging experience for customers. Additionally, businesses are focusing on simple and effective site design that allows for easy access to information and commonly used functions. Further personalization of the user experience by

understanding the effectiveness of different page layouts and product placements on the page can also drive customer satisfaction and increase usage.

Mobility

With the proliferation of mobile devices, customers now have the ability to go to an e-commerce site to browse and purchase goods from anywhere, anytime. SMS is used extensively in developing countries as a medium for short marketing campaigns to push product promotions and rebates directly to the consumers' devices. Mobile devices are also used in stores to scan product barcodes and find associated product content online, do price comparisons etc. Many retailers are using or experimenting with support for mobile payments in store, where consumers can pay for purchases using their PDAs or mobile phones.

Search

Internet Search is the dominant generator of traffic to e-commerce sites, often surpassing the traffic generated through other marketing campaigns. Most e-commerce providers pay special attention to ensure that their Web sites and product pages are highly ranked by well know search engines. Search Engine Optimization has become a key requirement for e-commerce and digital marketing.

Content Management

Storage, versioning, and publishing of product content have become simpler because of significant advances in content management platforms, such as Microsoft SharePoint®. Brand managers can easily update and publish product content, whether it is text, images, or multimedia. Lightweight workflows can be used for approval before content is moved from staging to production.

Web Services

Over the past decade, Web services have become the lingua franca of the Internet, connecting businesses in ways that weren't possible before. . Today's businesses are more agile and transparent due to inter-enterprise collaboration based on Web services, application-to-application messaging across firewalls, and advanced workflows. IT departments have used service-oriented architecture to better connect and manage their applications, allowing them to support business innovation.

Virtualization and Cloud Computing

Virtualization allows for multiple application environments to be executed on single physical hardware on demand. Hardware virtualization uses software to create a virtual machine (VM) that emulates a physical computer. This creates a separate OS environment that is logically isolated from the host server. By providing multiple VMs at once, this approach allows several operating systems to run simultaneously on a single physical machine.

In the past, administrators had to configure separate servers to configure applications that had conflicting software requirements (such as operating system versions), underutilizing the hardware storage, CPU or memory. Rather than paying for many under-used server machines, each dedicated to a specific workload, server virtualization allows those workloads to be consolidated onto a smaller number of more fully-used machines.

Cloud computing refers to the concept of using hardware resources such as CPU or storage as a service. Instead of purchasing all the hardware needed, IT managers and CIOs can use the required hardware as needed from a service provider.

Virtualization is enabling retailers to dramatically reduce hardware cost by consolidating servers needed to run different applications. During peak times, such as during the holiday seasons, retailers needed additional hardware to support increased traffic to their Web sites and higher number of transactions. Cloud computing, with the ability to dynamically increase or decrease the hardware requirements (referred to as *elasticity*) allows businesses to pay for hardware based on their actual usage.

Advertising

Businesses are increasing their online advertising, even while reducing expenditures on traditional mediums. Search and banner ads are helping to drive more traffic to the e-commerce Web sites. Businesses are placing ads not only on passive Web sites, but also in games and videos that are being delivered through personal computers, game consoles, and mobile phones. Advanced tools are being used to track the effectiveness of marketing and advertising campaigns.

Instant Messaging

For improved customer service on e-commerce sites, businesses are using innovative tools to allow service representatives to connect with customers. Many leading e-commerce sites allow customers to chat with a customer service representative by the click of a button. Several retailers provide a capability called *click to chat*, where a customer can establish a voice over IP connection to talk to a customer service representative in real time.

Payments

Processing online payments has become easier because of easy-to-use services such as PayPal. By using Near Field Communications (NFC) and radio frequency identification (RFID), customers can use phones and other handheld devices to submit payments in physical stores.

Prescriptive Architecture Fundamentals

A prescriptive architecture provides a proven, tested, and repeatable approach to application design that meets the business and technical requirements, while staying within the boundary of constraints and tradeoffs such as cost, performance and security. By following a prescriptive architecture, solution architects can design and build user friendly, scalable, extensible, and reliable software solutions. To reduce the business risks associated with building and deploying a software solution, a prescriptive architecture takes the following factors into consideration:

- The people who will use the solution being developed, and how they will use it
- The existing IT environment and systems that the solution must be integrated with
- How the solution will be deployed in production and what hardware constraints it needs to work with
- Quality requirements, such as security, performance, and manageability

- Business and technology trends that could affect the solution in the long term
- Solution maintenance and management
- Industry-specific scenarios that need to be addressed

Design Fundamentals

Application design is critical for the successful development and deployment of a software solution. A good design should be flexible enough to easily accommodate changes in user requirements, technology, and IT environments. The following subsections describe some key principles to follow when you design an e-commerce solution.

Create a Design That Is Built to Last and Built to Change

Design solutions should accommodate additional users, data, geographies, and so on, and at the same time be able to change to meet new functional requirements and business models. For example, you might be designing a three-tier system that is to be deployed within the four walls of the company. However, a good design should allow you to extend the same solution to be hosted in the cloud without requiring a major rewrite.

Design Incrementally and Iteratively

Avoid creating a big design upfront. Instead, design and verify key components, and prove or disprove key assumption. Then, start adding details, constraints, and so on. Taking an iterative approach might require modifications to previous design decisions; therefore, don't be afraid to go back and make changes.

Use Test-Driven Development

Test-driven development is a practice that is becoming increasingly popular. Where it makes sense, start by first writing the tests that you will need for the code the developers may right. Create designs and code that are testable. Wherever possible, use threat modeling and similar techniques to identify risks and vulnerabilities in your design and your code.

Separate Your Application into Logical Units

Break your application at the user interface, business logic, infrastructure, or the data layers into distinct features that do not overlap. The main benefit of this design principle is that it allows you to test and optimize individual components without having to deal with the entire system.

Create Reusable Code Modules

Make sure that each system component is responsible for only a specific feature or functionality. Make sure that throughout the system, components are reused rather than implementing the same feature or functionality repeatedly.

Use Well-Defined Interfaces

A component should not know the internal workings of another component and should only communicate via an agreed upon interface or contract. This helps ensure that your solution is easy to maintain and extend.

E-commerce Architecture Frame

An architecture frame is simply a table of key considerations that you should refer to as you design and develop a solution. The frame describes the considerations in sufficient detail and identifies the key issues associated with them. Table 2 is an example of an architecture frame to be used for designing and developing e-commerce solutions.

Table 2: Sample E-Commerce Architecture Frame

Area	Description	Key issues
Authentication and authorization	How to choose an authentication strategy. Specific approaches and options for a public-facing site and for administration/authoring interfaces. How to choose an authorization strategy. How to flow identity across layers and tiers. How to store user identities when not using Microsoft Active Directory® directory service.	Lack of authentication across trust boundaries. Lack of authorization across trust boundaries. Granular or improper authorization. Integration of custom user authentication stores.
Caching	How to choose an appropriate caching technology. How to determine what data to cache. How to determine where to cache the data. How to determine the expiration policy.	Caching data that is volatile. Caching sensitive data. Incorrect choice of caching store.
Communication	How to choose appropriate protocols for communication across layers and tiers. How to design loose coupling across layers. How to perform asynchronous communication. How to pass sensitive data.	Incorrect choice of transport protocol. Chatty communication across physical and process boundaries. Failure to protect sensitive data.
Coupling and cohesion	How to choose an appropriate layering strategy for separation of concerns. How to design highly cohesive components and group them within layers How to determine when loose coupling is appropriate between components within a layer	Incorrect grouping of functionality. No clear separation of concerns. Tight coupling across layers.
Data access (can be specific to reads of large lists and transactional data; can be combined with caching)	How to manage database connections. How to handle exceptions. How to improve performance. How to handle binary large objects (BLOBs).	Per-user authentication and authorization when not required. Chatty calls to the database. Business logic mixed with data access code.
Logging and instrumentation	How to determine which information to log. How to make the logging configurable.	Lack of logging and instrumentation. Logging and instrumentation that is too

	How to determine what level of instrumentation is required.	<p>fine-grained.</p> <p>Not making logging and instrumentation an option that is configurable at run time.</p> <p>Not suppressing and handling logging failures.</p> <p>Not logging business-critical functionality.</p>
User experience	<p>How to improve task efficiency and effectiveness.</p> <p>How to improve responsiveness.</p> <p>How to improve user empowerment.</p> <p>How to improve look and feel.</p>	<p>Not following published guidelines.</p> <p>Not considering accessibility.</p> <p>Creating overloaded interfaces with unrelated functionality.</p>
Scalability	<p>How to improve software performance and make it scalable.</p> <p>How to decide and deploy scalability across tiers.</p> <p>How to minimize the incremental complexity as the system becomes more scalable.</p>	<p>Not following published guidelines.</p> <p>Not considering future extensions to user or data volumes.</p> <p>Considering scalability as just a server issue: user experience and scalability have a significant impact on each other.</p>
Security	<p>How to put in the right user authentication and access controls.</p> <p>How to make the system deployment secure.</p> <p>How to secure the data.</p> <p>How to secure communication.</p>	<p>Not defining the security model in advance.</p> <p>Not following published guidelines on design and implementation.</p> <p>Not focusing on backend data security.</p>
Multi tenancy	<p>How to make an enterprise commerce system that can serve across countries and business divisions.</p> <p>How to isolate and process data.</p> <p>How to provide flexible behavior based on a tenant profile.</p>	<p>Not identifying potential multi-tenancy issues in advance (especially when the system is built/ deployed incrementally).</p>
Internationalization and multilingual support	<p>How to make an enterprise commerce system that can function across countries in multiple languages.</p> <ul style="list-style-type: none"> • How to apply Internationalization across tiers • How to deal with multi lingual content and data 	<p>Not considering all user experience and display format scenarios.</p>

The next section provides an analysis of the specific aspects of the architecture frame with related conclusions and recommendations for implementing an e-commerce system.

Authentication and Authorization

Security is one of the primary concerns related to creation, deployment, and management of an e-commerce system. Authentication and authorization are the two main aspects of security. A proper

design and implementation of architectural elements to take care of these aspects can ensure appropriate access security for the system. However, these alone will not make a system completely secure. To make a system more secure, you must address other security aspects, such as data security, network security, and operating system access.

Authentication

Authentication is the mechanism of verifying the identity of the entity (the *actor*) that is attempting to access the system. The actor could be a person or it could be another system that needs to access a service or data from the e-commerce system.

Typically, an e-commerce system has the access points and actors defined in Table 3.

Table 3: E-commerce System Access Points and Actors

Access point	Actors	Type of actor
Public e-commerce site	Anonymous user Registered user Privileged user Any other actor roles that are specific to the business context	External (to the e-commerce entity)
E-commerce site content management interface	Content author Content editor Content approvers Any other actor roles that are specific to the business context	Internal
E-commerce catalog view/ edit	Product manager	Internal
E-commerce order management	Fulfillment manager Fulfillment employee	Internal
E-commerce campaign management	Product manager Marketing manager Marketing employee	Internal
User management	Marketing manager Marketing employee	Internal
Reporting	Product manager Marketing manager Executive team (CxO)	Internal
Affiliate user interfaces	Marketing manager Affiliate entity personal	Internal and external
Catalog retrieval and update services	Back-end system(s) Affiliate applications	Internal and external
Order retrieval and update services	Back-end system(s) Affiliate applications Partner applications	Internal and external

The information in Table 3 is indicative of a typical e-commerce system. In specific systems, there will be variations based on business context and technology elements.

As the list of access points and actors indicates, there is a need for an authentication mechanism or multiple mechanisms to secure the different types of access points. When determining the mechanisms to use, consider the nature of the access points (for example, Internet, intranet, extranet, desktop, and so on), as well as the user information storage and access requirements. Select an appropriate mechanism for users who might use multiple access points to make sure that there is no duplication of identity management.

The following are key considerations:

- Each access point may or may not be part of a single application. In most cases, there will be different applications for specific access points.
- You need the flexibility to store internal and external user information in separate repositories.
- Minimize the need to manage and verify separate user identities for each access point. It should be possible for an actor to have a single identity that is used across all access points.
- Consider the level of protection that the authentication mechanism provides.
- Consider the storage and access features available for the authentication mechanism.
- Consider the scalability of the authentication mechanism.
- Consider any preferences based on the technology stack (for example, Microsoft® SharePoint works well with Active Directory).
- Evaluate the deployment complexity and the cost of the solution when using the chosen authentication mechanism(s).
- Consider your integration requirements. For example, in most integration scenarios, it is preferable to use a standard authentication mechanism for access from heterogeneous platforms.

Evaluate the management requirements for the authentication mechanism, including features and ease of use related to user management.

Authorization

The primary purpose of authentication is to verify the identity of the actors. Actors whose identity cannot be verified based on their credential or claims information will be denied access by the authentication mechanism. However, after an actor's identity is verified, it does not necessarily follow that the actor should have access to all parts of the system.

In an e-commerce system, after verifying an actor's identity, it is important to have an access control mechanism that ensures that only actors who have rights can access specific parts of the system. For

example, a registered user might need to gain access to all parts of the consumer shopping experience, but does not need access to content editing interfaces.

Authorization methods provide this second level of access control after the user is authenticated. Authorization will also have to include handling of non-authenticated or anonymous users (whose identity is not verified) in systems that provide limited access to anonymous users. An e-commerce system, for example, might permit an anonymous user to browse the catalog, view promotions, add items to a shopping cart, apply discounts, and so on. However, to purchase items and check out, the user might need to register and/or provide additional credentials.

Additionally, access control may have to be done at multiple levels. Many systems grant users of the same category the same type of access; for example, all registered users have the same access, and all product managers have the same access, and so on. Some systems might also require user-level access control specific to the identity of the individual user. Many systems have access rights based on other parameters, such as an organizational hierarchy, brand affiliation, department affiliation, and so on.

For these reasons, it is important for the authorization mechanism to provision user access based on user types, user identity, and other profile parameters, if possible. Most authorization mechanisms use a role-based authorization method that groups users into roles and then apply access rights and constraints at the role level.

The key parameters that decide access rights in an e-commerce system could be the following:

- User identity
- User role
- Other user profile parameters

Access rights based on the user identity and user role are the most straightforward and can be generic across systems. An access control based on other user profile parameters can range from being a simple look up to complex logic based on the profile parameters chosen, the nature of logic to be applied on the parameters, and the level within the application at which the access rights are to be applied.

The typical solution will use a role-based authorization solution, and will employ application-specific logic on top of that for addressing more complex requirements. If it is simple enough, the third aspect (other user profile parameters) could be added to the role definition.

User roles are assigned mainly for the purpose of applying access rights in each of the access points or applications. Therefore, it is possible that the same user identity could be treated as belonging to different user roles, based on the access point. For example, a user might access the campaigns administration access point in the role of marketing manager and also access the content management access point in the role of content approver. Therefore, an e-commerce system might require user roles specific to access points at a more granular level.

Table 4 lists the access points and respective authorization requirements for a typical e-commerce site.

Table 4: Typical E-commerce Site Access Points and Authorization Requirements

Access point	Actors	Access control
Public e-commerce site	Anonymous user Registered user Privileged user Other actor roles that are specific to the business context	Role-based + Data access and update rights are user-specific (a user can view and update only his or her user data)
E-commerce site content management interface	Content authors Content editors Content approvers Any other actor roles that are specific to the business context	Role-based, store-based, or channel-based For example, content author (role) for a music store
E-commerce catalog view/ edit	Product manager	Role-based, store-based
E-commerce order management	Fulfillment manager Fulfillment employee	Role-based
E-commerce campaign management	Product manager Marketing manager Marketing employee	Role-based, store-based
User management	Marketing manager Marketing employee	Role-based
Reporting	Product manager Marketing manager Executive team (CxO)	Role-based Other parameters, such as product, brand, store
Affiliate user interfaces	Marketing manager Affiliate entity personal	Role-based + Data access and update rights are user-specific (a user can view and update only his or her user data)
Catalog retrieval and update services	Back-end system(s) Affiliate applications	Role-based or identity-based
Order retrieval and update services	Backend system(s) Affiliate applications Partner applications	Role-based or identity-based
User information retrieval and update services	Back-end system(s)	Role-based or identity-based

Caching

Caching, if appropriately used, can boost the perceived performance of an application, especially a Web-based system. This is increasingly important for e-commerce systems. In a typical Web-based application, the following cache mechanisms are used:

- Content (HTML) caching – Caching of processed HTML output in ready-to-serve form; this eliminates processing the entire page for every request.
- Data caching – Caching of data that is used by the presentation tier; this eliminates going back to database or repository for every request.
- Content delivery network (CDN) caching – Caching of content with a CDN provider for speedier access to data across the globe.

However, because of the dynamic and transactional nature of e-commerce systems, the design and implementation of the caching mechanism should be approached with care, particularly with respect to the following considerations:

- What should be cached or not cached?
- Where should items be cached?
- What should the cache expiration and refresh policies be?
- How should the cache behave in different deployment configurations? For example, is it appropriate to require entirely different cache logic based on whether the system is running on a single computer or in a Web farm?
- How should read and write data be handled? Is it appropriate to include inventory information, or order information in the data to be cached?

Communication

In today's enterprise technology landscape, e-commerce system components as well as the other systems that they interact with are deployed in a distributed fashion. This requires the use of appropriate communication mechanisms between internal system components and between external systems.

The different communication channels that are typically required for an e-commerce Web system are described in Table 5.

Table 5: Typical Communication Channels for an E-commerce Web System

Communication boundary	Typical protocols
Browser-based user access	HTTP/ HTTPS
Communication between application layers	In-process Transmission Control Protocol (TCP) HTTP Simple Object Access Protocol (SOAP)
Communication with back-end systems (for example, with enterprise resource planning [ERP] systems)	HTTP SOAP APIs (In-process) Open Database Connectivity (ODBC) Component Object Model+ (COM+) Host protocols

	Proprietary communication Flat file transfer (File Transfer Protocol [FTP])
Payment Integration	HTTPS SOAP
Analytics systems	HTTP
Search engines	HTTP

Browser-based User Access

Browser-based user access is mandatory for any Web system. Typically, not all pages will need secure access. Also, because secure access is more resource intensive, selective use of secure access will facilitate a better performing system. Therefore, an e-commerce system should be designed to work in a mixed mode – with select pages that have secure access and the rest with HTTP access.

Communication Between Application Layers

Communication between application layers is a key consideration and a design decision. Microsoft offers multiple choices for implementing this communication, and you should evaluate these options carefully before choosing a specific pattern.

Typically, business components of an e-commerce system perform frequent data reads and updates. Therefore, we do not recommend that you implement the business logic layer and the data access layer on separate servers, unless your requirements require that you do so. Placing the business logic layer and data access layer on the same server enables faster data access and more efficient caching.

The presentation layer can interact with the business layer in either an in-process mode or by using one of the remote communication protocols. You should determine which approach to use based on the following requirements:

- A requirement to host business logic in a separate security zone (this might be required if you implement very sensitive business logic).
- A requirement to scale the Web and application tiers separately.
- A requirement to expose business logic as services because of constraints of the hosting architecture or because you must follow an enterprise architecture guideline when you host the business layer on a common platform.
- A requirement to expose business logic as services for consumption by applications that represent different channels and access points.

Communication with Back-end Systems

Communication with back-end systems is a key consideration for e-commerce systems because much of the important data must be synchronized with other systems, such as the ERP system. The choice of communication mechanisms is limited only by technology; it is important to choose a communication mechanism and protocol that suits the user experience, performance requirements, and data currency needs of the given context.

Payment Integration

This is an essential part of most e-commerce systems and is one of the key areas that require robust security, transaction sequencing, and exception handling. In most cases, the payment systems will need to be integrated over the wire via a standard protocol. Different techniques, including HTTP redirects, encryption, and secure HTTP and SOAP access are usually employed for communication with the payment system. There is not a specific “best” protocol that can be recommended. However, the choice of communication mechanisms must be made based on the following factors:

- Type of payment transaction required
- Communication mechanism supported by the chosen payment system
- Type of user experience intended
- Level of security required
- Nature of transaction processing and exception handling needed

Web Analytics Systems

Web analytics systems work by collecting browsing data from the Web site that is being monitored. Web analytics require the Web site to send data to the analytics system. Most analytics systems collect this data via HTTP interfaces. The e-commerce system Web pages use embedded scripts to record each browser visit to the page, and then forward the information to the analytics system.

Search Engines

An e-commerce system must be accessible by search engines to enable better positioning in search results and to encourage traffic to the site. Search engines act very similarly to browser clients, and use HTTP to access the site.

A more detail discussion of the rest of the considerations in the Architecture Frame is still work in progress and will be provided in the future releases of this white paper.

Technical Capabilities for E-commerce and Digital Marketing Solutions

In a previous section, we identified some of the technology enablers for e-commerce and digital marketing system. In this section, the key technical capabilities required for a robust and flexible e-commerce architecture are described in greater detail. This is followed by the description of Microsoft technologies that provide these technical capabilities.

Content Management, Publishing, and Delivery

Management and delivery of rich content is an essential part of a digital presence on the Web. For enterprises, this means deployment of a robust platform that can support the presentation and handling of rich content, authoring interfaces, content repositories, versioning, approval workflows, and publishing. The key aspect here is that content management is no longer an isolated function. Instead, the platform must be capable of tighter integration with other systems and other parts of the e-commerce system itself.

In the Microsoft technology stack, this capability is provided by the SharePoint application platform.

Rich Web and Client Presentation Capabilities

The Web user experience is being enhanced by newer ways of data representation and rendering within the browser. These enhancements are enabled by the technologies such as Asynchronous JavaScript (AJAX) and Silverlight, along with a combination of practical information architecture design. Content management and the Web presentation platform work together to support rendering of rich content and media, while ensuring the overall security and performance of the system.

These set of capabilities are supported out of the box for most scenarios with the technologies in the ASP.NET stack including ASP.Net AJAX and Microsoft Silverlight. Moreover, the ASP.NET stack provides for additional development and integration of custom components and scripts as required for enhancing the user experience in a given application context.

Stable Core Web Framework

Stability, reliability, and scalability are core qualities that are expected in a digital Web and e-commerce system. In fact, these features are taken for granted by many organizations and IT developers when they choose a platform. At the heart of the platform should be a robust, extensible, and scalable Web rendering engine that can support modern programming and transaction paradigms at both the server and client ends of the Web experience. Support for server-side processing, large scale concurrent access, caching, session management, secure communications, transactions, and so on, are considered core quality parameters that should be supported by the platform's Web framework.

In the Microsoft technology stack, this set of capability is provided by a combination of the Windows Internet Information Services (IIS) and ASP.NET services.

Data Repository

Technically, at the heart of every digital marketing and e-commerce system is content and data management, storage, rendering, and transactions. By default systems should persist most of the content and transactional data. Most of this data requires not just reliable storage, but a mechanism to query and process the data in multiple ways for rendering and transactional purposes. This calls for a robust and scalable data engine, and in most cases, a relational database that can store different content and data formats, as well as support high volumes and concurrent processing of transactions.

Microsoft SQL Server® provides a feature set that addresses this solution area. SQL Server can meet all the online transaction processing (OLTP), storage, and online analytical processing (OLAP) needs of an e-commerce system.

Commerce Engine

Across most e-commerce systems, there is a common pattern for rendering content and data and processing transactions. For example, most systems use the concepts of catalogs, products, variants, wish lists, shopping carts, check out process, billing and shipping addresses, shipping options, and payment methods. The basic pattern in which these are tied together is fairly consistent, while the

specifics of how e-commerce and business contexts overlap and how the logic is implemented can vary considerably from enterprise to enterprise. The existence of a basic pattern provides an opportunity for the creation of a core commerce engine that can handle the common functions, and leave the development team to focus on the implementation of core business processes and logic.

Microsoft Commerce Server 2009 Multi-channel Foundation Services provides a comprehensive framework for such a commerce engine. The next two sections provide a detail description of Commerce Server architecture and implementation best practices.

Integration Framework

Rendering content and data, presenting it to users, and enabling e-commerce transactions is the primary function of a digital Web and e-commerce system. However, in practical business scenarios this almost always means close integration with one or more additional systems for data exchange and process flow. The additional systems could be for user profile sharing, catalog data updates, order processing, shipment tracking, and payment or other transactional elements.

For these reasons, it is important for an enterprise class e-commerce system to provide for future extensibility and scalability by using standard interfaces. Typically, integration turns out to be an incremental and evolving process, and it is important that the integration function is not hard-wired into a set of pre-determined components, but rather built in as part of a framework that can be extended for new interface points and scenarios.

The following Microsoft technologies can be used individually or in combination to address integration requirements:

- BizTalk Server 2009
- Windows Communication Foundation (WCF)
- Windows Workflow Foundation (WF)
- SQL Server Integration Services (SSIS)

Analytics

The ability to enable decision making is one of the key aspects that should be supported by a digital marketing and e-commerce system. Analytics is the primary technology that facilitates decision making, along with other components, such as search and reporting. An appropriate analytics mechanism will help to reveal critical views of data from a large and growing repository of transactional records across systems.

In the context of an e-commerce system, it is very important to have the ability to tie together data from different application areas such as catalogs, user profiles, transactions, and campaigns. Cumulative data can be used to analyze, measure, and monitor multiple aspects, such as user experience on the site, site performance, performance of specific products and product categories, transactional effectiveness, and so on.

Analytics involve key platform capabilities, such as data collection, large volume data storage, data synchronization, data analysis, and reporting.

Microsoft SQL Server provides analytical capabilities such as storage, transactions, analysis, and reporting out of the box.

Scalable Application Server Platform

Presentation of content and information and processing of transactions are two key capabilities of a digital marketing and e-commerce system. These capabilities are not used just for Web browsing and purchasing, but are also integrated with mobile devices in varied ways, including mobile browsing, applications, and texting, among others.

To meet future content delivery and content rendering requirements, an e-commerce system must be scalable to handle the growing number of users and traffic. The content update components also need to be scalable, but this applies primarily to a projected increase in the amount of data, rather than user and traffic growth.

To enable real-time transactions that integrate well with the online experience, it is important to have components in place that support concurrent transactional operations in real time across repositories and back-end systems. It is also important for these components to be able to scale up or out. Many systems will start small, and it should be possible to have a component structure that can support low scale deployment with minimal costs that can then gradually mature into a high scale deployment. For these reasons, having a truly scalable application server platform that enables transactional processing is a key component of the architecture.

Microsoft Windows Server and IIS work together to form a reliable and scalable application server platform to meet these needs.

Search

The challenge with content and data management systems of high volume is to capture the maximum information for current and future processing, and at the same time have a mechanism to find that information quickly so that users can make choices. Search is an integral component that can support both finding the data and decision making. The capability to search correctly and fast enough is a key factor in finding data. The capability to present relevant search results in a contextual manner is a key factor in decision making.

Another factor related to search is how friendly the system is to searches by external search engines. External search engines have a big influence on the browsing behavior of users. It is not only important for the e-commerce system to be self-sufficient in providing a search experience, but it is also important that the system integrates well with public Web portals and search engines to drive more traffic to the site.

Therefore, in any digital Web and e-commerce system, search capabilities are important factors for success, along with appropriate mechanisms to enable SEO (Search Engine Optimization) and related analytics.

The Microsoft technologies that can address these needs are:

- SharePoint search services
- Microsoft Search Server 2008

- FAST search
- Microsoft Bing search

Campaign Management, Delivery, and Tracking

The experience that a user has when visiting a Web store is very different from the experience of visiting a real world store. This is true not just for the purchase process itself, but also for what the user can see and do before, during, and after the purchase process.

In a real world store, retailers practice a wide variety of techniques to influence buying behavior and purchase decisions. These techniques range from traditional discount offers to using real estate and aesthetics. The customer experience in the digital world is very different, which might limit the application of some of the real world techniques. However, the digital experience can enable new avenues for e-tailing.

Campaigns in the e-commerce system can include a selection of incentives—from highlighted discount offers on specific product pages to integrated e-mail, search, and portal-based product campaigns. Typically, a campaign needs to support discounts, packaging, promotions, cross selling, advertisements, and targeting.

In the context of a multichannel e-commerce system, it is essential to have a framework in place to target, launch, execute, monitor, and analyze campaigns effectively. Microsoft Commerce Server 2009 provides out-of-the-box support for most of these campaign management and delivery requirements. For broader campaign management, delivery, and analytics that include search ad placement and promotions via Web portals such as MSN, a comprehensive set of features are provided by the Microsoft Atlas product suite.

Microsoft Platform for E-commerce and Digital Marketing

Microsoft technologies provide a comprehensive set of options to realize an e-commerce and digital marketing system architecture, as illustrated in Figure 2. The platform components are modular and extensible, which means that the platform can be built and configured at different scales of functionality and performance to meet specific needs. The platform is versatile and flexible enough to cater to different business requirements. It can work for someone who is starting a small-scale Web presence that can be expanded to a larger scale with future investments. It also meets the requirements of someone who needs to develop an e-commerce framework, and then provision sites and features incrementally.

Microsoft Platform for E-commerce and Digital Marketing

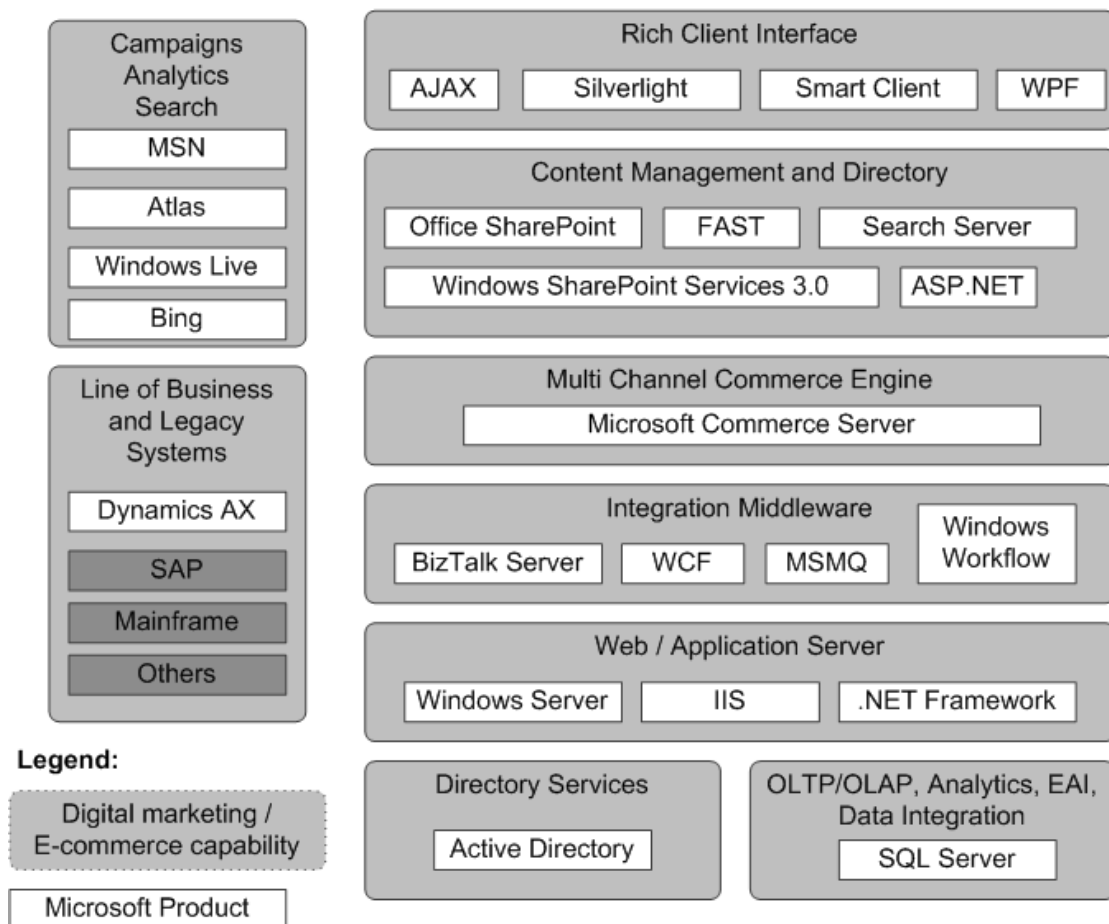


Figure 2. E-commerce framework

.NET Framework 3.5

The Microsoft .NET Framework 3.5 platform provides the following benefits:

- Proven scalability and performance.
- Fully managed environment for memory management and code access security, in addition to standard security provisions such as encryption.
- A rich set of components and libraries that allow you to build applications quickly.
- A uniform programming model across Web applications, Windows applications, and services-based applications.
- Windows Communication Foundation (WCF)—a unified communication framework that provides one model for multiple communication channels and bindings, including SOAP, binary over TCP, and queue-based communication.
- A base for building service-oriented applications that adhere to proper data, service, and message contracts in the communication layer.

- Support for object-oriented programming by means of a rich class library and inherent framework support.
- Advanced language features such as Language Integrated Natural Query (LINQ) for better programmability of datasets, along with a related framework implementation that uses LINQ specifically for database and XML processing.
- ASP.NET, which allows .NET to be used for Web applications, and provides capabilities such as AJAX and multi-factor support, including mobile support by means of ASP.NET mobile controls.
- Integration with Visual Studio Team Suite, a uniform integrated development environment (IDE) for rapid application development.

Windows Operating System

In addition to the Microsoft technologies mentioned previously, the Windows Server® operating system provides the foundation for high performance computing for e-commerce applications. Windows Server is a robust, scalable operating system that provides all the functionality required to support e-commerce applications. In addition to the base operating system, Windows Server includes the following:

- IIS, a full-featured Web server
- Message Queuing
- COM+, a sub-system for hosting COM and .NET components that provides scalability and extensibility features

Windows Server provides support for scalability, both for scaling up (to a maximum of 64 processors for the Datacenter edition), and for scaling out (up to 8 nodes in a cluster). The 64-bit edition allows the operating system to make full use of 64-bit hardware.

SQL Server

SQL Server is primarily a high-performance database engine, and provides additional capabilities in terms of data integration and analytics.

The SQL Server database server has a proven track record of high performance, reliability, and scalability. SQL Server can be integrated with the .NET Common Language Runtime (CLR) within the database engine to provide a mix of relational operations and logic-intensive operations.

SQL Server Reporting Services (SSRS) provide features for designing, invoking, rendering, and scheduling reports. It also provides a default portal for managing report publishing and access, as well as report definition and customized rendering. SSRS also allows rendering in multiple formats, such as Microsoft Excel, PDF, and HTML. SSRS can be used to design reports and printable documents.

SQL Server analytics allow large volume data collection and analysis. Combining SQL Server analytics with SSRS, Microsoft Office, and Microsoft Office SharePoint Services results in a comprehensive platform for data collection, analysis, reporting, and presentation. Figure 3 illustrates this comprehensive solution.

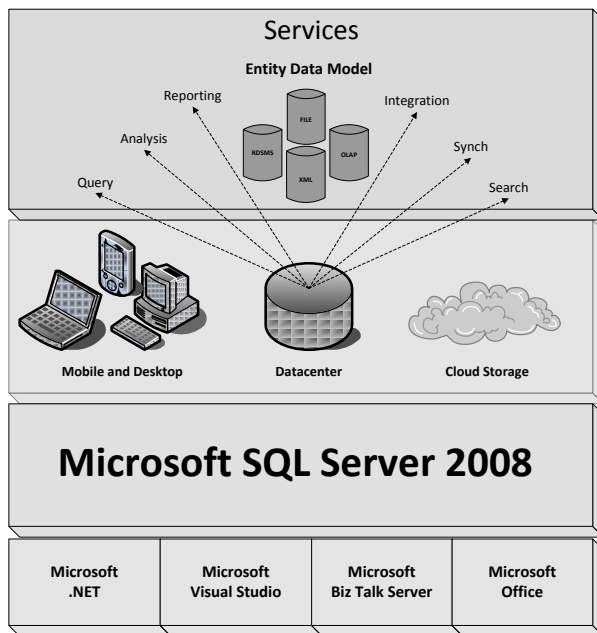


Figure 3. Comprehensive analytic and reporting services

Microsoft Commerce Server

Commerce Server provides a rich set of e-commerce core functionalities to manage catalogs, users, orders, campaigns, and service integration. Commerce Server integrates with SharePoint Services to provide ready to use e-commerce Web Parts and content management capabilities. Its core is enabled by a multi-channel foundation that facilitates configuration of user device (media channel) level e-commerce functionalities and features.

Figure 4 illustrates these capabilities.

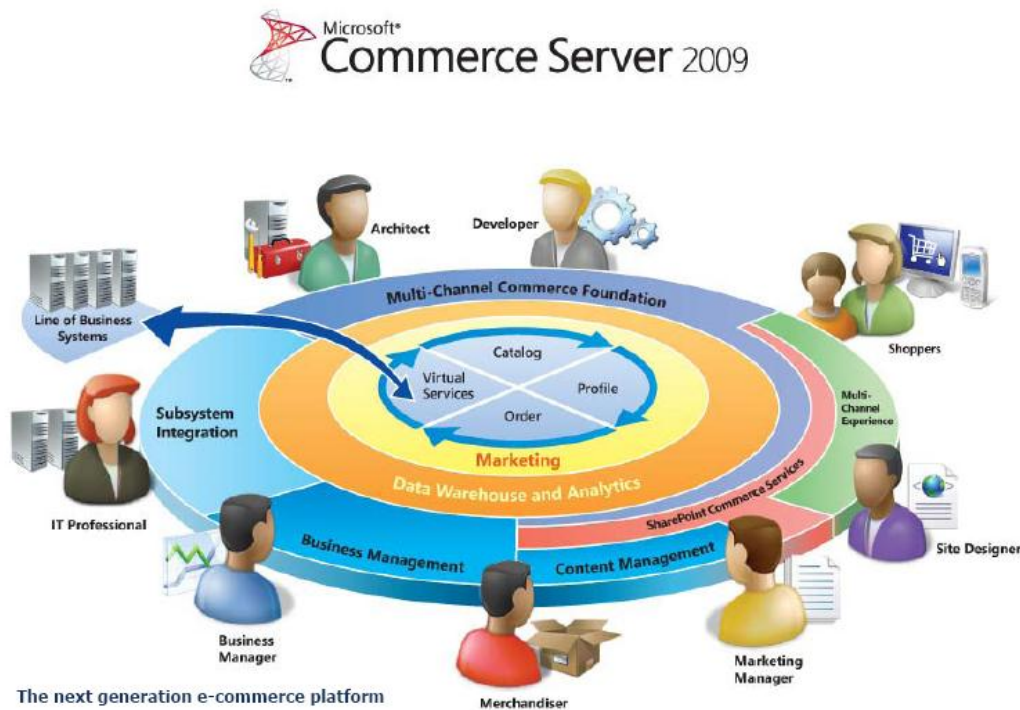


Figure 4. Commerce Server capabilities

Microsoft BizTalk Server

Integration is one of the key technical areas related to deploying transactional and e-commerce systems. Microsoft BizTalk Server provides features that address multiple integration architecture patterns. BizTalk Server addresses the core aspects involved in message definition, transformation, exchange, and monitoring, and provides core components and a scalable framework for deploying an enterprise-level integration system, including hub and spoke systems, middleware, and enterprise service bus (ESB).

BizTalk features are important in an e-commerce system because they address integration with back-end systems that manage inventory, fulfillment, finance, and other areas.

ASP.NET

ASP.NET is the server-side Web technology that provides a framework for creating dynamic Web applications. ASP.NET is built on the .NET Framework and integrates with different versions of IIS to provide a scalable, dynamic Web content delivery platform that provides UI, security, performance, and utility features such Web session management and caching.

ASP.NET applications can be created by using any .NET Framework development tool, such as the Microsoft Visual Studio® development system. It provides a flexible and extensible programming model, with well-defined APIs. The modular architecture of ASP.NET facilitates customization, and can be easily integrated with other .NET and non-.NET systems and services.

One of the primary contributors to a Web application's success is its user interface. ASP.NET provides dynamic Web controls that can be used in custom applications to implement custom controls.

SharePoint Server

SharePoint Server is Microsoft's platform for information portals, content management, and collaboration. It also provides capabilities such as search, analytics, business processes forms, reporting, and social computing. SharePoint Server is built on ASP.NET technology and provides a consistent way for delivering static and dynamic content on the Web. Figure 5 illustrates the capabilities of SharePoint Server.

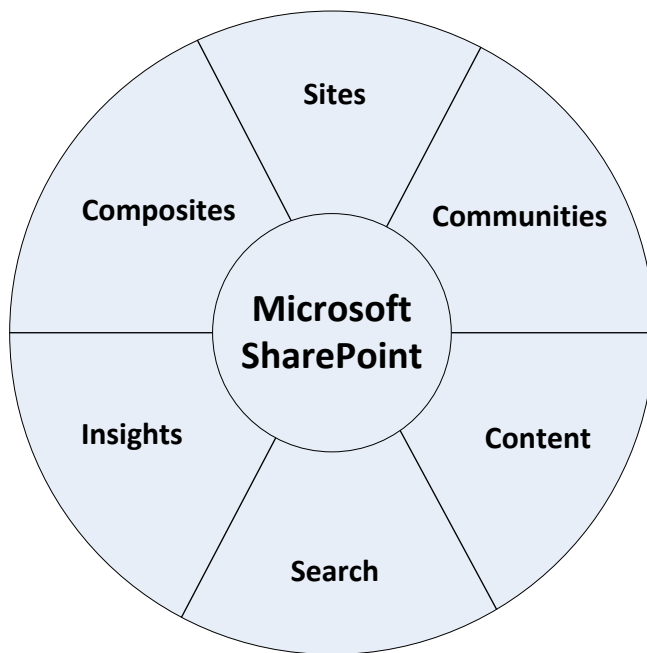


Figure 5. SharePoint Server capabilities

The ASP.NET base and the flexibility provided by the Web Part framework makes SharePoint a good choice for building e-commerce front ends with built-in content management and publishing. Such systems can also use SharePoint security and feature/Web Part provisioning to build and deploy customizations and extensions that provide additional business and domain specific functionality, such as shopping carts and checkout processes. Also, because SharePoint runs on the ASP.NET infrastructure, any extensions, including Web Part development, are similar to and can use common ASP.NET development techniques, making the development and maintenance of customizations and extensions a natural process for a .NET team with SharePoint skills.

Bing

Bing is Microsoft's search and decision engine that provides a search experience grounded in the understanding of consumer behavior and needs. In the past decade, the number of Web sites has grown from just over 8 million sites to more than 230 million (October 2009 Web Server Survey, Netcraft). Netcraft also estimates that there are more than 300 billion Web pages, which host all manner of content from static Web pages to video, audio, and social networking sites. New tools are needed to turn all this information into results that allow a user to make decisions. Bing includes features that provide a more engaging search experience and for decision making, such as the following:

- Relevant results
- Enhanced auto-suggest
- Real-time searching
- Enhanced preview
- Middle-of-the page answers
- Search sharing

- Bing for mobile devices
- Bing maps
- New video experience
- Visual search
- Cross-session history
- Simplified key tasks
- Bing shopping
- City slide shows
- Weather event results
- Local information
- Travel-related services
- Health (based on Wolfram|Alpha results)

Businesses can use Bing for online advertising by placing search or display advertisements in the context of search queries. They can use Bing maps to provide store location services that provide point-to-point directions. Additionally, they can use Bing's search and social networking capabilities to provide an immersive customer experience.

Bing helps businesses and consumers make better decisions by collecting, analyzing, and uniquely presenting real-time content from Twitter and Facebook. This allows businesses to gauge the consumer interest and concerns about their products and services.

FAST

The FAST Enterprise Search Platform (ESP) is the enterprise search solution from Microsoft that provides relevant, personalized search results, unifying structured and unstructured content across hundreds of document types. The FAST Unity federation framework extends FAST ESP to blend and manage content from multiple search indexes and includes support for over 85 languages. FAST ESP provides the following:

- Contextual and dynamic navigation – FAST ESP analyzes customer query terms and categorizes the results. It dynamically guides customers to the appropriate product category, changing the customer interface to offer them additional refinement options based on their queries. FAST ESP displays context-sensitive choices that relate to the query terms. The value of FAST ESP's contextual navigation is that the search engine can drive the site structure and navigation based on its understanding of the source content. Furthermore, it can correctly interpret customer queries to not only return meaningful information, but also guide the customer to the appropriate categories. It also provides advanced query refinement to add and remove filter choices
- Featured content and product promotions – FAST ESP offers businesses the ability to highlight featured content based on business goals, inventory, and product promotions. The ability to inform customers of additional merchandise and promotional offerings allows a

business to manage its product line while potentially creating opportunities for additional revenue streams.

- **Advanced query processing** – FAST ESP’s advanced query processing offers customers “no fail” searches. It employs features such as spell check, synonym processing, and other features to interpret customer-supplied query terms. . This allows customer to quickly find the product for which they are searching, even if they mistyped the query terms.
- **Federated search** – Through its search federation capabilities, FAST ESP facilitates querying of different external data sources (for example, databases and public Web sites), aggregating the results, and providing a single unified display of results to users. This federation capability takes the source system’s knowledge of its data and aggregates the search results based on FAST ESP’s relevancy and ranking factors. This provides users with a single, comprehensive interface of returned results based on a single query. Additionally, FAST ESP offers many different types of content connectors that support multiple languages across 400 document formats, enabling FAST to index a variety structured and unstructured data sources to give users a unified interface.
- **FAST and SharePoint** – FAST ESP and SharePoint 2007 are integrated to provide search-based portals for information discovery and collaboration. FAST SharePoint Connector allows indexing of SharePoint data with full support for capturing and honoring SharePoint document-level security permissions. The FAST ESP Web Parts makes it easy to build FAST ESP–based search sites in SharePoint by simply dropping in and configuring the appropriate Web parts. These Web parts are designed to be open and customizable, and support the use of XSLT for styling.

Search Server

Microsoft Search Server 2008 allows businesses to empower their people to find the information they need quickly, through a familiar Web-style search interface and an easy-to-use query syntax. Search Server 2008 Express is a free download from Microsoft that enables relevant search results immediately without extensive configuration, by using a ranking engine developed in collaboration with Microsoft Research and Windows Live Search.

Windows Presentation Foundation (WPF) and Silverlight

Windows Presentation Foundation provides the framework for rich client interfaces. This, combined with the Microsoft technologies for auto update and data synchronization, enables development of feature-rich smart client applications with offline capabilities. Windows Presentation Foundation (WPF) gives developers a unified programming model for building rich Windows smart client user experiences that incorporate UI, media, and documents

Microsoft Silverlight™ is a powerful development platform for creating engaging, interactive user experiences for Web, desktop, and mobile applications when online or offline. Silverlight is a free plug-in, powered by the .NET Framework and compatible with multiple browsers, devices, and operating systems, bringing a new level of interactivity wherever the Web works.

These technologies provide opportunities to create engaging and uniform experiences for users on the desktop and on the Web. In an e-commerce scenario, WPF might be used to build impressive

client applications that target end-user desktops, kiosks, or devices. Silverlight can be used to deliver a rich Web experience that combines catalog data and marketing content.

Atlas

Microsoft Atlas services provide digital media technologies for advertisers, agencies, and publishers. The solutions for advertisers and agencies enable the unified management of digital marketing campaigns across display banners, rich media, search, video and Web sites.

The key features of Atlas solutions for advertisers are:

- Atlas Media Console – A single center for campaign planning, delivery, and reporting—for display advertisements and beyond.
- Atlas Rich Media – Comprehensive, easy-to-use rich media campaign management tools and services.
- Atlas Search – Precision management for paid search campaigns, spanning all engines, keywords, and bids.
- Atlas Advanced Analytics – A blend of technology, analysis, and optimization that significantly upgrades a Web site's performance.

The solutions for publishers maximize advertising revenue across all available and developing digital media platforms.

Active Directory

Microsoft Active Directory® directory service is a central component of the Windows platform. It allows a business or enterprise to manage the identities and relationships that make up network environments. Active Directory can act as the central directory service for the enterprise, and store user data and other information. Active Directory can be used as a user authentication store and works in a well-integrated manner in a Windows Server environment. All Microsoft server products can natively integrate with Active Directory. Active Directory also supports open interfaces and can be called as an LDAP service.

Microsoft Dynamics AX

Microsoft Dynamics® AX 2009 is a comprehensive business management solution for midsize and larger organizations. Dynamics AX 2009 software works like and integrates with familiar Microsoft software, such as Microsoft Office and Microsoft Office SharePoint Server 2007. Dynamics AX is built to make it easy to do business across locations and countries by consolidating and standardizing processes, helping to improve productivity, providing visibility across an organization, and helping to simplify compliance.

For application-level integration Dynamics AX provides programming APIs and integration adapters. Apart from providing core business enabling functionality, Dynamics AX provides facilities for Web enablement and cloud integration that support multi-channel e-commerce. Dynamics AX can provide multi-channel integration and Web front-end interfaces natively or it can integrate with a SharePoint or Commerce Server-based front end, depending on solution requirements.

Architecting Digital Marketing and E-Commerce Systems with Microsoft Commerce Server 2009

Commerce Server 2009 provides a multilayer framework with ready-to-use components and services at every layer. The basic architecture supports service orientation and is flexible enough for deployment using variations of object-oriented and service-oriented architecture (SOA) patterns, depending on the usage context.

Typically, there are two ways to use Commerce Server in a typical e-commerce solution:

- You can use Commerce Server *as* the solution – In this approach, you use the ready- to-use components, services, and features, including the Commerce Server user interface.
- You can use Commerce Server as a platform – In this approach, you use the programmability and extensibility of Commerce Server to build additional components and context-specific functionality, based on Commerce Server.

Commerce Server has a modular architecture that allows you to use Commerce Server features along with features from other products, such as SharePoint and BizTalk. This modularity allows you to use the Commerce Server framework as a whole solution or as part of a composite solution, by using the Commerce Server foundation components and APIs.

Later sections of this paper discuss these two approaches in detail, and make recommendations about when to use Commerce Server as a solution and when to use it as an integration platform. We also describe best practices for using Commerce Server in either of these modes.

A Methodology for Architecting a Commerce Server 2009–based System

As with architecting most other business systems, the architectural design for a Commerce Server 2009 system is driven by the functional and architectural frames. These provide the context that the system must function within. Therefore, it is critical to use processes, techniques, and tools that work within these frames, and the solution design and implementation derived from them.

Typically, architecting a Commerce Server 2009 system involves the techniques described in the following subsections.

Scenarios

An effective technique for capturing the functional frame is to identify scenarios and describe them in the context of the proposed solution. According to Microsoft Solution Framework (MSF) Agile process guidance, a *scenario* is defined as a type of work item that records a single path of user interaction through the system. A scenario records the steps used to reach a goal or complete a process. Some scenarios describe a successful path, while others describe an unsuccessful one

Ultimately, a system should be designed and built to provide solution features or functionality. However, to determine this functionality, it is important to first identify the scenario. Identifying scenarios gives a true picture of system usage, which can lead to identifying special cases, as well as some quality of service requirements.

User Stories

User stories are a more detailed form of scenario descriptions. These describe the details of user behavior, data collection, and processing at a more granular level. If the design team uses this approach, it can choose to develop user stories for select scenarios or for the complete set of anticipated user actions, depending on the nature and complexity of the system being built and the implicit understanding that the team has about the business context and requirements.

Identifying Non-Functional Requirements

Non-functional or quality of service requirements influence the decisions and tradeoffs identified in the [E-commerce Architecture Frame](#) section, described previously. These requirements overlap with the functional requirements, and the overlapping areas should be used to derive the key architecture and design decisions.

Product Feature Alignment

One of the key considerations when architecting a Commerce Server 2009 system is to ensure appropriate use of the product and platform features. Your approach should map the correct feature to the solution requirements. For this reason, you should not use the typical waterfall development process model, where you capture requirements before you begin the actual design. During the project planning phase of the development life cycle, you should analyze your requirements and compare them to the Commerce Server features to determine whether existing functionality will meet your needs and to focus design and development efforts appropriately.

Figure 6 is a sample comparison of functional requirements with existing Commerce Server functionality.

No.	Feature	Feature Details	Product / Platform Feature Used	Custom Development
	Catalog Management			
	Basic product catalog	<ul style="list-style-type: none"> * Create, update, delete, retrieve categories, products, product variants, properties. * Hierarchical catalog * Product belonging to multiple categories 	<ul style="list-style-type: none"> * Standard catalog feature * Product category associations * Primary parent category * Catalog and inventory manager * Catalog manager * Sharepoint Commerce Services web parts 	* Custom webpart for product details
	Catalog sets	* Grouping multiple catalogs to target specific user profiles - like individual, customer	<ul style="list-style-type: none"> * Catalogset * Catalog and inventory manager * Catalog manager * Commerce Server manager 	
	Virtual Catalogs	* Catalogs with different price books, discounts, currency, language, etc.	<ul style="list-style-type: none"> * Static/ dynamic virtual catalogs * Catalog and inventory manager * Catalog manager 	
	User defined catalog attributes / properties	* Like size, color, etc.	<ul style="list-style-type: none"> * Property definition * Category definition * Property definition * Catalog and inventory manager * Catalog manager 	* Customize webpart for display of custom properties
	Product search	<ul style="list-style-type: none"> * Ability to search for products based on criteria * Sorting ability 	<ul style="list-style-type: none"> * Product sequencing * Dynamic and full-text search * Catalog and inventory manager * Catalog manager 	* Customize the search display UI
	Product recommendations	<ul style="list-style-type: none"> * Most popular products * Recommended products * People who bought this also bought..." etc * Cross sells * Upsells 		<ul style="list-style-type: none"> * Customize the style * Custom web part for displaying recommendations

Figure 6. Sample comparison of Commerce Server functionality to functional requirements

Identifying Required and Appropriate Customizations

By using Commerce Server 2009, you can leverage the functionality that is available out of the box and develop additional features with custom code. This approach ensures optimum timelines for development, testing, and launch. It is important to make decisions about which Commerce Server features to use, what should be built as custom components, which of the custom components should work with existing Commerce Server components (for example, pipelines or profiles), and whether any customizations or extensions to the Commerce Server framework itself are warranted. You should attempt to keep customizations and changes to the Commerce Server framework to a minimum, while ensuring appropriate use of existing components.

This approach allows you to layer the architecture and employ two different development paths: one for configuration and customization and the other for custom component and page development. This approach can streamline your iterative development process and help to provide your business with an early view of how the solution will work. For example, Iteration 1 could provide a base version, showing the functionality provided out of the box (including SharePoint Commerce Services Web Parts). Subsequent iterations could evolve this view into a form that more closely resembles the end solution by adding custom components incrementally.

Identifying Design Patterns for Customizations

When you use Commerce Server, it is important to first establish usage guidelines for features and platform components. Because the platform provides multiple options for layering and deployment, you need a clear definition of the application layering that you will use, as well as a plan for how the components will be structured. For example, you will need to make the following design choices:

- Will the presentation layer call the foundation services directly, or will it use a wrapper layer that provides caching?
- Will the foundation services be invoked in-process or by service calls?

While the platform allows you to make these decisions, your choices should be informed by the requirements and context of your solution. As you make these decisions, document them as application-specific patterns to be used in your implementation. For example, you might have patterns for the following technical scenarios:

- How does the presentation layer fetch catalog information from Commerce Server?
- What should the implementation be if the checkout process has to update the Commerce Server database and simultaneously update information in the inventory and payment system?
- How should fetch operations use caching? Which caching method should be used for which type of data?

These and other key technical scenarios can be identified and represented as class and sequence diagrams that illustrate the respective patterns.

Identifying the Integration Points

Commerce Server 2009 includes support for integration with legacy and line-of-business systems. The architecture is compatible with services-based integration, and, with the introduction of Commerce Server adapters for BizTalk, it can also be easily integrated with BizTalk-based architectures.

It is important to identify integration scenarios upfront and make decisions about integration patterns. Some of these patterns could use the BizTalk components, allowing the adapters and BizTalk features to be leveraged, and some of them might use other mechanisms, such as SSIS or custom components, depending on the nature and frequency of the integration interfaces.

In many cases, the e-commerce system will need to interact with back-end interfaces in real time; for example, to display updated inventory information or to send order information for fulfillment processing. In such cases, it is also important to identify and map the quality of service requirements list for the overall solution to specific interfaces.

Identifying Additional Products and Features

Apart from the core commerce framework, the system design will have to address additional solution areas that Commerce Server does not support out of the box. These include areas such as content management and search that are essential to the e-commerce architecture.

As indicated previously in this paper, most of these areas are addressed fully or in part by other Microsoft technologies. Therefore, we recommend that you evaluate the existing technologies to determine whether they meet some or all of your requirements, and then make appropriate decisions about product selection and custom development. The identification process should resemble the following:

Content management:

1. Identify scenarios and features.
2. Compare them with SharePoint, other products, and custom development.

Search:

3. Identify scenarios and features.
4. Do a product alignment analysis of Commerce Server search (for data) and SharePoint search.
5. Evaluate the use of FAST or an alternate search engine, if required.
6. Evaluate and decide whether to integrate with online search services such as Bing.

Campaigns and analytics:

7. Identify scenarios and features.
8. Determine whether existing Commerce Server features can provide required functionality.
9. For global campaigns involving Live search and portals such as MSN, determine if Atlas can provide the required functionality.

10. Evaluate the use and integration of any other product or in-house software that has been identified previously.
11. Finalize the product development plan.

Deployment Architecture and Capacity Planning

Quality of service requirements can be realized not only by appropriate use of platform features and coding best practices, but also required by appropriate planning and deployment of the solution infrastructure. Commerce Server 2009 provides flexibility by supporting multiple deployment configurations, including both SharePoint and Commerce Server configurations.

Web Layer

One of the key considerations in Web deployment is *application modularity and isolation*. These qualities can lead to security management improvements in the short term and to improved scalability in the longer term. In addition, hosting the commerce site in the SharePoint shell allows you to more easily manage the site hierarchy. You can use the SharePoint features and administration tools to extend the site and provide independent access points for separate authentication mechanisms.

In addition, the ability to partition the web to site collections and individual site hierarchies allows you to create isolated front-end access points. For example, the access points could be brand specific or country specific or could use any other criteria. Taking a careful look at the deployment design options in SharePoint, and then comparing them to your design decisions are important steps in finalizing the architecture.

Application Layer

Another important aspect in deployment is deciding on the physical layering of the application. Commerce Server supports both two-tier and multi-tier deployments. As in any other transactional system, you should carefully evaluate your requirements for scalability, availability, and security when you design the physical layer. For example, if the web and app layer scalability requirements are homogeneous and there is no specific requirement to host application logic in another zone, it is appropriate to use a combined web-app tier deployment, with just the database layer and integration layers separated in another zone.

Database Layer

Configuration of this tier is critical to the system because most operations, including content delivery, are dependent on fast access to data. You should determine the deployment configuration for the SQL Server database by evaluating the solution architecture and sizing data. You can use clustering or mirroring for availability purposes, after evaluating scale up configurations and federated database configurations to determine the scalability requirements.

Sizing and Capacity Planning

The application deployment configuration should specify the logical and physical aspects of deployment. However, to determine the resources required currently and those required for future growth, you must supplement this information with appropriate sizing and capacity planning data. The capacity planning for the commerce architecture must be done based on the following steps:

12. Size the data collection.
13. Use sizing calculations to provide a theoretical base for the capacity planning.
14. Complete performance benchmarking of the system by using transactions that are indicative of the projected load. You will need to perform this benchmarking multiple times during the construction of the system to collect performance data and to benchmark the resource usage. This data then can be re-applied to the sizing data calculations to improve capacity planning. While theoretical calculations based on historical data and specific data from general benchmarks are very effective, you should also perform benchmarking for the specific context.

You should perform separate sizing calculations for the Web, application, and database tiers, based on their respective parameters. On the Web tier, the calculations must be based on guidance and performance benchmarks related to IIS, SharePoint, and Commerce Server. On the application tier, the calculations must be based on Commerce Server guidance. On the database tier, the calculations must be based on SQL Server guidance.

Table 7 provides a list of factors and decision points that influence sizing and capacity planning. While this list, which is based on building e-commerce systems, is useful, it is important to review it in the context of your solution.

Table 7: Sizing and Capacity Planning Influencing Factors and Decision Points

	Influencing factors	Decision points
Web tier	Number of requests. Number of concurrent requests. Average page size. Request execution time (from performance benchmarks). Resource use for indicative scenarios (from performance benchmarks). Volume of content to be stored in file system. Peak usage projections. Failover and availability specifications.	Number of processors. Number of servers. Memory per server. Storage type and configuration. Storage capacity. Connectivity to other tiers. Failover and standby configuration.
Application tier	Number of requests. Number of concurrent requests. Average request and response size. Request execution time (from performance benchmarks). Resource usage for indicative requests (from performance benchmarks). Peak usage projections.	Number of processors. Number of application servers. Memory per server. Connectivity to other tiers. Failover and standby configuration.

	Failover and availability specifications.	
Database tier	Number of requests.	Number of processors.
	Number of concurrent requests.	Number of SQL Server computers.
	Request execution time (from performance benchmarks).	Memory per computer.
	Resource usage for indicative requests (from performance benchmarks).	Storage type and configuration.
	Peak usage projections.	Storage capacity.
		Connectivity to other tiers.
		Failover and standby configuration – cluster vs. mirror.

Refer to the following documents when performing size and capacity planning for Commerce Server 2009 architectures:

- [Plan for Performance and Capacity \(Office SharePoint Server \)](http://technet.microsoft.com/en-us/library/cc262971.aspx) (<http://technet.microsoft.com/en-us/library/cc262971.aspx>)
- [Microsoft Commerce Server 2009 Default SharePoint Site Performance Guide](http://www.microsoft.com/downloads/details.aspx?FamilyID=61fac357-9df5-4324-b17d-bc6af9d980a4&displaylang=en) (<http://www.microsoft.com/downloads/details.aspx?FamilyID=61fac357-9df5-4324-b17d-bc6af9d980a4&displaylang=en>)
- [Microsoft BizTalk Performance Optimization Guide](http://msdn.microsoft.com/en-us/library/cc558617(BTS.10).aspx) ([http://msdn.microsoft.com/en-us/library/cc558617\(BTS.10\).aspx](http://msdn.microsoft.com/en-us/library/cc558617(BTS.10).aspx))

System Architecture

This section describes a system architecture that addresses the functional requirements of an e-commerce system and the considerations identified in the architecture frame in [Table 2](#). The system architecture is illustrated and explained using different views that highlight key aspects of the proposed system.

- The conceptual view represents the high level components of the system without emphasis on the system design. The system as seen from outside is represented here.
- The logical view is a mapping of the required functionality into modules. It serves as validation that, at a high level, all key required functionality is included in the solution.
- The development view describes the different layers and the components of the application, along with an indication of the software stack that will be used for these elements. This view ensures that considerations for maintainability and future scalability are taken into account, and that the functionality is mapped from the logical view with necessary links to the software implementation. This view maps the application view into the actual software components that will be used to realize the system architecture.
- The deployment view shows how the system will be deployed physically. This view maps the software components in the software view onto physical nodes, and depicts the communication between these nodes.

Reference Implementation

The architecture views, description, and design templates explained in subsequent sections of this document are based on a reference digital marketing and e-commerce solution that is built with Commerce Server 2009. Commerce Server 2009 comes with a sample implementation that uses SharePoint Commerce Services. The reference implementation is built on SharePoint Commerce Services, and contains necessary customizations and additions for payment processing and user experience. Integration with Dynamics AX using BizTalk Server is provided for back-end data processing. To highlight the best practices for using Commerce Server as a platform, an ASP.NET front end was developed as an alternate Web site or channel that uses the Commerce Foundation Services remotely. This was developed by modifying the .NET pet shop front end implementation to connect to Commerce Foundation Services instead of its own back end.

System Architecture – Conceptual View

This view of the architecture illustrates all subsystems, actors, and external integration points. Figure 7 provides a conceptual view of a typical e-commerce system based on Commerce Server.

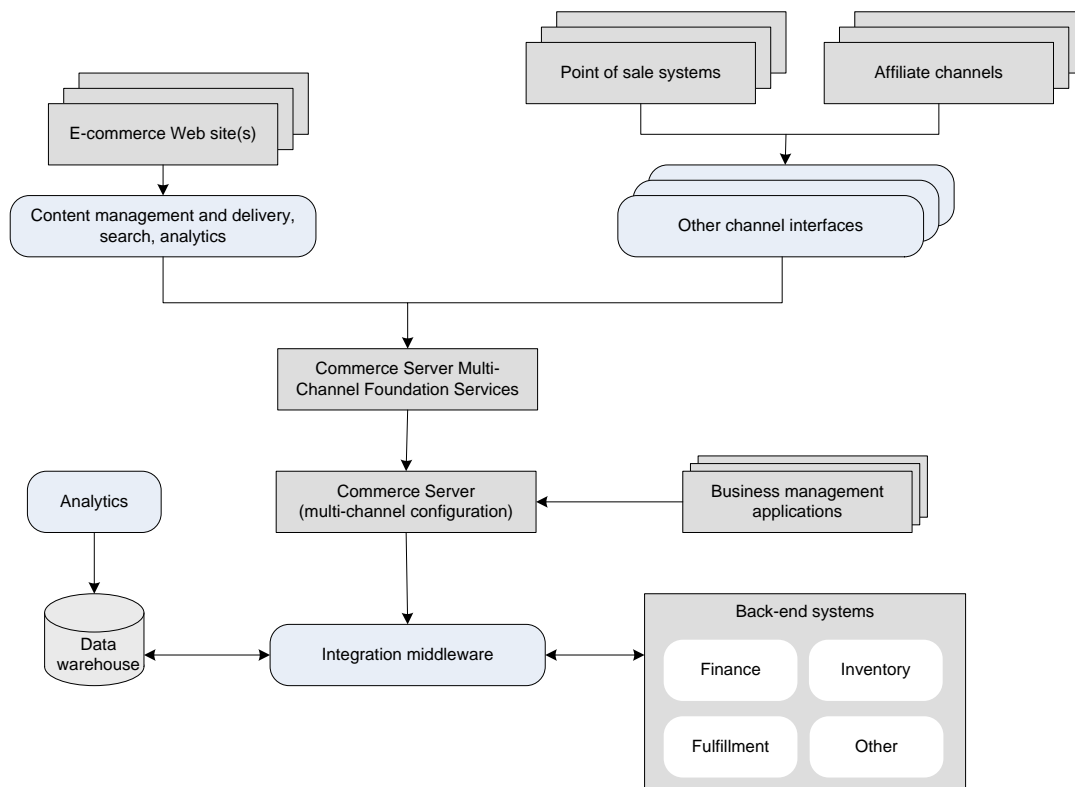


Figure 7. Conceptual view

Content Management and Delivery

The content management and delivery platform provides the basic framework for delivering digital content in a business to consumer or business to business environment. This includes document management and publishing, Web content management and publishing, and management and publishing of rich media. The platform provides features required for creating, editing, and uploading content, along with customizable workflows for approval and publishing. The platform supports both staging and production deployment configurations to provide appropriate isolation of

production and editable data. The platform is scalable and facilitates hosting of a collection of Web sites with a well-defined hierarchy.

Multiple Channel Interfaces

Apart from the main digital marketing Web site(s) of the business, there could be other channels of content delivery and e-commerce. These could include retail channel systems, affiliate systems, partner interfaces, and so on, and could also facilitate content delivery and incoming transactions. These additional channels can be hosted or run as separate instances that are connected to the core commerce engine and leverage the Commerce Server 2009 Multi-Channel Foundation Services.

Multi-Channel Foundation Services

The Multi-Channel Foundation Services provide a channel-aware façade layer around the core commerce engine to service context-specific data, as well as perform channel-specific processing of transactions, thus enabling the centralized commerce engine to serve multiple channels based on context. The foundation services follow a loosely coupled interface pattern that does not require the consuming channel to be tightly integrated with the underlying commerce data model. The request response model employed enables generic interfaces that can work with different request and response instances, as long as the right components are configured in the foundation services operation sequences.

Commerce Engine

The commerce engine is the heart of Commerce Server 2009, and provides management, repository, and run-time capabilities for the core e-commerce areas: product catalogs, user profiles, orders, and marketing campaigns. The commerce engine provides a scalable and configurable runtime that can process e-commerce-specific transactions by using prebuilt components. The commerce engine has its own database repositories that can store and process data specific to the commerce implementation.

The commerce engine is programmable and can be enhanced by plug-in components to do business specific processing and interfacing.

Business Management Applications

The business management applications provide a user interface that allows business personnel to manage data and functionality configured within the commerce engine. This set of applications allow users to define and manage catalogs, maintain user profiles, track orders, and create, run and track targeted campaigns.

Integration Middleware

The integration middleware abstracts the interaction required between the core e-commerce system and other line-of-business systems. The line-of-business systems could have interfaces ranging from legacy to state of the art service-oriented integration. The integration middleware provides the routing, translation, transformation, and orchestration services required for end-to-end processing, from shopping to delivery.

Data Warehouse

The data warehouse is the central repository that collects data related to products, transactions, users, and campaigns, across multiple channels. The data warehouse collects and stores data from multiple sources—in addition to other databases, data can be collected from Web logs, exception logs, and other sources that provide data related to users or transactions. Collection and categorization of data is done in the data warehouse in preparation for analytics. The data warehouse must be implemented with sufficient data integration points across the system, primarily by importing files and database jobs.

Data Analytics

The data analytics system uses the data collected in the data warehouse to produce multiple complex views of the system, including aggregated and multi-dimensional data representations. Based on these views, various analytics reports can be delivered via the reporting front end.

Web Analytics

The Web analytics component facilitates the collection and analysis of data related to user browsing on the Web site. At a minimum, it works with browse logs on the server to identify page hit volume, frequency, and patterns. At the next level, it is possible to log, track, aggregate, monitor, and report events specific to marketing and commerce activities. For example, the component can track ad clicks, which products are added to carts, and purchase information for a user or group of users, and then analyze the data.

Search

Search is an integral part of an e-commerce system. A customer's ability to find information and/or a specific product is central to the success of an online business. In addition to looking for product information, users search for related information that can help in decision making. It is important for the search system to provide an integrated experience to the user, and present results from all relevant sources, including the commerce database, content repository, and any related systems. For example, a user looking for information on a product might like to see product images, descriptions, reviews, the product manual, warranty information, information from social forums, and so on.

Back-end Systems

The digital Web and e-commerce system will have to interface with one or more back-end line-of-business systems that act as repositories for data, as well as provide functionality such as inventory management, order fulfillment, and payment processing. These systems are the core of the enterprise, and typically run on heterogeneous technologies. It is important to enable standard ways of communicating with these systems. This will involve configuration and/or building code in the commerce system to consume the services of the back-end systems, as well as (in some cases) making changes to the back-end systems to provide access points for the commerce system.

System Architecture – Logical View

The logical view shown in Figure 8 illustrates the next-level functional components of the system and their placement in relation to each other. The illustration also shows the dependencies across the high-level components.

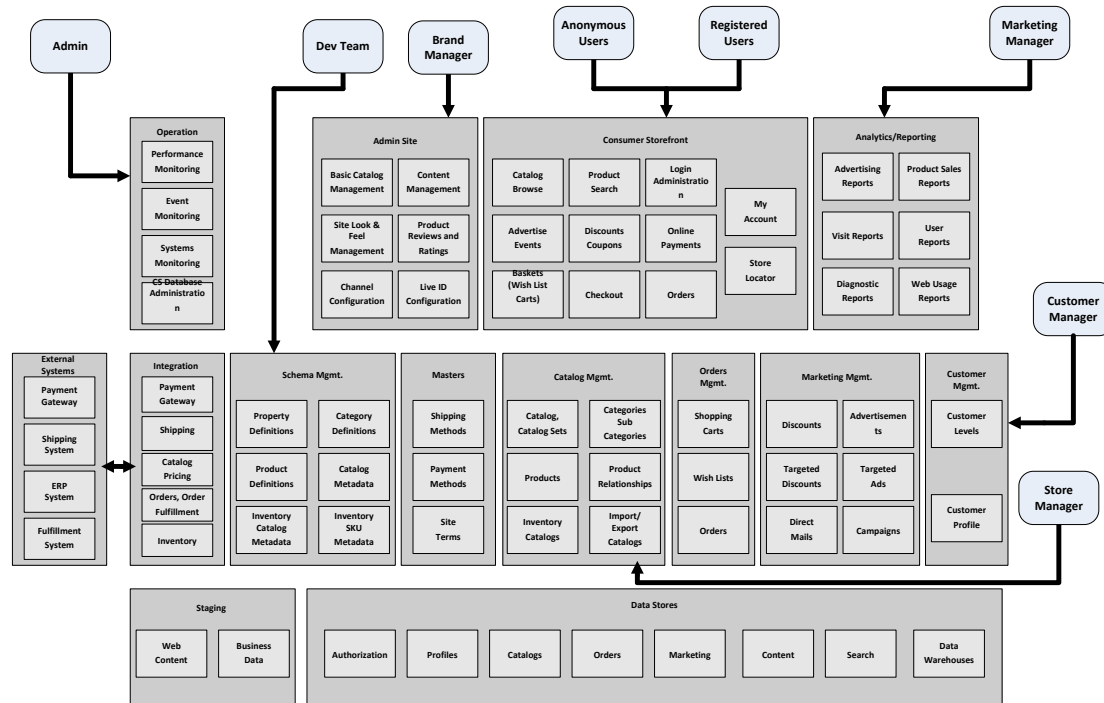


Figure 8. Logical view

Consumer Store Front

The consumer store front is the key delivery access point for content, data, and transactions. This is the Web site front-end accessed by consumers for finding information, updating reviews, completing purchases, and making recommendations. Essentially, this is the Web real estate available to the enterprise, and allows the business to engage with the consumer and provide them with more information and guidance that enables better brand understanding and revenue generation.

The key capabilities of the commerce system should be fully available from this access point, whether these capabilities are focused on content management, search, transactions, or integration.

Catalog Browse

This functional block allows the customer to browse the product catalog in multiple ways. Users can navigate to specific product information by product category, brand, or other classifications relevant to the business context. This block is implemented as a composition of UI and business components. The key focus for the UI components is to present the information in a user friendly way, and the business components implement the logic needed to retrieve the requested information by using the multi-channel foundation services of Commerce Server.

The UI components are implemented as Web parts and can use the different ASP.NET technologies, including Ajax and Silverlight, to provide an intuitive, rich, and secure user interface. The additional

logic is implemented as .NET components that interface with the foundation services. This modular design allows you to manage incremental updates and enhancements more easily.

Enabling the Commerce Server cache can help to reduce round trips to the repository for catalog information. Additional custom caching at the application level can be used for data that changes less frequently and that requires complex querying of the catalog itself.

Figure 9 is a representative class diagram and Figure 10 illustrates the call sequence.

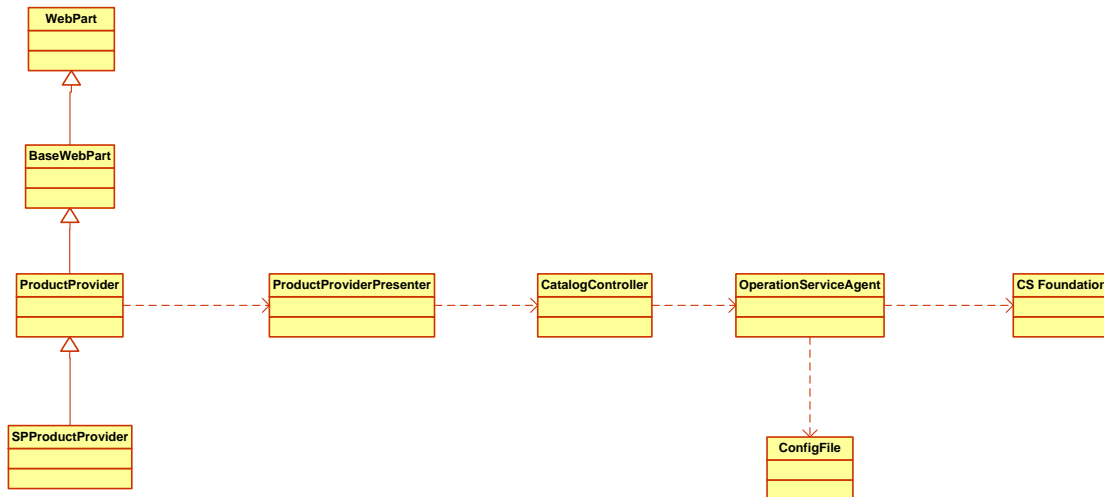


Figure 9. Class diagram

The catalog information is fetched from the Commerce Server repository and displayed on the screen by using a Model-View-Presenter (MVP) pattern. The **ProductProvider** Web Part (View) uses the **ProductProviderPresenter** (Presenter) class for presentation. The **ProductProviderPresenter** class uses the **CatalogController** to fetch data from the Commerce Server Repository (Model) by using the Commerce Foundation Services. The **OperationServiceAgent** is the client-side component that facilitates calls to the Commerce Foundation Services. The controller uses **OperationServiceAgent** and fetches the data from the model. The **ProductProvider** inherits from a base Web Part, which provides for common features. The **SPPProductProvider** extends this view and implements SharePoint-specific interfaces.

Using the MVP pattern permits a clear separation of concerns in the implementation. The provider and presenters are UI-specific and focus on display of the data. The controller takes care of data fetching and processing.

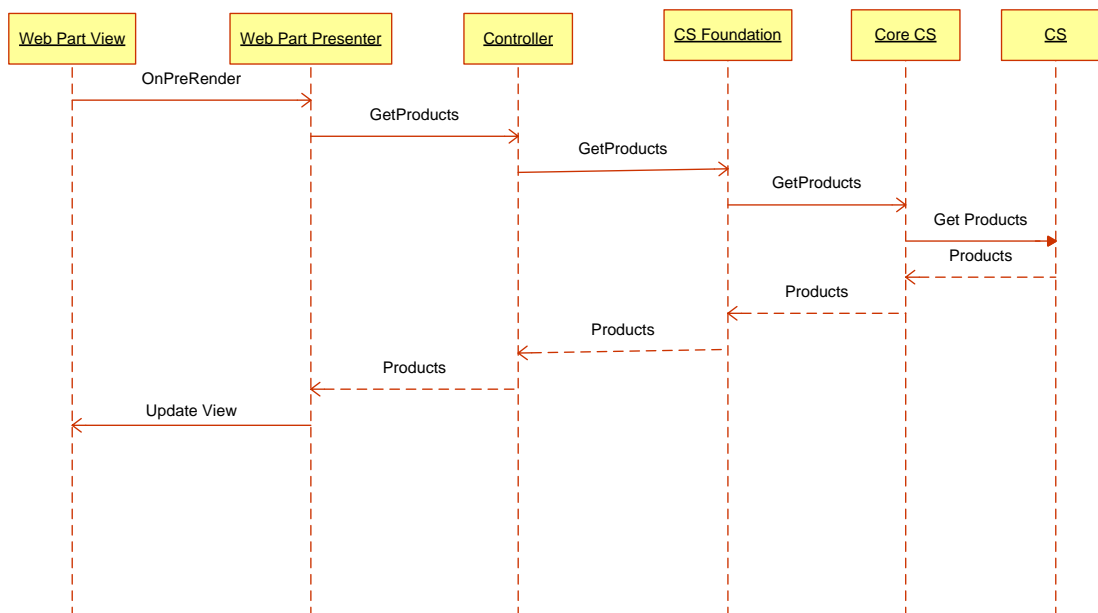


Figure 10. Call sequence

The call sequence illustrated in Figure 10, above, represents the basic flow to fetch data into the view. The calls that go from the controller to Commerce Foundation Services are routed by the **OperationServiceAgent**, based on the deployment configuration: either as in-process or WCF remote calls. This helps to allow the same code to service both single-tier and distributed deployments.

The Commerce Foundation Services provide channel-specific operation sequence configurations. These invoke processing logic that is channel-specific based on incoming context information. Every call to the Commerce Foundation Services can pass context information including channel and user data.

The Commerce Foundation Services provide a set of standard operations that can be configured as part of operation sequences. Apart from these, custom operations can be added as needed. Standard operations are also provided for invoking the Commerce Server Core functionality. The core functionality provides the core APIs and Commerce Server pipeline configurations for content selection and transactions.

Alternate call sequence flows could be executed, depending on caching policies. For example, the data might be cached by the presenter or controller to reduce the round trip. The Commerce Server Core engine can be configured to cache data in memory to reduce database access.

SharePoint Commerce Services Web Parts for Catalog Browsing

SharePoint Commerce Services for Commerce Server 2009 provide the following Web Parts for catalog browsing:

- Product Query Web Part – Displays the products and variants from a catalog.
- Product Details Web Part – Displays the different products and product details.
- Images Viewer Web Part – Displays different product, catalog, and other images.
- Site Map – Displays the map of the site.

For more information about SharePoint commerce services, see [http://msdn.microsoft.com/en-us/library/dd328362\(Commerce Server.90\).aspx](http://msdn.microsoft.com/en-us/library/dd328362(Commerce Server.90).aspx)

Product Search

User experience is an important part of this functional block. Commerce Server provides catalog search capability. The searches can be based on keywords or on more advanced criteria based on parameters. In addition, SharePoint provides the facility to search the content on the Web site. The product search functionality can combine these two features to provide the user with an integrated search experience. The search is displayed in a Web Part, which invokes the respective searches by using the SharePoint API and Commerce Server foundation services, and then displays the results to the user.

To leverage search functionality to the fullest, you will need to configure SharePoint and Commerce Server search services as explained in the corresponding product documentation.

To enable SharePoint and Commerce Server search capabilities, you need to configure the following Web Parts:

- Search Box Web Part – Displays a search box that can be configured to search both the e-commerce site and any static content.
- Search Paging Web Part and Search Results Web Part – Displays the results of the search in a two-column pane, allowing shoppers to easily see the products or content on your site, and easily page through a large number of results.

Logon and Registration

This block provides the key functionality for managing and verifying user identity. The components are built on the SharePoint and ASP.NET security infrastructure, with necessary customizations to handle user data specific to the business context.

UI controls are built as Web Parts for the registration forms and are hosted on SharePoint pages of the front-end site. The business components perform the necessary validations for registration, and then pass the information to the Multi-Channel Foundation Services for processing and storage. The user information is stored in the Commerce Server Profiles System and can be managed by the Commerce Server Manager user interface.

The store front uses forms authentication for user logon verification. (The SharePoint site is extended for forms authentication.) A custom membership provider that integrates with the Commerce Server Profiles System via the foundation services is used to configure the authentication mechanism. The logon UI controls should be built as Web Parts that invoke the standard forms authentication API for credential verification and session initialization.

To enable logon and registration capabilities, you need to configure the following Web Parts:

- Registration Web Part – Provides the functionality to let users register on the e-commerce site.
- Change Password Web Part and Forgot Password Web Part – Provides functionality for shoppers to reset or change account passwords.

- Live ID – Provides functionality to associate a shopper account with a Microsoft Live ID account, so that the shopper can associate their Live ID account to a particular shopping site.
- Address Book Web Part – Provides the interface to maintain address book.
- Address Detail Web Part – Provides address management functionality, such as adding and editing addresses.
- Credit Cards Web Part – Provides the interface to view and start credit card administrative tasks.
- Credit Card Web Part – Provides the credit card management functionality, such as adding a credit card and credit card information.
- My Profile Web Part – Provides the management functionality for account profile information.

Baskets

This block provides the functionality required to manage the different baskets for the user, including the shopping cart and wish list. Additionally, the user can create and maintain custom lists to manage and track products of specific interest.

Basket management and processing involves a good amount of data processing and calculations to present up-to-date basket information to the user, especially for shopping cart functionality. The Commerce Server Core provides built-in support for basket processing and calculations, including application of coupons and promotion. These calculations use pipeline components that are configurable and extensible, so any modifications should be achieved by appropriate changes and plug-ins to the pipeline configuration and components. For the front-end UI, the shopping cart is rendered as a Web Part that fetches and updates information by using the foundation services. The foundation services, in turn, invoke the Commerce Server Core, which runs the pipeline processes to provide the up-to-date status of the basket.

To enable basket capabilities, you need to configure the following Web Parts:

- Add to Cart Web Part – Enables shoppers to add products to the shopping cart.
- Shopping Cart Web Part – Displays all the merchandise that the shoppers want to purchase.
- MiniCart Web Part – Displays a summary, usually in the upper-right corner of the page, of the total items and amount in a shopper's basket.
- My Wish Lists Web Part – Provides shoppers with a way to manage all of their lists.
- Wish List Detail Web Part – Displays the details of a shopper's wish list.

Checkout

This functional block implements the complete process checkout process. The checkout process consists of multiple steps that involve specifying the shipping and billing addresses, shipping preferences, payment options, and payment details, and confirming the purchase. This sequence takes the user through multiple pages, each of which is implemented as one or more Web Parts. All

the pages require the corresponding Web Part to invoke the Commerce Foundation Services to perform a data fetch or write from the Commerce Server Core. Some of these steps, such as shipping charge calculation or applying gift coupons, require basket re-processing to determine the final purchase price. In these cases, the foundation service must invoke the appropriate pipeline and return the results to the front end.

To enable checkout capabilities, you need to configure the Checkout Web Part, which manages the checkout process. This Web Part is responsible for the flow of checkout operations and for providing a uniform navigation model. This checkout process might include address and credit card management actions.

Orders

A basket is converted to an order when a purchase is completed. The Order subsystem provides functionality to store, track, retrieve, and manage orders. Commerce Server provides built-in functionality for storing and tracking orders. The Commerce Server Core pipelines can do the necessary processing when a basket is converted to an order. The Commerce Server storage provides for order profile storage and the Order Manager interface provides the UI for managing orders from the back end.

For the store front, Web Parts must be created to display order related information to a user. These Web Parts, in turn, call the foundation services to fetch the order information. The BizTalk Commerce adapters synchronize order data to and from the back-end systems in a loosely coupled fashion.

To enable order processing, you need to configure the following Web Parts:

- Order Details Web Part – Displays a read-only version of the shopping cart and checkout information.
- Order History Web Part – Displays previous completed orders, including order status.

Online Payments

The e-commerce system design supports multiple payment options for the user, including online payments. The checkout process permits a user to choose one or more payment options. These options are then used to process the payments after the user confirms the order. The payment functional block has the following components:

- A UI for selecting the payment option.
- Integration at the presentation and business logic level with the chosen payment system. For example, the presentation layer might perform a set of redirections for the user to connect to the payment system directly, and complete the payment, or the business logic layer might call a service from the payment provider to perform a specific financial transaction.
- Processing of payment information, including:
 - Updating order and related records.
 - Updating user-related records (such as reward points).

- Recalculating pending totals.

The UI components are implemented as Web Parts embedded in the payment-related pages. For online payments, the presentation layer has navigation logic to take the user to a payment gateway interface and then return the user after the payment is completed.

The business logic is implemented as a set of .NET components that perform the validations and processing of result values from the payment gateway.

The Commerce Server pipelines provide a built-in framework for core payment processing and calculations. As part of development and deployment, you will need to update the framework to include any custom logic required by the payment options that you wish to include.

To enable online payments, you need to configure the Checkout Web Part, which manages the checkout process. It is responsible for the flow of checkout operations and for providing a uniform navigation model. The checkout process might include address and credit card management actions.

Campaigns

Campaigns cover the functionality for delivering advertisements, promotions, and discounts according to the configuration in the Commerce Server Core. Typically, campaigns involve:

- Display of advertisements at pre-defined spots according to defined rotation logic
- Display of promotion and discount information at pre-defined spots and in the related product context
- Applying the discounts and promotions to baskets and purchases
- Tracking advertisement views
- Tracking promotion and discount usage

The UI part for campaigns is implemented as independent Web Parts that are embedded in product pages, as well as specific logic in some of the main Web Parts for displaying the promotion information. The Web Parts use the foundation services to obtain the right information to be displayed and render the content accordingly. The Commerce Foundation Services use the core components, such as the Content Selection Pipeline, to deliver the correct ad and promotion content, based on context information from the Web Parts.

To apply promotion and discount values and percentages to a basket, you must configure the basket-related pipelines. Include any required custom logic as additional pipeline components or as script changes to existing components.

Front-end events, such as ad clicks, are logged in the IIS logs for tracking purposes, along with data logs based on pipeline events. This data is aggregated later in the data warehouse and used for tracking and analytical reporting. See the [Analytics and Reporting Architecture](#) section for details.

Figure 11 provides an overview of the Content Selection Framework that manages campaign elements such as discounts and advertisements.

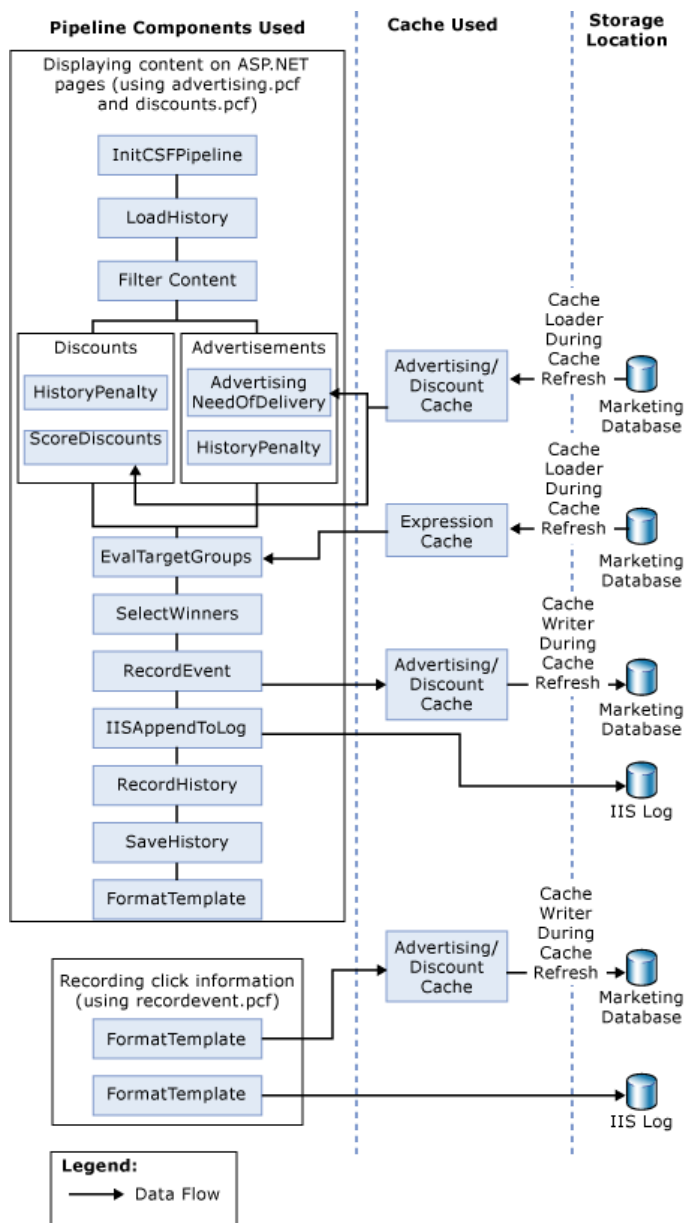


Figure 11. Overview of Content Selection Framework

The Content Selection Framework is executed as part of the Commerce Server Core services. This is invoked by the respective operation sequence from the Commerce Foundation Services. The Content Selection Framework uses the Commerce Server pipeline infrastructure to configure and execute the components that select, filter, and process the data content to be returned for display. This loosely coupled sequencing allows you to arrange components to meet application requirements for content filtering, targeting, and tracking. For more information, see the [Commerce Server Content Selection Framework Architecture](http://msdn.microsoft.com/en-us/library/dd464541(CS.90).aspx) on MSDN ([http://msdn.microsoft.com/en-us/library/dd464541\(CS.90\).aspx](http://msdn.microsoft.com/en-us/library/dd464541(CS.90).aspx))

To enable campaign processing, you need to configure the Discount Ad Web Part, which displays advertisements and discounts that are relevant to both the current shopper and current site context. This Web Part works in conjunction with Marketing Manager and the various campaigns that are set up.

Integration

Integration is one of the key components of this architecture. The e-commerce system needs to interface with multiple back-end and external systems to send and receive customer, order, catalog, and inventory data. The integration block implements the features needed for this integration, and can also be used by the other blocks as required.

The integration block implements proxies and access points as to meet integration requirements. For direct service-based integrations, the integration layer uses a configuration and proxy-based pattern that invokes the services directly. For integration that requires more complex transformation, orchestration, and asynchronous processing BizTalk provides the necessary framework and components. The integration layer abstracts this from the rest of the application so that the appropriate integration mechanism can be selected based on access and endpoints without affecting the rest of the application.

Depending on the enterprise context, this pattern or variations of this pattern can be applied. Some enterprises use separate integration middleware or an enterprise service bus based on BizTalk Server to provide integration services. In this type of architecture the integration block functionality will be limited to the processing required to call the middleware.

Windows Communication Foundation (WCF) technology provides rich features for integration in a secure, interoperable way. For these reasons, WCF services and proxies should be the primary choice for the integration block. Secondary technologies to consider include .asmx files, HTTP, or ftp.

Admin Site

An architecture that is based on Commerce Server 2009 provides advanced content management capabilities that leverage SharePoint features. The admin site is a securely configured separate access point for managing content displayed on the Store front. The store front site and the admin site both access the same content database, but can be configured for different authentication schemes. The admin site is used by authorized users from the retail organization who are responsible for creating and approving content. The admin site is secured by Active Directory-based authentication, and the store front site is created as an extended site with forms authentication.

The admin site facilitates the management activities described in the following sections.

Site Navigation and Usability Management

SharePoint's built-in site management features can be used to create and maintain site hierarchy, page layouts, and other features related to the user experience. The administration and content creation interfaces for SharePoint provide a rich facility to create and manage sites and applications for the front end.

Content Management

You can use SharePoint's built-in content management features to create, manage, and publish content. SharePoint Commerce Services for Commerce Server 2009 provide a framework in which content from SharePoint can be combined with Commerce Server catalog content and presented to the user.

Product Reviews and Ratings

Review and rating functionality is built in as a Web Part that facilitates customer reviews and then manages the review content in the SharePoint repository. The Web Part retrieves the review content based on the product context, and displays the information in the appropriate pages.

Basic Catalog Management

SharePoint Commerce Services provide a set of Web Parts that can be used to write updates to the Commerce Server catalog. This provides an integrated way to manage both the SharePoint content and the catalog content from Commerce Server.

Channel Configuration

This feature allows you to assign a *channel name* to a specific e-commerce site. A channel is any Web site that is targeted at a specific segment or device. The channel name is used throughout Commerce Server 2009 so that business and presentation logic can be tailored to address the specific needs of that channel.

Live ID Configuration

You will need to configure a Web Part to support Live ID-based authentication. This configuration permits users who have Live IDs to register and access the site by using their Live IDs. .

SharePoint Commerce Services Web Parts for Administration

SharePoint Commerce Services for Commerce Server 2009 provide the following configurable Web Parts for administration:

- Channel Configuration Web Part – This Web Part is a SharePoint list that allows you to add a channel name to the site. This is implemented as a custom list.
- Inline Product Editor Web Part – This Web Part lets content creators and administrators update specific product information on the site directly, providing a what-you-see-is-what-you-get (WYSIWYG) experience. This allows authorized users to quickly edit descriptions and titles, or any other attribute information. These changes are reflected back to the master catalog in Commerce Server 2009.
- Simple Property List Web Part – This is a utility Web Part, and can be used indirectly as part of other Web Parts, and can also be used to enable the selection of various properties from the Commerce Server Core systems.
- Product Provider Web Part – This is a utility Web Part that can be used indirectly as part of other Web Parts, and can be used by other Web Parts to marshal data calls to Commerce Server and return product information.

Catalog Management

The catalog management functional block enables an authorized user to manage and update catalog data. The functionality is implemented as a smart client desktop application. The application uses Web services published by the Commerce Server Core, and provide an integrated interface to manage all data and configuration related to catalogs in the store. This interface allows users to

manage categories, products, product relations, inventory information, and multiple types of catalogs.

The out-of-the-box version provides most of the common functions required and can be extended programmatically if necessary.

For more information about Commerce Server Catalog Manager, see [What Are the Business Management Applications?](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx) on MSDN ([http://msdn.microsoft.com/en-us/library/dd328708\(CS.90\).aspx](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx)).

Order Management

The order management interface enables an authorized user to view and manage information related to shopping carts, wish lists, and orders. This is another extensible smart client desktop application that uses the Commerce Server Core Web services.

For more information about Commerce Server Order Manager, see [What Are the Business Management Applications?](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx) on MSDN ([http://msdn.microsoft.com/en-us/library/dd328708\(CS.90\).aspx](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx)).

Marketing Management

This is another smart client application that provides comprehensive features for creating and running advertisements, discounts, and promotions for the store. It also provides features to combine multiple offers in a campaign for simpler management and tracking.

Targeting based on profile properties is one of the key features that can be used across multiple content delivery and communication scenarios.

For more information about Commerce Server Marketing Manager, see [What Are the Business Management Applications?](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx) on MSDN ([http://msdn.microsoft.com/en-us/library/dd328708\(CS.90\).aspx](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx)).

Customer Management

The customer management provides features for reviewing and updating customer profile, basket, and order information. This is also a smart client application that calls the Commerce Core Web services.

For more information about Commerce Server customer management, see [What Are the Business Management Applications?](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx) on MSDN ([http://msdn.microsoft.com/en-us/library/dd328708\(CS.90\).aspx](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx)).

Schema and Master Management

Commerce Server provides a flexible and extensible way to define the schema of the data that is to be stored. For example, the Commerce Server catalog can be configured to manage different varieties of products with different properties without any custom code or database changes. Internally, Commerce Server schemas maintain a representation of the product data and are used by the core engine to present and store data.

The schema and master management interface allows you to define and maintain schemas, and is another smart client application that uses Commerce Server Core Web services.

For more information about Commerce Server schema and master management, see [What Are the Business Management Applications?](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx) on MSDN ([http://msdn.microsoft.com/en-us/library/dd328708\(CS.90\).aspx](http://msdn.microsoft.com/en-us/library/dd328708(CS.90).aspx)).

Analytics and Reporting Architecture

Figure 12 shows the architecture of the analytic and reporting services.

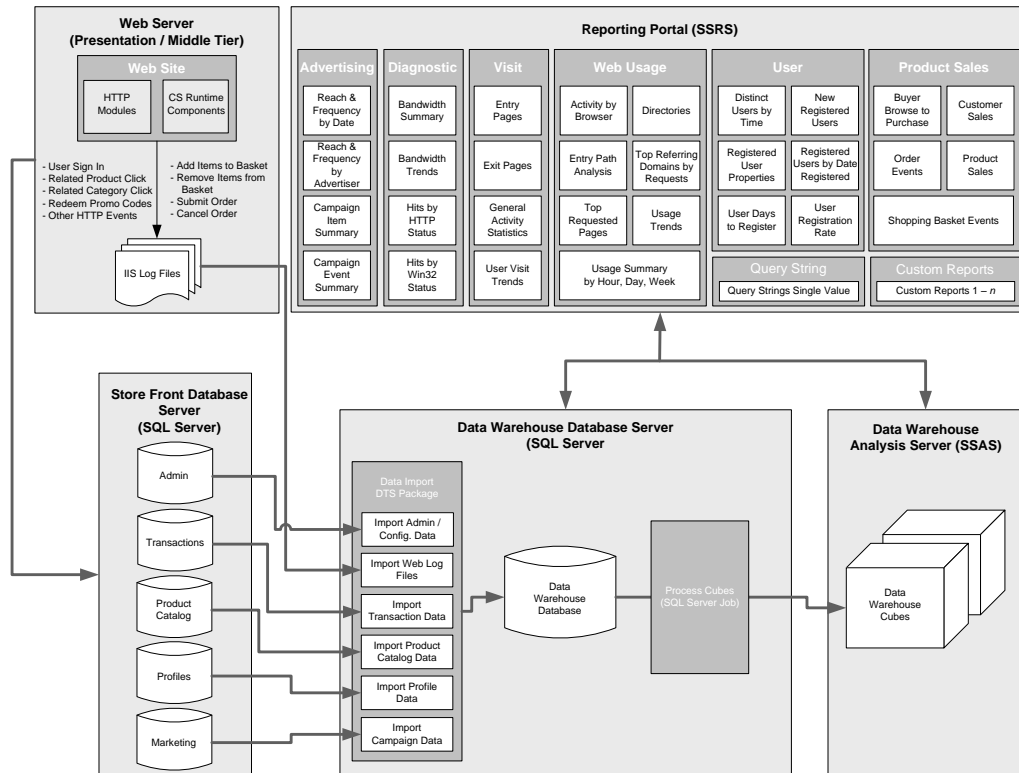


Figure 12. Analytic and reporting architecture

Commerce Server 2009 provides a well-defined reporting architecture that can be used out of the box. The key components are:

- Data import packages that collect data from multiple sources, including Web log files and transaction databases
- Data warehouse schema
- Cube processing jobs
- Analysis cubes
- SSRS reporting portal
- Report definitions for SSRS

As part of the implementation, Commerce Server 2009 reporting components should be deployed and customized as follows:

15. If necessary, modify and add to the data import packages so that they collect additional context-specific data.
16. Design new cubes (if needed).
17. Add or modify report definitions.

Back-end Integration Architecture

The e-commerce system must interface with back-end systems to synchronize data and processes, such as catalog details, profiles, and order processing. This paper describes a real world integration of an e-commerce site with a Microsoft Dynamics AX system. The Microsoft Dynamics AX system is set up as the back-end system, and takes care of basic catalog information and order processing. This requires an appropriate real-time synchronization of data across the two systems via appropriate integration mechanisms.

BizTalk Server provides adapters for integration with Microsoft Dynamics AX. (It also provides adapters for other systems, such as those from SAP). Commerce Server 2009 has commerce adapters that can work with BizTalk. BizTalk provides message definition, mapping, and orchestration capabilities, as well as interfacing with a varied set of interface points and protocols.

A BizTalk-based integration architecture that leverages the adapters and out-of-the-box orchestration and scalability features is illustrated in Figures 13, 14, and 15.

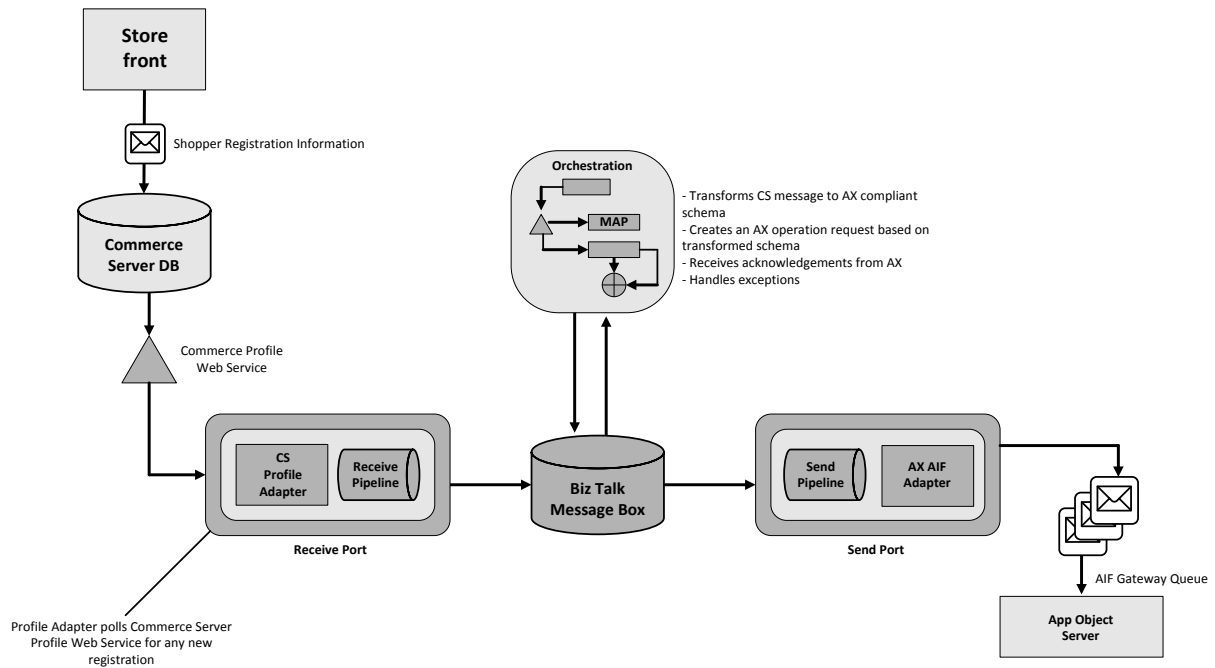


Figure 13. BizTalk-based integration architecture – profile integration

The example shown in Figure 13 illustrates the profile data integration design. Customer profiles are created and updated in the Commerce Server repository based on user registration and profile updates on the Web site. This profile data is to be synchronized with the back-end system (in this case, Dynamics AX ERP) for proper transaction processing by the back end.

The design uses the Commerce Server profile adapter for BizTalk. This is configured at the receive port and mapped to an orchestration. The Commerce Server profile adapter looks for additions and changes to the profile store and invokes the orchestration. The orchestration is configured to process and map the data to the form as required by the Dynamics AX adapter for BizTalk. The Dynamics AX adapter for BizTalk is configured at the send port and will take care of communication and data exchange with Dynamics AX.

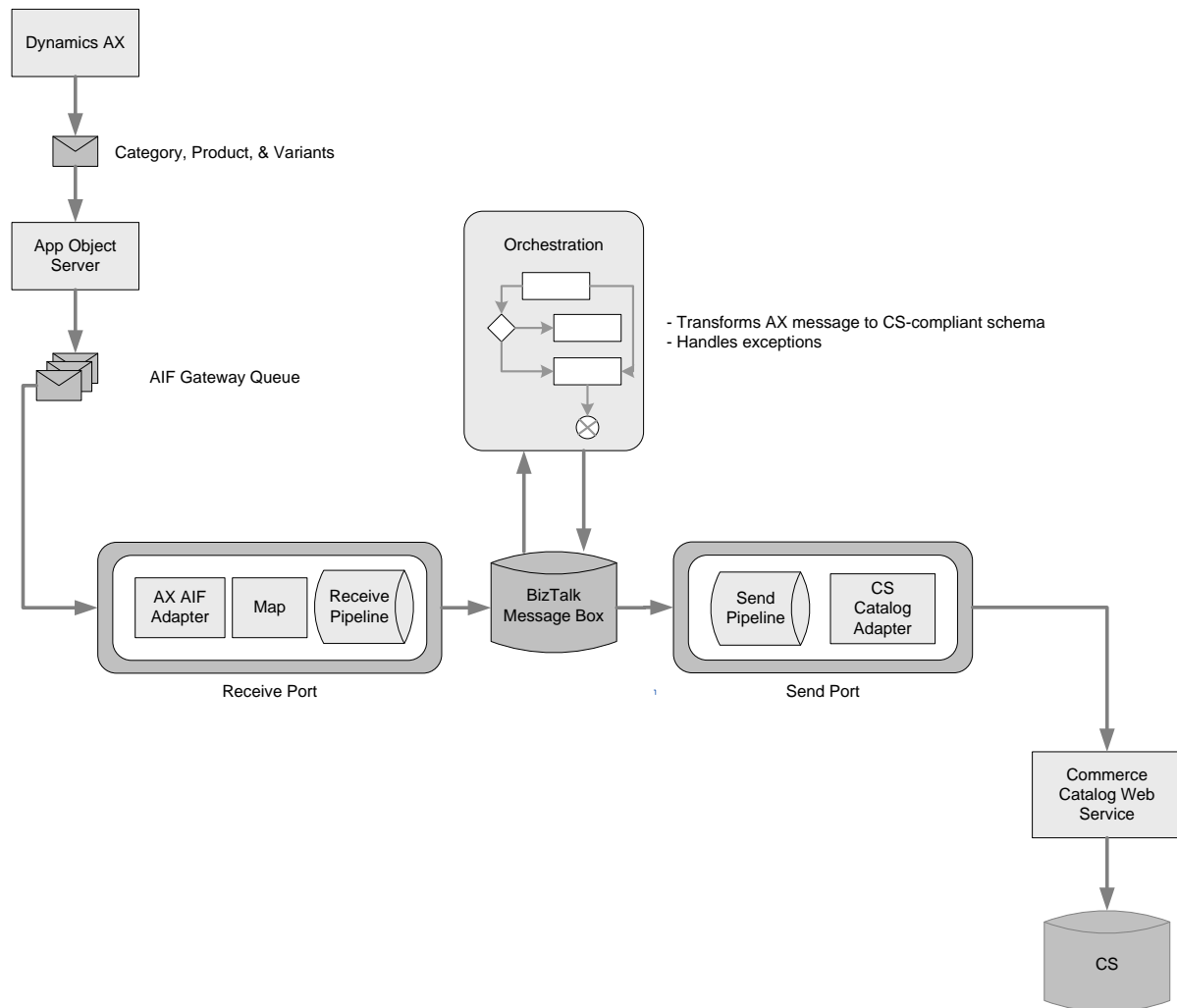


Figure 14. BizTalk-based integration architecture – catalog integration

The back-end Dynamics AX is the designated master system with respect to the base catalog data. This data needs to be updated in the Commerce Server database periodically or on a real-time basis. The design illustrated in Figure 14 uses the Commerce Server and Dynamics AX adapters for BizTalk to synchronize catalog data from Dynamics AX to Commerce Server. After an update of catalog information, the data is written into the Application Integration Framework (AIF) queue. The Dynamics AX adapter gets notified and reads the information to invoke the orchestration processing in BizTalk. The orchestration has the Commerce Server send adapter configured as the send port and writes the information into the Commerce Server repository.

This design allows for changes in the Commerce Server repository to be reflected almost in real time. The staging environment should be configured to receive these updates. The catalog content can be published to the production environment after a proper review on the staging environment.

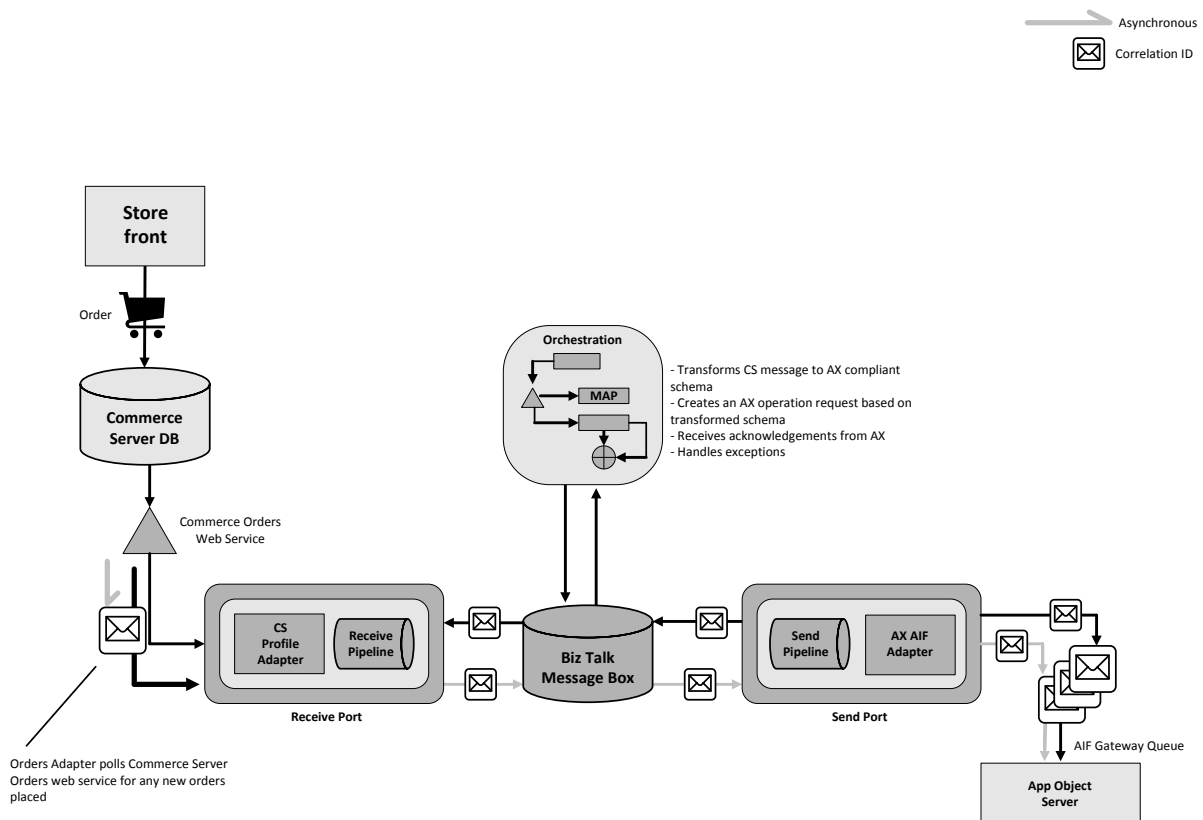


Figure 15. BizTalk-based integration architecture – order integration

The Commerce Server system receives and processes orders from the different channels, including payment and purchases. Typically, the order is then passed to the back-end systems for further processing to complete fulfillment, shipping, payment reconciliation, and so on. Figure 15 shows the configuration of Commerce Server and Dynamics AX adapters for BizTalk to enable order data synchronization from Commerce Server to Dynamics AX.

The Commerce Server receive adapter polls the Commerce Server repository by using the Commerce Server Web services, and retrieves the orders to invoke an orchestrated processing. The orchestration is configured with a Dynamics AX adapter at the send port. This writes the order information into Dynamics AX (via the AIF queue), which is then stored and processed by the Dynamics AX system.

System Architecture – Development View

The recommended architecture is based on SharePoint and Commerce Server platforms that follow the fundamental ASP.NET architecture. Extensions and new component development in this architecture should follow the best practices for all these core frameworks. Each of these products and platforms has well-defined behavior, boundaries, and tight integration points that can be used to implement this composite architecture.

Figure 16 illustrates the recommended pattern for developing additional Web Parts and components that work across the different application layers.

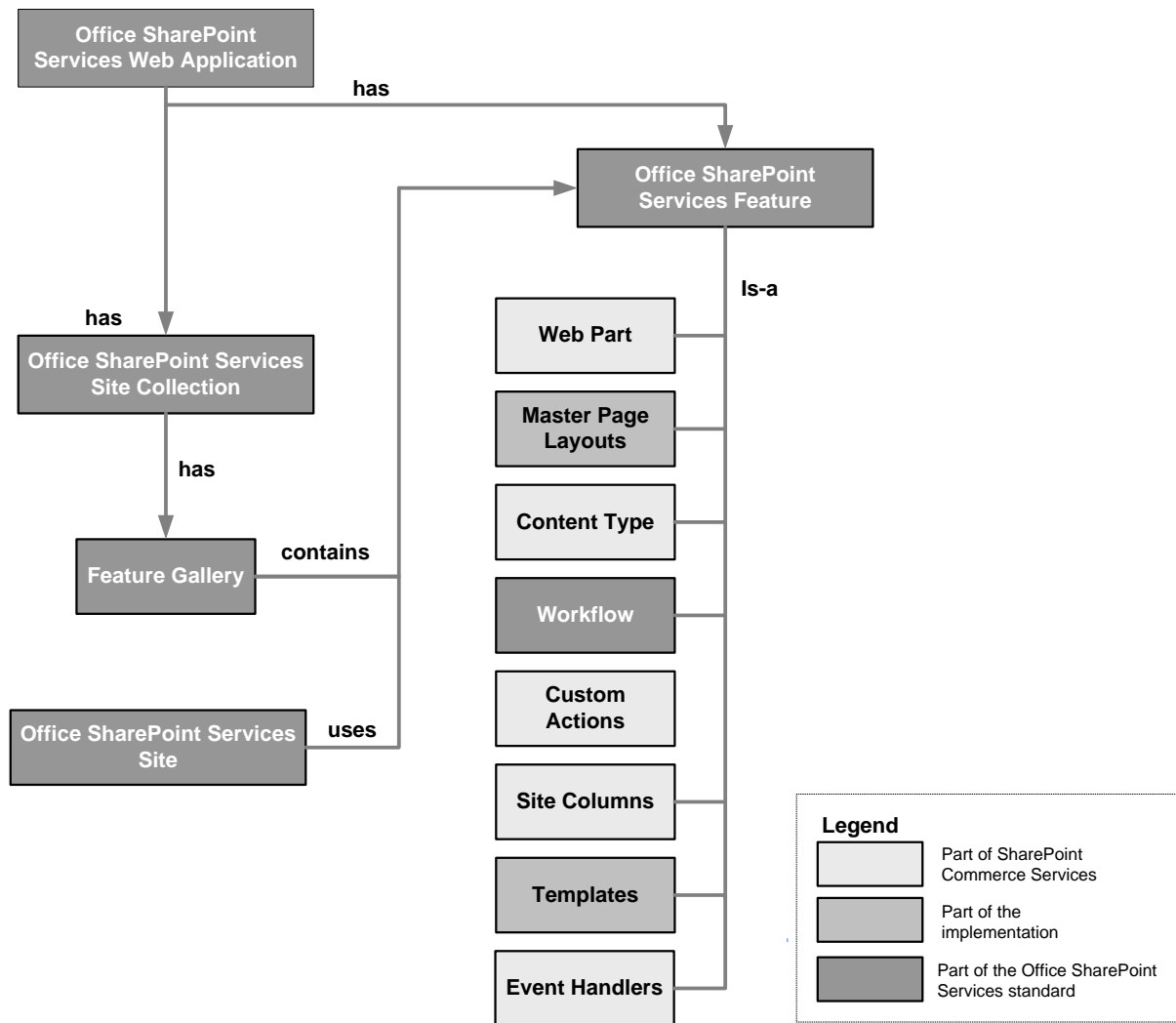


Figure 16. Architectural pattern for Web Part development

Figure 16 illustrates the composition of components that implement the SharePoint site. The modular approach shown enables plug and play of UI components as Web Parts within the SharePoint shell. The SharePoint infrastructure is leveraged for content management, related templates, content storage, and indexing. The transactional components are built as Web Parts that adhere to the layered calling pattern. There is no tight association between a particular dynamic functionality (for example, the displaying the product details) and the page on which it is displayed. The dynamic display is implemented as a Web Part and can be included into any SharePoint page, provided that the Web Part can fit in the layout of the page and all required dependency Web Parts are also included on the page. This allows for a flexible model that facilitates a loosely coupled content layout and display.

The front-end Web site is deployed as a SharePoint site collection in a SharePoint Web application. Depending on the nature and scale of the deployment, a site collection can have its own content database, or it can use the content database attached to the Web application. The site collection has a root site that acts as the main site. Sub-sites for this site could be designed and deployed on an as

needed basis for specific purposes; for example, pages related to a user's account, special promotions, or special store areas for a specific category of products.

The feature gallery of the site collection can have the different features, including Web Parts configured for the site and its sub-sites. This allows a modular and incremental deployment model. The base site can have use of the basic site and navigation configuration and a set of basic features. Additional features can be implemented as Web Parts or designed and deployed as separate feature packages.

If entities are running multiple site collections (for example, for different brands), the features can be deployed at the Web application level and enabled selectively for each brand as required. This model provides flexibility for developing, deploying, and managing front-end-specific content and dynamic display and processing logic separately, while allowing them to work together to present a unified view to the user.

System Architecture – Deployment View

The deployment view of the recommended Commerce Server 2009 architecture is illustrated in Figure 17.

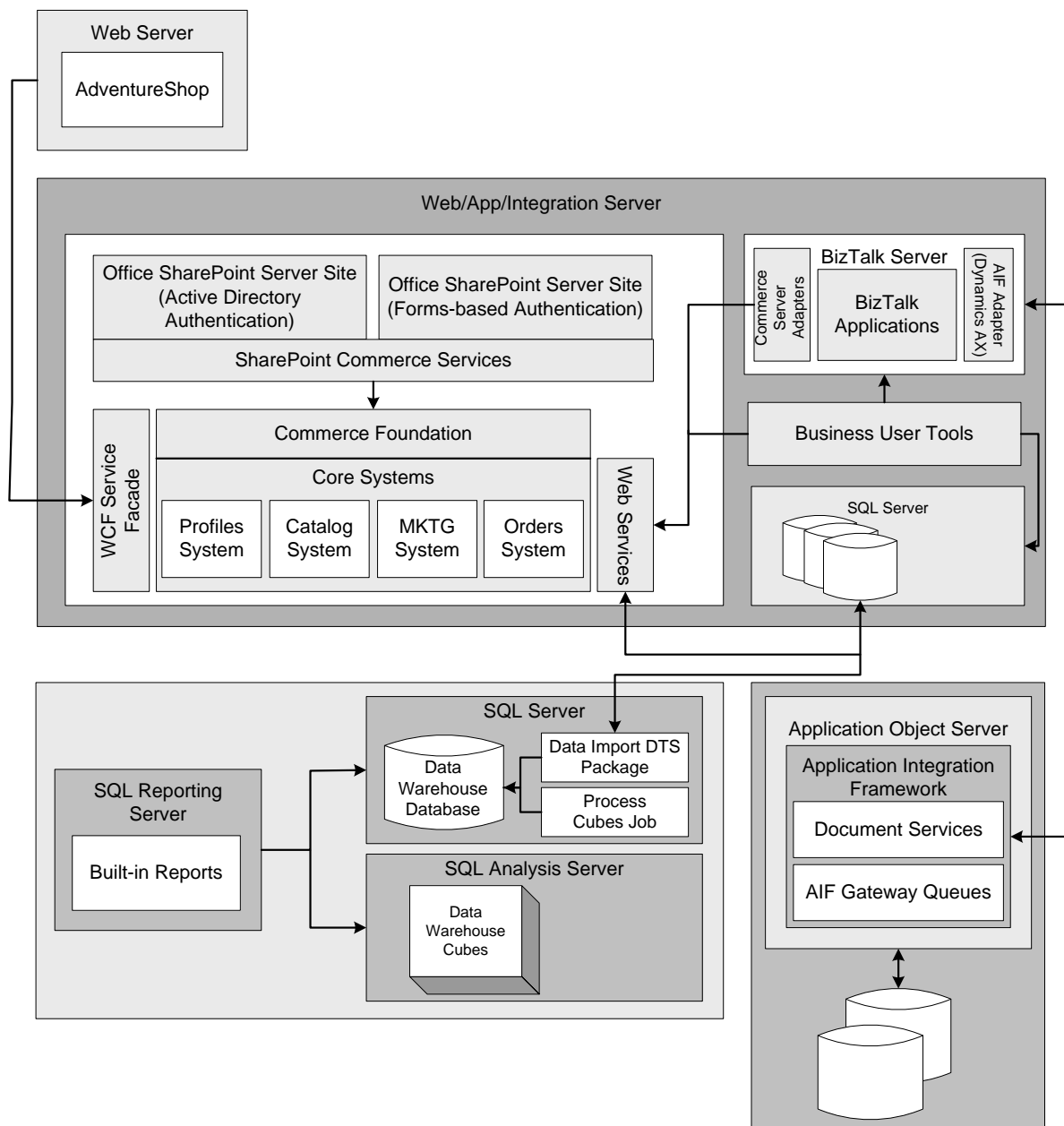


Figure 17. A deployment for Commerce Server 2009 system

Figure 17 shows a typical Commerce Server–based system deployment. Two different approaches can be taken for deployment. In one approach, the Web and application servers are combined, with both SharePoint and Commerce Server 2009 running on the same physical server computers that act as the front end. In the other approach, SharePoint is deployed on the front-end Web servers and Commerce Server 2009 is installed on application servers, with the Commerce Foundation Services published as WCF interfaces for remote consumption.

Figure 17 also illustrates the use of a pure ASP .NET–based Web site (Adventureshop) as the front end, which in turn can use the Commerce Foundation Services remotely from the application tier. This model is preferred when the system must serve multiple channels that might be implemented as separate Web sites or client applications.

The SQL Server layer provides the OLTP, analysis, and reporting services. The OLTP servers are deployed as a separate cluster. These could be Windows Server clusters or mirrored instances that are based on system availability requirements. Analysis services point to the data warehouse database, which is deployed on a separate SQL Server instance. Reporting services instances can also point to the data warehouse and analysis services, depending on the reporting needs.

BizTalk Server is deployed on its own server computer or cluster, depending on the scalability and availability requirements. BizTalk is configured to interface with both the Commerce Server servers (that host the core Web services) and the backend Microsoft Dynamics AX (or other ERP) servers.

The deployment illustrated in Figure 17 shows the SharePoint Commerce Services calling into the Commerce Foundation Services. This interaction can use different communication mechanisms. By default, the calls are in-process. The foundation services can also be deployed on a different physical tier, and the SharePoint Commerce Services (and any custom presentation tier components) can use the foundation services remotely to better facilitate multi- tier architectures.

Figures 18 and 19 are sequence diagrams that illustrate sample call sequences for each of these scenarios. The calls to the Commerce Foundation Services are routed through the service locator class by the operation service agent. The service locator looks up the commerce client configuration to determine the mode of calling. Depending on whether in-process or remote mode is configured, the service locator instantiates a service container or service proxy, which in turn calls the Commerce Foundation Service component or service.

This design model makes it convenient to implement the service and the presentation services layer without dealing with the specifics of how the foundation service is published and consumed. The mode can be switched appropriately based on the deployment design and scalability and server configurations.

Figure 18 illustrates the call sequence for the in-process mode and Figure 19 shows the call sequences for the remote mode calls using the service proxy.

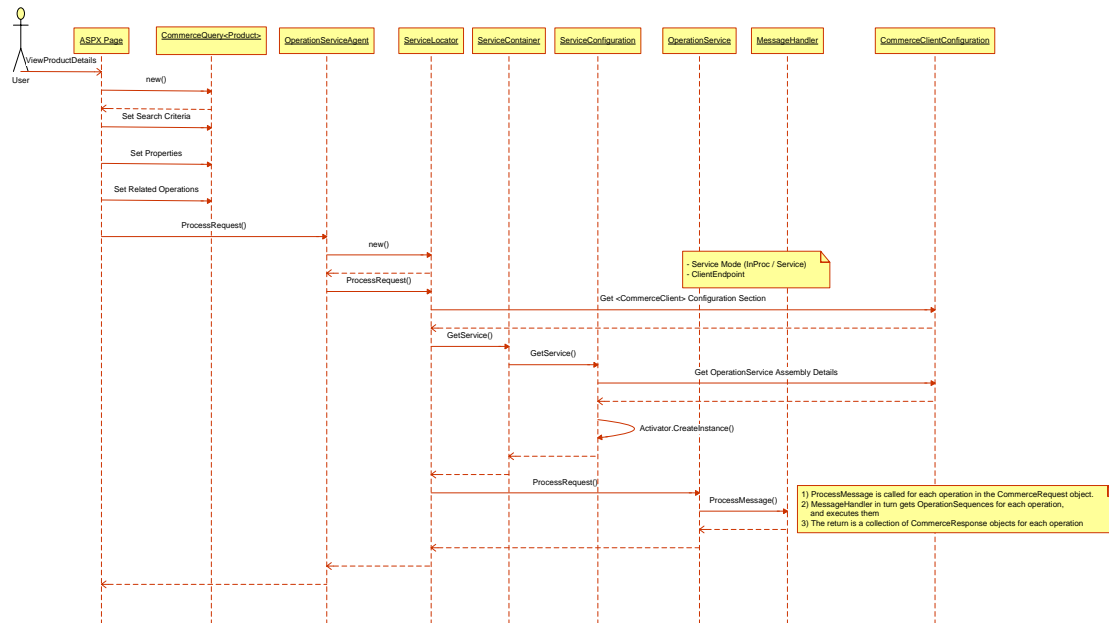


Figure 18. In-process invocation of foundation services

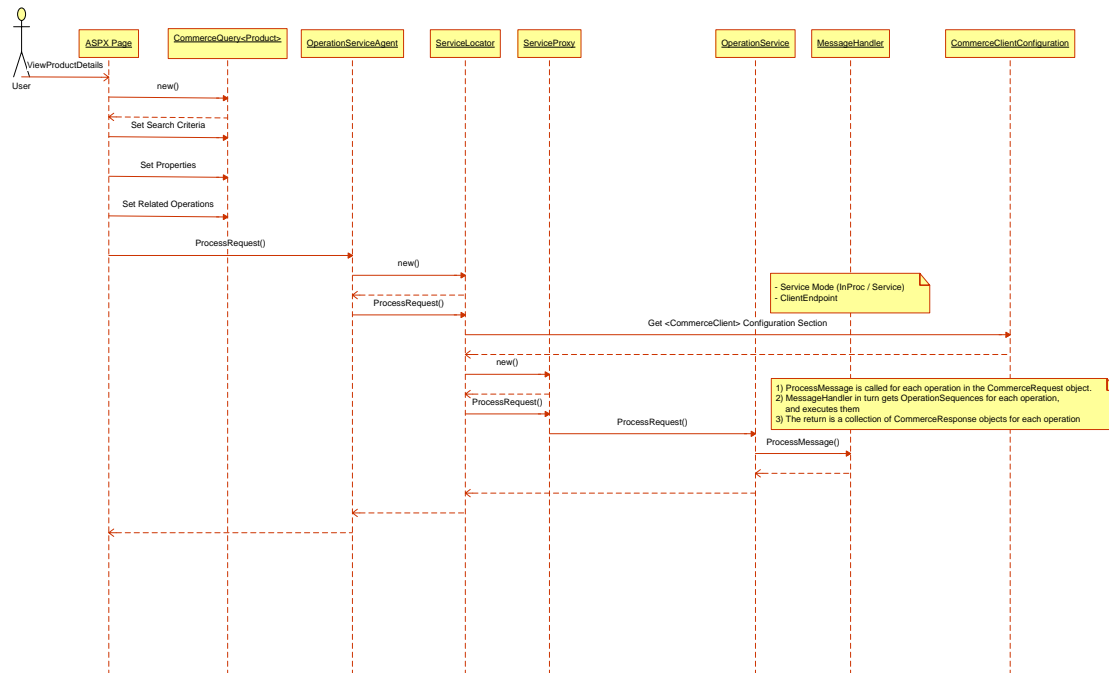


Figure 19. Remote invocation of foundation services

Common Trades-offs

The following deployment options have tradeoffs that you should consider as you design your system.

- Deploying site collections vs. deploying sites
- Catalog data synchronization with the back-end; frequent updates ensuring that the content is current vs. performance
- Multi-lingual support: using variations vs. resource files
- Extending the management UIs
- Inventory management and synchronization.
- User experience vs. performance (with some samples)
- Other decision points based on the scenario

Designing Commerce Server 2009 Systems

As you approach the design of a Commerce Server 2009 solution, start by considering whether to use Commerce Server as a solution or usage of Commerce Server as a platform that you will customize. In either case, the approach should be to identify and follow the proven design patterns that apply to your scenario.

Choosing an approach will help you to focus on patterns related to that approach, and avoid creating a complex implementation with a mix of conflicting patterns (which is possible because of the composite and integrated nature of a Commerce Server solution).

Table 8 describes the typical approaches and related patterns that are recommended for a Commerce Server 2009 implementation. These approaches are explained more fully in the sections that follow the table.

Table 8: Typical Approaches to a Commerce Server 2009 Implementation

Deployment option	Features available	Remarks	When to use
Commerce Server 2009+ MOSS 2007	Complete Commerce Server functionality Content management Site hierarchy organization and related management Web Part and feature framework SharePoint Commerce Services Integrated search	Provides for deploying and managing an e-commerce system by combining enterprise-level commerce, content management, and search functionality. SharePoint based deployment provides great flexibility to organize the e-commerce sites at multiple deployment levels – Web apps, site collections, sites or sub-sites – depending on the specific needs. This will allow for	Use for medium to large complexity e-commerce Web sites, especially if content management and styling are important. Ideal for scenarios in which the e-commerce site coexists with another corporate or marketing site. Also useful if there is a need to manage the entire system as a platform of multiple Web sites (multi-brand, multiple stores, or

		<p>access control at each level as best suited.</p> <p>SharePoint search provides great capabilities to enable flexible content search across the site. This can be combined with Commerce Server search to provide an integrated search experience for users.</p>	multiple channels)
Commerce Server + WSS 3.0	<p>Complete Commerce Server functionality</p> <p>Site hierarchy organization and related management</p> <p>Web Part and feature framework</p> <p>SharePoint Commerce Services</p>	<p>Provides for deploying and managing an e-commerce system by combining enterprise-level commerce and site management features. Content management and search are not provided out of the box.</p>	<p>Use for small to medium complexity e-commerce systems that might not require extensive content management capabilities. Search might still be required. It could be implemented as a custom feature or it could be based on Microsoft Search Server or another third-party solution.</p> <p>If you have a small site that is not content-heavy, the search functionality can be handled by the Commerce Server search features.</p>
Commerce Server 2009 + ASP.NET	<p>Complete Commerce Server functionality</p> <p>ASP.NET platform for custom implementation of presentation layer</p>	<p>Provides a strong enterprise-level commerce engine.</p> <p>The front end must be custom built or built on an alternate content management framework. ASP.NET Web Parts can be used to componentize the presentation layer and support reuse of components across stores and channels.</p> <p>An alternate search mechanism such as Microsoft Search Server can be used if required.</p>	<p>Can be a choice if the enterprise architecture requires an alternate content management platform, or if you need to develop a very different looking user interface and features than those provided by SharePoint Commerce Services, or if you have licensing constraints.</p>
Any of the above with	Apart from what is	Commerce Server	Can be a good choice if

WCF interface enabled for Commerce Foundation Services	mentioned above: Service based interface to foundation services Flexibility for distributed deployment – i.e., SharePoint/ WSS/ ASP.NET front end Web servers connecting via WCF to Commerce Server App tier	Foundation services can be enabled for WCF interfaces by just configuration if the implementation follows the recommended coding practices to call into Commerce server	the architectural frame requires distributed deployment due to structural or security reasons, or if there is a need for the Web tier and application tier to have different levels of scalability
---	--	---	--

System Design

Commerce Server 2009–based architecture provides multiple options for solution design and deployment based on the priority of solution features as well as infrastructure and architecture constraints.

The key components of an architecture based on Commerce Server 2009 are the E-commerce engine, content management, search, product management, campaign management, user management, order management, and integration with back-end systems. Depending on the emphasis and priority of features that fall into each of these buckets, different combinations of solution components can be assembled. A set of design templates that reflect both the architectural frame and typical scenarios is described in the next section.

Design Template – Commerce Server 2009 Full Framework Solution

This design template leverages the full functionality of the Microsoft e-commerce technologies and provides a comprehensive solution. Commerce Server 2009 provides the entire commerce engine and management features. Microsoft Office SharePoint Services 2007 is used for content management and search, and BizTalk is used for integration with back-end systems. This is a good approach for a situation that requires:

- A good e-commerce and management platform
- A robust content management platform
- Integrated search
- Integration with legacy and line-of-business systems
- Specific monitoring and tracking for integration

Figure 20 illustrates a design based on this template.

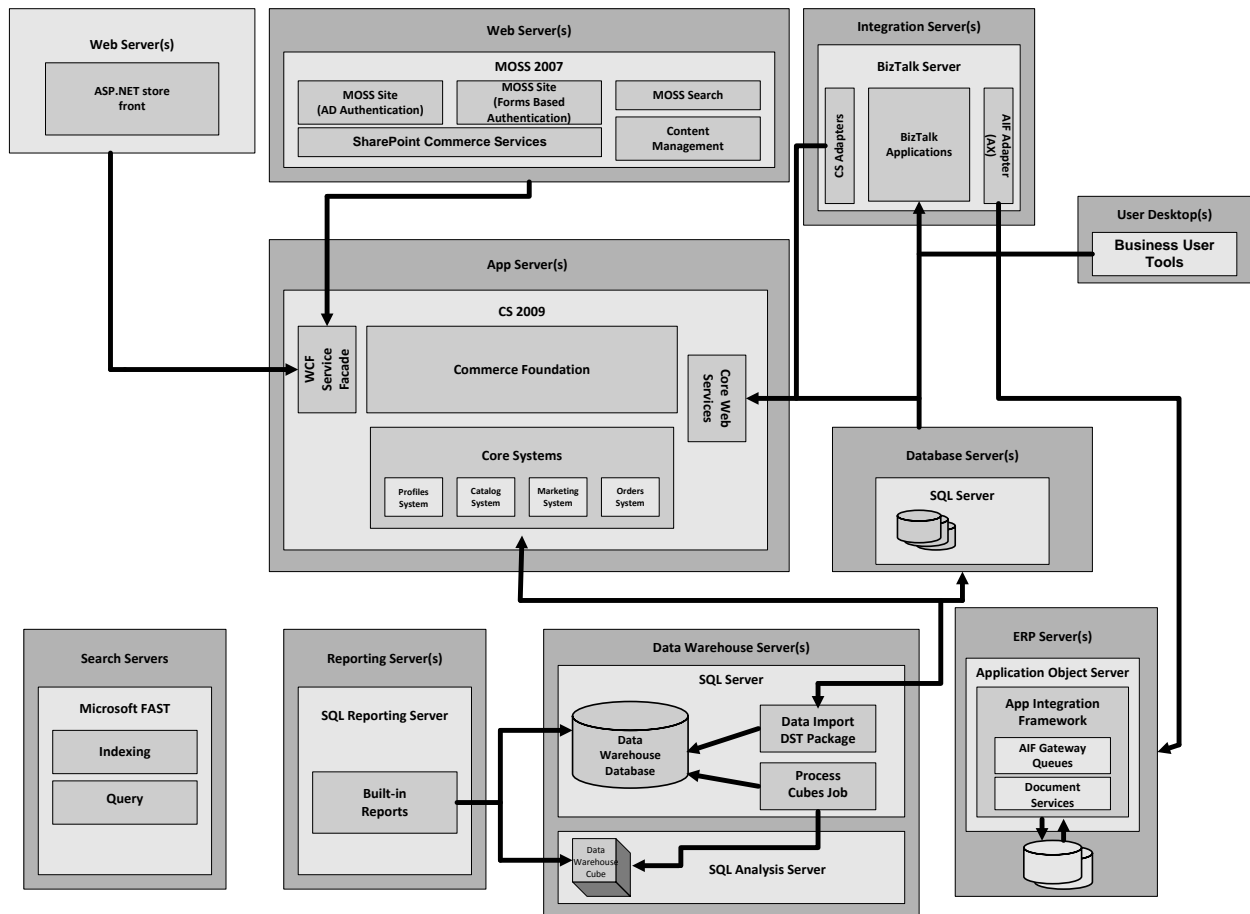


Figure 20. Design template for a full framework solution

Content Management

In this design template, Microsoft Office SharePoint Services is recommended as the content management platform that can fully use SharePoint features for content creation, review, approvals, staging, and publish cycle.

Web Site Deployment

This design template recommends a standard SharePoint deployment model for the Web sites based on a Web application/site collection/site hierarchy, and standard SharePoint configurations for different access points and authentication mechanisms.

The front-end application is deployed as a SharePoint application. The e-commerce site is configured as one or more site collections with sub-sites if required.

Web Commerce Functionality

This design template recommends hosting the SharePoint Commerce Services along with Office SharePoint Services to leverage the Web UI functionality for catalogs, marketing, orders, and user profiles. The services provide Web Parts that can be customized with minimal effort. These are best suited for situations where the required functionality can be matched to the features provided by the Commerce Services Web Parts. This is possible in most cases because the Web Parts provide rich features and are configurable.

Search

Office SharePoint Services provides support for integrated search. However for more advanced and scalable search needs, Microsoft FAST ESP solution can be used. Commerce Server 2009 provides the core e-commerce Web and management functionality.

E-commerce Engine

The e-commerce application layer is deployed as a separate physical tier in this design template. This is recommended if the overall objective is to implement a system that will be the base for multiple brand, marketing, and e-commerce sites. This deployment design is also well-suited for integration of multiple channels through the same service instance.

Product and Campaign Management

The business applications provide a good way to implement catalog, product, user, orders, and campaign management. In addition, this design template can use the SharePoint Commerce Services administrative Web Parts to provide a Web UI for administrators to use when they update basic catalog data.

Integration with Line-of-Business and Legacy Systems

We recommend that you use BizTalk Server to assemble the integration layer. By using BizTalk adapters, you will need minimal custom development to connect to the back-end systems and Commerce Server.

Use this design template if:

- Most presentation layer features required a match with SharePoint Commerce Services.
- Content management needs are important.
- You need to use built-in search features.
- Time to production is critical.
- You require a platform or framework for hosting and managing a large number of sites.
- You need to provide support for multi-channel commerce.
- No architectural constraints restrict your use of Office SharePoint Services or BizTalk.

Design Template – Commerce Server 2009 with SharePoint Commerce Services on Windows SharePoint Services

There are scenarios in which the primary objective is to provide the e-commerce functionality without much emphasis on content management. In this situation, a Windows SharePoint Services–based design allows you to use Commerce Server by still leveraging the SharePoint Commerce Services functionality. Figure 21 illustrates this type of design.

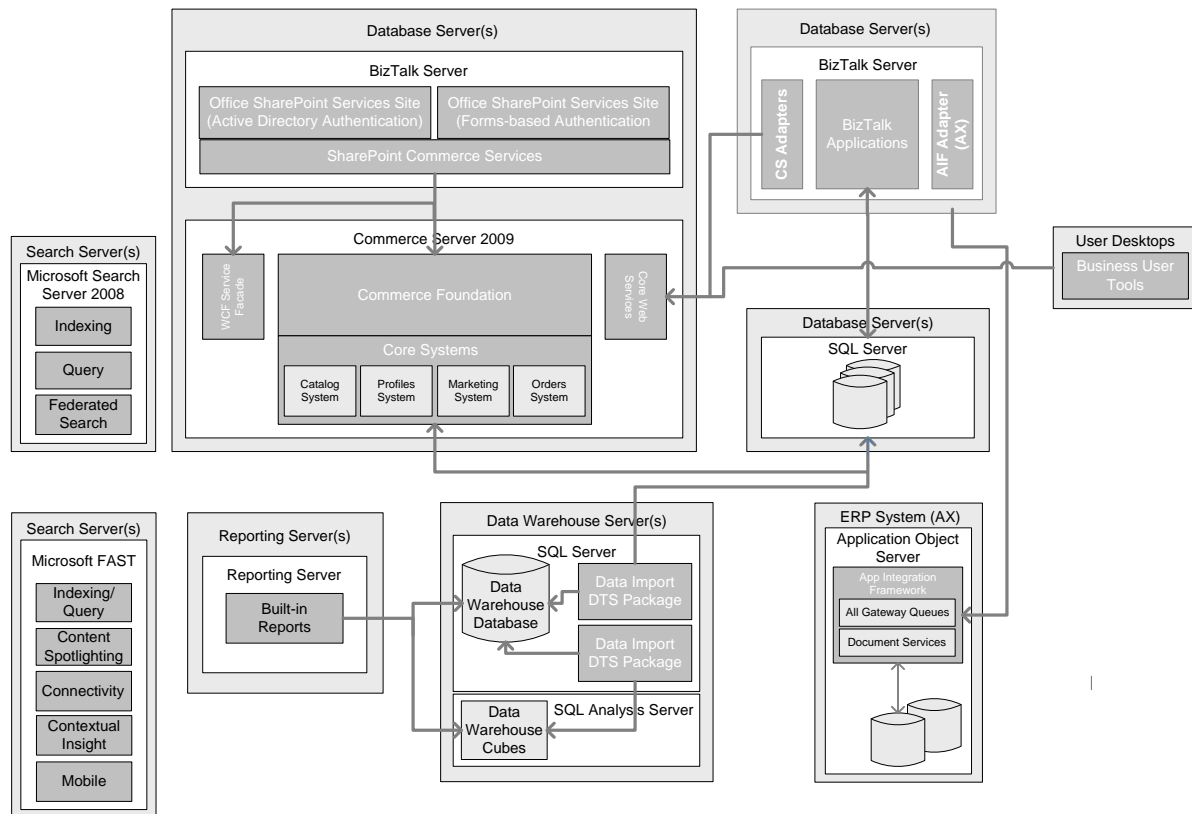


Figure 21. Design template – Commerce Server with SharePoint Commerce Services on Windows SharePoint Services

Content Management

This design template provides limited content management features. The Windows SharePoint Services–based deployment supports basic page site templates, page layouts, and the Web Part collections. However, management by business users is limited and must be done primarily by using the IT team or a custom solution built on Windows SharePoint Services 3.0. The Web Part framework is fully available, and therefore, SharePoint Commerce Services can be fully leveraged as long as the feature set matches.

Web Site Deployment

This design template recommends the standard SharePoint deployment model for Web sites based on a Web application/site collection/site hierarchy, and standard SharePoint configurations for providing different access points and authentication mechanisms.

The front-end application is deployed as a SharePoint application. The e-commerce site is configured as one or more site collections with sub-sites, if required.

Web Commerce Functionality

This design template also recommends hosting the SharePoint Commerce Services within Windows SharePoint Services 3.0 to leverage the Web UI functionality for catalogs, marketing, orders, and user profile. This works best if the required functionality can be matched to the features provided by

the Commerce Services Web Parts, which is possible in most cases because the Web Parts provide rich features and are also configurable.

Search

WSS provides basic search features, but has limited content sources, relevance, indexing, connectivity, and administration features. We recommend that you use a solution based on either Microsoft Search Server or Microsoft FAST ESP to enable advanced and scalable search features, which can then be combined with Commerce Server 2009 search functionality.

E-commerce Engine

Commerce Server 2009 provides commerce Web and management functionality. The multi-channel foundation services can be deployed either in the same tier as the presentation layer or as a separate tier, depending on whether you need to offer foundation services functionality as a centralized service across sites and channels.

Product and Campaign Management

The business applications provide a good way to implement catalog, product, user, order, and campaign management. In addition, this design template can use the SharePoint Commerce Services admin Web Parts to provide a Web user interface that administrators can use to update basic catalog data.

Integration with Line-of-Business and Legacy Systems

We recommend that you use BizTalk Server to assemble the integration layer. By using BizTalk adapters, you will need minimal custom development to connect to the back-end systems and Commerce Server. Depending on your integration requirements and integration interfaces, you might need to implement alternate approaches that involve direct service calls or you might need to design a custom integration layer.

A custom integration layer based on WCF and WF technologies can be used if the integration involves mostly direct synchronous calls. Integration that requires a good amount of transformation, orchestration, and connection with legacy systems is better accomplished with a BizTalk-based approach. BizTalk supports transformation and orchestration, and is supported by a wide range of adapters to legacy and line-of-business systems.

This design template provides most of the features of the full solution except for content management. You can implement Windows SharePoint Services-based search; however, a better solution would be to use one of the recommended search solutions such as Microsoft Search Server or Microsoft FAST ESP.

Use this design template if:

- Most presentation layer features required have corresponding SharePoint Commerce Services features.
- Content management and publishing needs are not of the highest priority.
- Time to production is important.

- The solution is intended to have a limited number of sites.
- Support for multi-channel commerce is required.
- There are architectural or other limitations on using Office SharePoint Services.

Design Template – Commerce Server 2009 with an ASP.NET Front End

This design template caters to a scenario in which a pure ASP.NET front end must leverage the e-commerce engine functionality of Commerce Server. This design template does not use SharePoint Commerce Services and related Web Parts. A custom ASP.NET front end is to be developed, and it will call the Commerce Server Foundation Services directly for all e-commerce functionality across the catalog, users, orders, and marketing. Figure 22 illustrates this template.

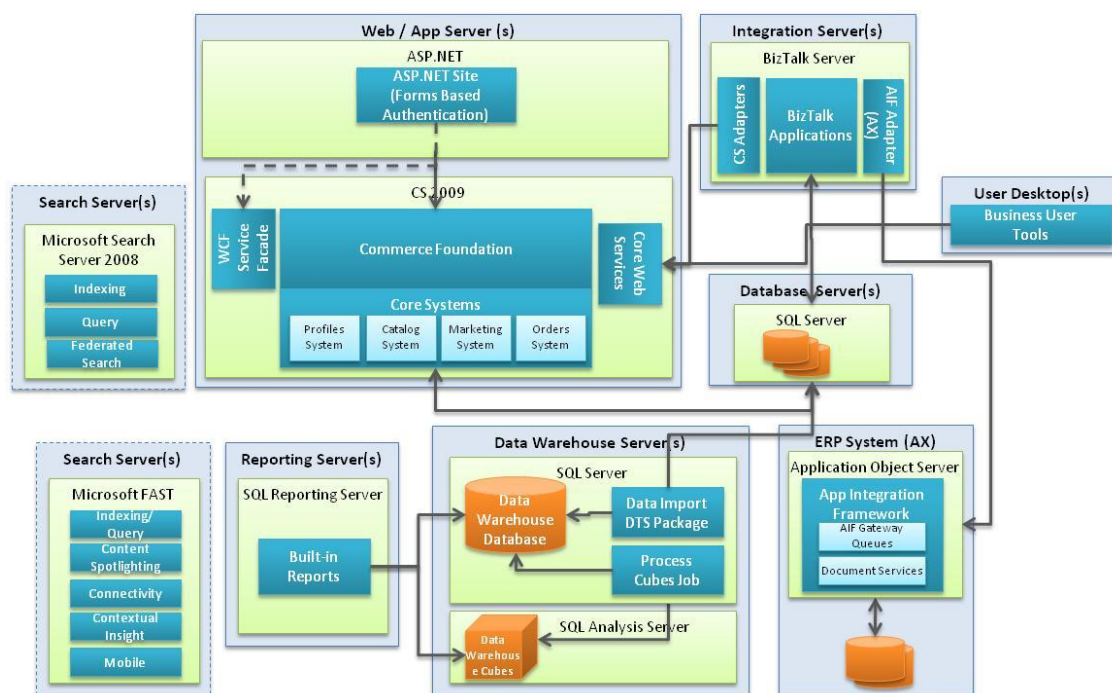


Figure 22. Design template – Commerce Server 2009 with an ASP.NET front end

Content Management

This design template provides almost no content management features out of the box. However, you can build and add content management features of your own. If you use a third party content management solution, it can use the ASP.NET master pages and Web Parts to provide a consistent UI and consistent behavior.

Web Site Deployment

For this design template, we recommend the standard ASP.NET deployment model for Web sites based on IIS Web applications. The abstraction level provided by the SharePoint infrastructure to maintain Web application site collection hierarchies and navigation is not available in this template. Therefore, this design template has limited flexibility with respect to extending and providing multiple access points with multiple authentication mechanisms out of the box, especially when compared to a SharePoint-based solution. It is still possible to do such a deployment, but it requires additional deployment design and maintenance time.

Web Commerce Functionality

This design template is based on ASP.NET and can host some of the SharePoint Commerce Services Web Parts. However, this approach provides very limited Commerce Services flexibility and extensibility.

This design template is best suited for solutions that require a custom set of UI behaviors for the Web Parts.

Search

For this design template, we recommend that you use a solution based on either Microsoft Search Server or Microsoft FAST ESP to enable advanced and scalable search features. You can combine these search features with Commerce Server 2009 search functionality.

E-commerce Engine

Commerce Server 2009 provides the commerce Web and management functionality. Multi-channel foundation services can be deployed in the same tier as a presentation layer or as a separate tier, depending on whether you need to provide foundation services functionality as centralized services across sites and channels.

Product and Campaign Management

The business applications provide a good way to implement catalog, product, user, order, and campaign management features. The SharePoint Commerce Services admin Web Parts cannot be used out of the box because this will require running them under SharePoint. If your solution requires a Web front end, you will need a custom implementation. The custom front end could make limited use of the SharePoint Commerce services admin Web Parts.

Integration with Line-of-Business and Legacy Systems

We recommend that you use the BizTalk Server to assemble the integration layer. By using BizTalk adapters, you will need minimal custom development to connect to back-end systems and Commerce Server. Depending on your integration requirements and integration interfaces, you can use alternate approaches that involve direct service calls or you can design a custom integration layer.

You can use a custom integration layer based on WCF and WF technologies if the integration involves mostly direct synchronous calls. Integration that requires a good amount of transformation, orchestration, and connection with legacy systems is better accomplished with a BizTalk-based approach. BizTalk provides support for transformation and orchestration and is supported by a wide range of adapters for legacy and line-of-business systems.

This design template provides all of the e-commerce engine and management features, and leaves the implementation of the front end completely to the solution developer.

Use this design template if:

- Most presentation layer features required do not align with SharePoint Commerce Services.
- Content management and publishing needs are not of highest priority and you need a highly custom UI.

- Time to production is not very critical.
- Solution is oriented toward deploying and maintaining a limited number of sites.
- Support for multi-channel commerce is required.
- There are architectural or other limitations on using Office SharePoint Services or Windows SharePoint Services.

Design Template – Commerce Server 2009 Solution that Extends SharePoint Commerce Services

This design template addresses scenarios in which there is a need to have a good balance of out of the box features and customizations. In this approach, the focus is on using product features as much possible, and deploying and enhancing the solution by building extensions and customizations to the framework. Figure 23 illustrates this design template.

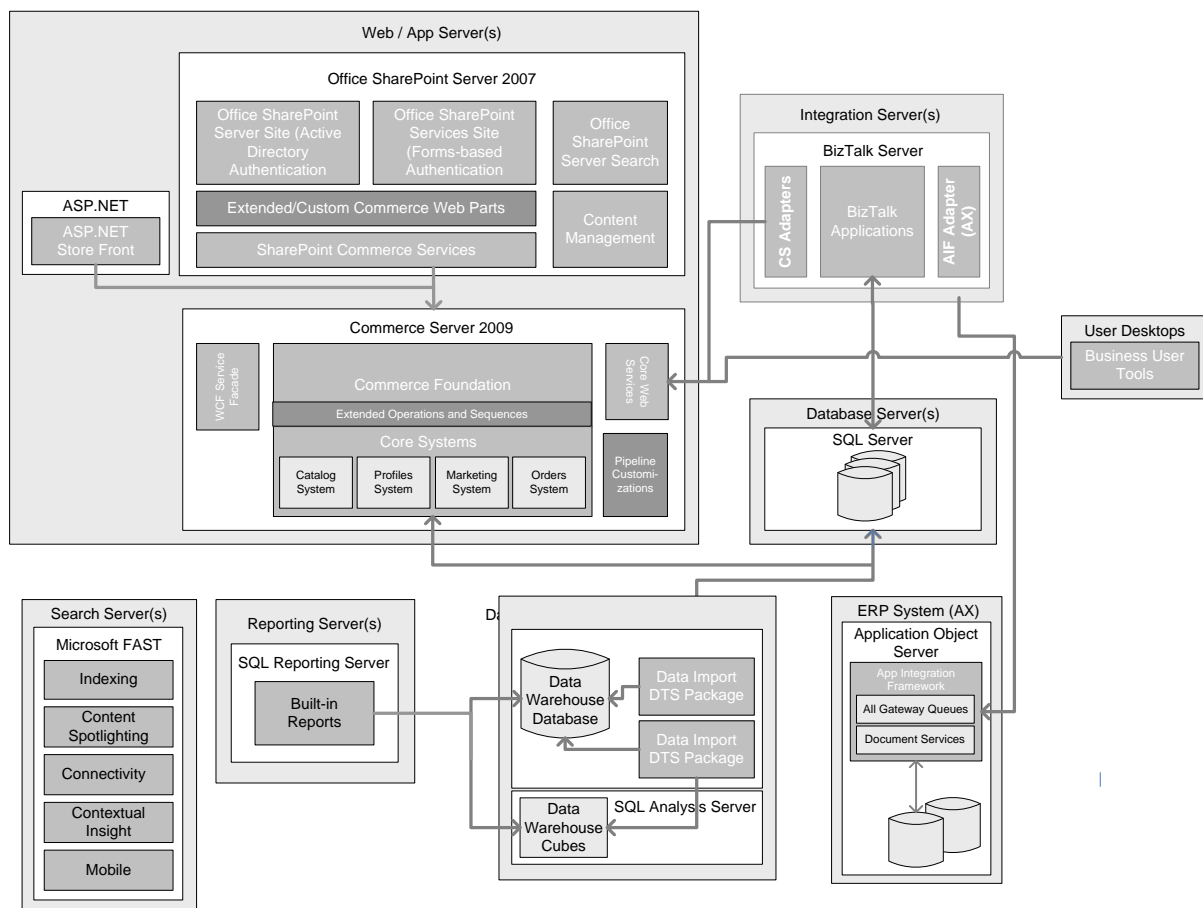


Figure 23. Design template – Commerce Server 2009 solution that extends SharePoint Commerce Services

Content Management

In this design template, Office SharePoint Server is recommended as the content management platform because it provides complete SharePoint features for managing the content life cycle (creating, reviewing, approving, staging, and publishing).

Web Site Deployment

For this design template, we recommend a standard SharePoint deployment model for Web sites based on the Web application/site collection/site hierarchy, and standard SharePoint configurations for providing different access points and authentication mechanisms.

The front-end application is deployed as a SharePoint application. The e-commerce site is configured as one or more site collections with sub-sites, if required.

Web Commerce Functionality

For this design template, we recommend hosting the SharePoint Commerce Services with Office SharePoint Services to leverage the Web UI functionality for catalogs, marketing, orders, and user profile. These services provide Web Parts that can be customized for style, navigation, usability, and display behavior with minimal coding effort. This is suitable if the behavior and functionality needed for the Web site is in line with the functionality provided by the SharePoint Commerce Services Web Parts.

If required, you can extend the Web Parts or create custom Web parts to enhance the functionality. In this situation, we recommend that you model the new set of Web Parts on those provided in SharePoint Commerce Services. You can deploy and enable the custom Web Parts for specific sites, as needed.

You can use this approach when the solution feature requirements are similar to those that are available in Commerce Services, but require specific customizations and extensions.

Search

Office SharePoint Services provides support for integrated search. However, for more advanced and scalable search functionality, you can use Microsoft FAST ESP. Commerce Server 2009 provides the core e-commerce Web and management functionality.

E-commerce Engine

The commerce application layer is deployed as a separate physical tier in this design template. We recommend this approach if your overall objective is to implement a platform or framework that will be the base for multiple brand, marketing, and e-commerce sites. This deployment design is also well-suited for integration of multiple channels through the same service instance.

Product and Campaign Management

The business applications provide a good way to implement catalog, product, user, order, and campaign management. In addition, this design template can use the SharePoint Commerce Services admin Web parts to provide a Web UI that administrators can use to update basic catalog data.

You can enhance this capability further by extending or adding admin Web Parts based on your specific requirements.

Integration with Line-of-Business and Legacy Systems

We recommend that you use BizTalk Server to assemble the integration layer. By using BizTalk adapters, you will need minimal custom development to connect to back-end systems and Commerce Server.

Use this design template if:

- Most required presentation layer features align with SharePoint Commerce Services, but require enhancements.
- Content management is important.
- You need to leverage existing search features.
- Time to production is critical.
- You require a platform or framework with enhanced custom functionality for hosting and managing a large number of sites.
- You need support for multi-channel commerce.
- You do not have any architectural constraints that restrict your use of Office SharePoint Services or BizTalk.

Development Best Practices for Digital Marketing and E-Commerce Systems Using Commerce Server 2009

One of the most important aspects of implementing the prescribed architecture is to make sure that the features built in to the platform are used effectively and that the additional features and customizations are made in a manner that supports modular development and deployment.

Solution Structure

A comprehensive Commerce Server 2009 solution consists of technology components that belong to the categories described in the following subsections. The solution must be organized into different layers for presentation, business logic, and data access. The Commerce Server architecture already provides a layered architecture base, and this should be leveraged.

ASP.NET Web Parts and Web Pages

This represents the bulk of the presentation layer and should be organized as a separate Web project.

The presentation layer should use the Commerce Server Foundation Services—either in-process or through WCF calls—to invoke the e-commerce logic. Figure 24 illustrates this structure.

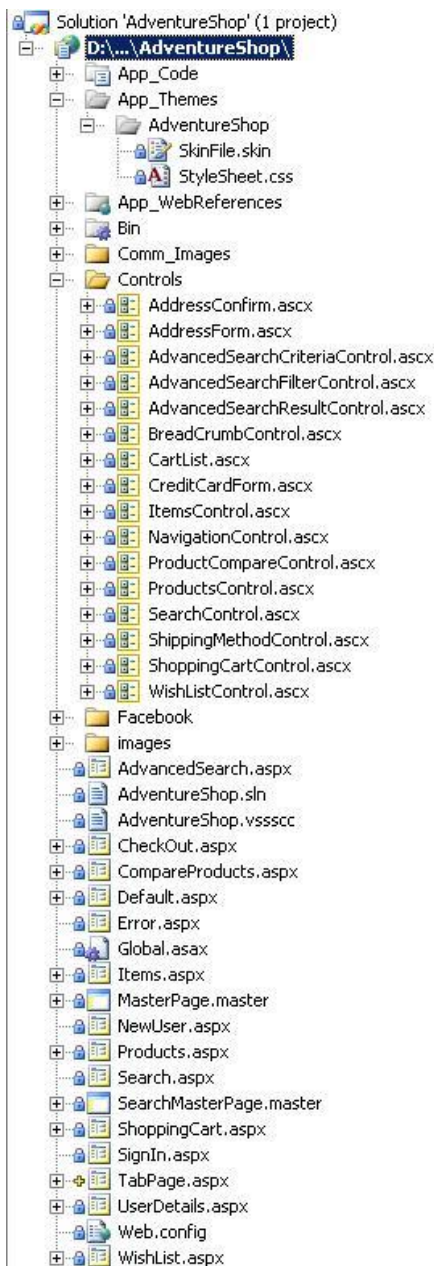


Figure 24. Web Parts and Web pages organization

.NET Components (Business Logic, Data Access)

Most of the e-commerce functionality is encapsulated in the foundation services. You can add business logic to the channel-specific operation sequences by building and configuring custom operation sequence components.

- The Commerce Server Core pipelines also could be modified to add or modify e-commerce–related business logic. It might be preferable to use custom operation sequence components rather than custom pipeline components. This is purely a performance consideration because the latter requires COM interoperability.
- If there is other logic that is not specifically related to e-commerce, it should be developed as a separate parallel layer that is invoked from the presentation layer. The additional business logic can be invoked in-process or through a WCF service, depending on the

architecture of the solution. This additional business logic should be separated into a one or more specific Visual Studio projects.

- All Commerce Server data access should be via the foundation services and will thus leverage the Commerce Server Core API for access. If specific, required core functionality is not available in Commerce Foundation Services, you can use the core services after careful evaluation.
- Use the multi-layered approach for any custom data repository access. Create a separate data access layer called from the custom business logic layer or from custom components from the foundation services layer. The data access layer should be separated into one or more specific Visual Studio projects.

Office SharePoint Services Web Parts

Separate these Web Parts into a separate Visual Studio project. The Office SharePoint Services Web Parts can be basic SharePoint Web Part implementations or they could use ASP.NET user controls or Web Parts from other Visual Studio projects within themselves.

Develop ASP.NET user controls or Web Parts that can be encapsulated into SharePoint Web Parts. This allows you to use these Web Parts (with minimal changes) in both ASP.NET and Office SharePoint Services Web sites. Use separate Visual Studio projects for the SharePoint Web Parts and the ASP.NET user controls and Web Parts. This ensures that SharePoint-specific DLLs are not needed when the Web Parts are used in an ASP.Net application.

Office SharePoint Services UI Elements

Develop Office SharePoint Services layouts, master pages and other UI elements as a separate deployment package.

BizTalk-Specific .NET Components

You can organize these components as a separate solution that has the required files and references to the required custom components and Commerce Server interfaces.

Create a separate BizTalk application for each integration interface: catalog, inventory, order, and profile.

Configure all XML schemas to be part of one common BizTalk application, and deploy that common application on the default BizTalk application. Other BizTalk applications can then refer to the schemas.

Figure 25 illustrates this structure.

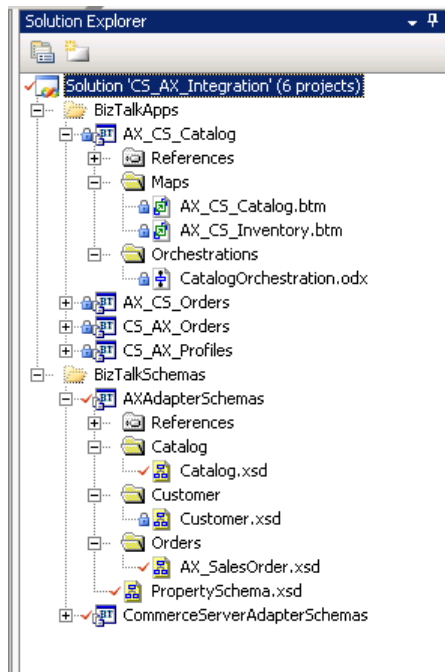


Figure 25. BizTalk .NET components

WCF Services

The Commerce Server Foundation Services can be exposed as WCF services with the required configuration.

Additional services should be organized as a separate Web project that refers to multi-channel foundation services, the Commerce Server Core APIs, or other custom components, as required.

Build Setup

A typical Commerce Server 2009 build process involves the steps described in this section. It is a good practice to prepare an automated build that replicates these steps and then deploy it in a continuous or scheduled build environment. This will facilitate early and continual verification in an iterative manner. The Visual Studio Team Foundation Server (TFS) build features can be used to enable this.

The Commerce Server build process occurs as follows:

1. Retrieve the latest source code.
2. Build the .NET components.
3. Build the Commerce Server 2009–related components.
4. Build the ASP.NET components.
5. Build the Office SharePoint Services s–related components and Web Parts
6. Prepare the deployment package(s): SharePoint, Commerce Server, and BizTalk (as required).
7. Deploy in the development environment.
8. Report the build results.

Performance Benchmarking and Verification

Performance and scalability are key requirements for an e-commerce system. It is a good practice to focus on these aspects starting at the beginning of your project. Use the following steps as a reference guide for building performance-related aspects into your development process.

1. Refer to ASP.NET and Commerce Server performance guidance as you plan and develop the system.
2. For performance benchmarking, select four or five use cases or user stories that cover the end-to-end flow within the system.
3. Begin compiling performance measurements from the use cases during the initial stages of development, as the use cases are being built.
4. Monitor the performance variations, and then make required changes to the implementation and guidelines.
5. Plan for formal performance testing towards the final iterations.
6. Leverage the .NET, IIS, ASP.NET, and Commerce Server configurations to the fullest before deploying any customizations for performance improvement.

Site Structure

As you develop the site structure, develop the Web Parts so that they can run within the site context in any site structure. Do not assume any specific site structure when developing functionality and Web Parts. Instead, refer to your functional specifications and use cases, and research best practice guidelines for the configuration that you will be implementing.

Create the e-commerce front-end site so that it can run in its own site collection. Typically, this is deployed as the root site of the site collection. This helps provide flexibility to attach the site to a common or separate content database, depending on content size, deployment, and other related requirements

Ideally, the digital marketing site and the e-commerce site should be separate sites that can be deployed and updated independently. This is useful because functional update requirements and the content update process could be different. Plan these as separate site collections without any dependency on mutual relative paths. This will allow you to deploy them as separate Web applications if necessary.

If you need to set up sub-stores for multiple brands, you can deploy them as sub-sites of the main e-commerce site or as individual site collections. Typically, when you use SharePoint Commerce Services for Commerce Server 2009, there is only one Commerce Server application in the back end for each front-end site. Therefore, if you need to set up separate commerce applications—for example, for different brands or different countries—it is better to model the front end as separate applications also.

Use the SharePoint content features to store and maintain custom content or configuration data used by the presentation layer. Evaluate the performance impact and benchmarks before deciding on the SharePoint feature to be used. If you need a different option, also evaluate the use of a custom database for these purposes.

Invoking Foundation Services

The presentation layer should invoke Commerce Server Multi-channel Foundation Services for read and write operations with the e-commerce engine. The invocation may result in in-process calls or WCF calls, depending on the deployment configuration used. This is a very flexible model that can be overloaded to service newer and custom processing requests. However, each call incurs a processing cost from the calling layer, through the foundation services, and down to the Commerce Server Core.

A good development approach is to minimize the number of calls to the e-commerce engine for a particular use case or Web page implementation. Note that we do not recommend an approach that sacrifices modularity, and packs use case-specific call sequences into the processing of a particular service. A more balanced approach is to use query and transactional services that can be leveraged individually or in combination. Because you cannot completely avoid situations that require multiple calls, consider using one or more of the following techniques for the implementation.

- Enable the Commerce Server cache. This does not minimize number of foundation service calls, but it makes some of them faster.
- Use a cache mechanism in the front end. This could be the ASP.NET cache or a more sophisticated cache solution, such as Windows Server AppFabric Caching (formerly code named “Velocity”). For more information, see <http://www.microsoft.com/windowsserver2008/en/us/app-main.aspx>.
- Use an implementation pattern that minimizes the number of calls; for example, the product-specific pages could use the product provider (non-visual) Web Part, which can fetch the page data as needed. The page data can then be used by all other consumer Web Parts on the page.
- Use an implementation pattern that optimizes the volume of records fetched to the front end. This is effective for lists and searches.

Integration Interfaces

An e-commerce system will need to interface with multiple external systems. Because of the multi-layer nature of the e-commerce architecture, you should carefully consider your options when you decide which layer(s) will be integrated with a specific external system. The following are some key considerations:

- Interfaces that involve transactions should be integrated within the foundation services or one of the core pipelines.
- Read interfaces can be integrated either at the foundation services or the presentation layer, depending on the nature of data.
- Integration that involves control flows across systems and that affects the user experience is better implemented at the presentation layer, with some customizations in foundation services to fetch or store related data (such as payment gateway integration with redirections to and from the payment gateway).

- Data centric integration that does not affect the user experience is best done offline or asynchronously. This can be done at the database layer or at the commerce core Web services layer, in most cases. However, in some cases, data integration that might involve more transaction processing might need to invoke the foundation services.
- Integration that involves orchestration across systems or long-running transactions should be classified separately and handled in a separate orchestration layer. Depending on the complexity, you could use BizTalk Server orchestration services or a custom implementation based on Windows Workflow Foundation.

Security

Create a development and test environment that uses properly secured deployment and user accounts for development, management, and test access. This will ensure that limited and privileged access to the different parts of the site are followed from the beginning of the project and will not be a deployment concern.

Take measures to secure the following:

- Web site access
- Foundation services access
- Operation sequence privileges
- Database access
- Commerce Server Manager application access and roles
- Office SharePoint Services deployment, with proper roles and access settings

Threat Modeling

Do a proper threat model for the site. Threat modeling is an effective technique for identifying and implementing security on an enterprise scale system. Threat modeling works based on understanding the views into the system, characterizing the security of the system, determining threats, and recommending mitigation options.

For more information about security threat modeling, see the articles about [threat modeling concepts, techniques, and tools](http://msdn.microsoft.com/en-us/security/aa570411.aspx) on MSDN (<http://msdn.microsoft.com/en-us/security/aa570411.aspx>).

Encryption

Identify sensitive data and decide on the encryption techniques to be used at different layers of the application. Consider encrypting the following:

- Any data within the user browser
- Any data in transit between the user browser and the site
- Data processed within the application
- Data stored in the repository

- Data in configuration files

Depending on the type of data and encryption scenarios, consider using techniques such as key-based encryption, hashing, or masking.

Authentication and Authorization

For authentication and authorization consider using built-in features from Office SharePoint Services 2007, Commerce Server 2009, and ASP.NET, as compared to complete custom development. Make sure that that you follow best practices recommendations such as the ASP.NET provider model when you deploy custom authentication or authorization.

Secure Page Access

Plan for secure page access in advance. Compare the non-functional requirements related to secure access:

- Is mixed mode required (where only a subset of pages are secured with https access)?
- Will user session data be shared across secure and non-secure pages?
- Are there domain name and URL mapping requirements for both modes, and what are they?
- Will relative paths be used for access, including access to Office SharePoint Services content?

Estimating Commerce Server Development Efforts

When you estimate and plan the development of a Commerce Server–based solution, a key aspect to consider is the alignment of your requirements with existing features provided by Microsoft technologies. Your goal should be to use out-of-the-box functionality effectively before designing custom extensions and solutions. Your estimate and planning should take these factors into account.

Development efforts fall primarily into the following categories:

- Software installation.
- Deployment configuration for the front end, including SharePoint applications, content databases, application pools, site collections, and the site structure.
- Security configuration, including the authentication store(s) and authorization store.
- Creation of UI elements – master pages, page layouts, themes, and so on.
- Configuration of Web Parts and other parameters.
- Setting up the e-commerce application, including configuring Commerce Server resources; setting up Web services; defining the catalog and inventory schema; configuring profiles; defining the catalog, products, and related information; and configuring the marketing parameters for campaigns, advertisements, discounts, and so on.
- Customizations to SharePoint Commerce Service Web Parts.
- Extensions to Web Parts and other functionality.

- Customizations and extensions to operation sequences and pipelines.
- New Web site functionality (across layers); for example, a product comparison feature.
- Integration with external systems.
- Search setup and configuration.
- Search customizations.
- Configuration of Web analytics and reporting.
- Custom reports.
- Packaging and deployment.
- Deployment and configuration staging.
- Content and data migration.

Figure 6, earlier in this paper, is an example of a requirements analysis to determine how much configuration, customization, and custom development is required for a solution. The test effort percentages will also depend on the amount of custom development involved.

After you have determined how much existing functionality can be used, you can estimate the customization and custom development efforts. There are several methods of calculating the effort involved in creating custom features or customizing existing ones. Most of the configuration and customizations can be reduced to multiples of specific known units of work, and the estimation done based on that.

As you create the project plan, make sure that you schedule multiple iterations. The initial couple of iterations can focus more on the out-of-the-box features, refining the technology alignment evaluation, and getting the end-to-end framework running. Also, as you prepare the schedule, group the work items into different tracks that can run in parallel, with an appropriate regular build mechanism.

One of the approaches that we have found successful is to establish a team structure that has experts assigned to a specific development track, rather than assigning people to end-to-end use case implementations. For example, Group 1 takes care of all installation and configuration aspects of SharePoint and Commerce Server across the development, test, and production environments, and also addresses requirements for packaging and deployment. Group 2 is assigned to the user experience, UI implementation, and enabling the out-of-the-box features—which is more of configuration and customization focus. Group 3 is responsible for customization that involves new code and extensions, with regular integration iterations into the Group 2 activities. If additional new functionality and integration is required, another group can be added, and that work will merge into the main track periodically, based on iterations.

Consider this type of team-based, iterative approach when you estimate the required work effort for your project.

Extending Commerce Server and Office SharePoint Server Web Parts

SharePoint Commerce Services provides a number of Web Parts that address display and functional requirements for catalogs, orders, marketing, profiles, management, and search. Your development approach should focus on product alignment, and you should attempt to use existing Web Parts as much possible.

However, there will be scenarios and use cases that require a different behavior, display, navigation, or user experience than what is provided. In such cases, you have two options: one is to customize the existing Web Part, and the other is to extend the existing Web Part.

Although they have separate concerns or functionality, all Web Parts follow a variation of the model-view-presenter (MVP) pattern. Additionally, many Web Parts isolate the display into HTML and XSLT code, separate from the data fetch and processing logic. To customize the display as well as the navigation and user experience, your first priority should be to look for customizations to the cascading style sheets (CSS), HTML, and extensible style sheet language (XSL) to which the Web Part refers.

If you need to alter the fetch behavior or processing, you might have to make changes to the implementation code of the view, presenter, or controller. In such cases, carefully evaluate the change design, and then make appropriate additions and updates to the code.

MSDN provides guidelines for extending a Web part. Any changes to default Web Parts will have to be signed by your own separate strong name keys, and then deployed. Also, pages that use the Web Parts might need to be modified to use the new Web Parts.

All the assemblies of the SharePoint Commerce Services are delay signed. For each area, such as catalog, profile, or order, the source code is organized into three project types: common, ASP.NET Web Part project, and SharePoint Web Part project. The common project contains the entity, controller, and other related classes. The ASP.NET Web Part project contains the presenter, view Interfaces, ASP.NET user controls (that implement the interfaces), ASP.NET Web Part, and other related source code. The SharePoint project contains the SharePoint Web Parts that extend the ASP.NET Web Parts and implement the SharePoint-specific interfaces.

Dependencies

Any source change to these projects requires a recompilation; therefore, you will need to re-sign them with your own key. Additionally, if there are changes to the common project, then the ASP.NET Web Part and SharePoint Web Parts have to be modified and re-compiled. Similarly, changes to the ASP.NET Web Part project source code will require the SharePoint Web Part project to be changed and recompiled.

The following sections describe examples of customizing and extending the Web Parts based on the reference implementation.

Customizing the Product Details Web Part

The Product Details Web Part is a consumer Web Part that fetches data from the product provider Web Part hosted on the same page. This Web Part displays product detail property values, such as product name, description, variants list, and price.

Our reference implementation required a change in the way that the information was arranged in the display. The data to be displayed was already being fetched by the product provider Web Part and was available to the product details Web Part. Because this is a display change, it could be done by modifying just the display arrangement within the XSLT.

Required Changes

Create and apply a new XSLT for the product details.

Steps Involved

7. Create the new XSLT.
8. Add the new XSLT to the Web Part.
9. Change the Web Part property to use the new XSLT.

Extending the Product Provider Web Part

This Web Part fetches product-related information, including all relationships. Typically, this is used as an invisible part on a page that is used as the product data source by other consumer Web Parts on the same page.

This Web Part implementation in the Commerce Server 2009 R1 and R2 fetches only the “cross sell” relationships and not custom relationships configured via Catalog Manager. We had a requirement to configure and display custom relationships across products. This is a fetch behavior change that cannot be facilitated just by an HTML or XSL change, and therefore we chose to extend the available Web Part.

Required Changes

- Changes to the controller to modify queries to fetch custom relationships.
- Because the controller will change, the common project needs to be re-compiled with a new key.
- Based on the dependency described above, downstream changes are required and the ASP.NET and SharePoint Web Part projects must be recompiled.

Additionally, the **ProductXML** class that constructs the XML consumed by the XSL must be changed to accommodate custom relationships.

Steps Involved

10. Take existing SharePoint Commerce Services source code, and create a new Visual Studio project.
11. Make necessary namespace and/or library name changes.
12. Include only modified and necessary code; the rest can refer to the original library.
13. Sign with the new key.
14. Deploy in the global assembly cache (GAC).
15. Create an entry in the safe controls list in the Web.config file.

16. Deploy in the Office SharePoint Services Web Part gallery.
17. Use the new Web Part in pages.

Extending the Product Query Web Part

The Product Query Web Part helps to fetch data from a specific catalog and category, based on a category name or ID and the pagination information. In the implementation, the category home page was using this Web Part to display products within a category. The same page needed another Product Query Web Part to display featured products within a category from another virtual catalog. However, there was a design time limitation hard-wired into the Product Query Web Part that prevents coexistence of more than one of this Web Part on the same page. Additionally, there was a functional need to use the same Web Part on the home page and on other pages, such as the category home page with different display items. This required applying a different query based on which page was hosting the Web Part.

Required Changes

- Changes to the Web Part code to remove the design time check.
- This requires recompilation and resigning of the ASP.NET Web Part project and consequently the SharePoint Web Part project.
- Additionally, for the changes to the query based on page, the controller must be changed to apply the category filter, if available. This requires recompilation and re-signing of the common project which holds all the controllers.

Steps Involved

18. Take the existing SharePoint Commerce Services source code, and create a new Visual Studio project.
19. Make necessary namespace and/or library name changes.
20. Include only modified and necessary code; the rest can refer to the original library.
21. Sign with the new key.
22. Deploy in the GAC.
23. Create an entry in the safe controls list in the Web.config file.
24. Deploy in the MOSS Web Part gallery.
25. Use the new Web Part in pages.

Extending the Order History Web Part

The order history Web Part fetches the list of previous orders for the logged in user, and displays them in a paginated display. The sort order of the order records in the display is not based on a specific property. Our implementation required the order history to be sorted by order date.

Additionally, the Web Part does not display any records when the page is loaded. Records are displayed only after the user selects the date range filter criteria and submits the search. In our

implementation, we required the Web Part to display order history for a specific date range (for example, one month) by default.

Required Changes

- Changes to the fetch and display behavior of the Web Part. The changes cannot be accomplished just by HTML and XSL changes, and requires an extension.
- Changes to the controller to implement a sort order based on order date.
- Re-compilation and re-signing of the common project, and subsequently the ASP.NET Web Part and SharePoint Web Part projects.
- Changes to the View (User Control) to default to a specific date range.

Steps Involved

26. Take the existing SharePoint Commerce Services source code, and create a new Visual Studio project.
27. Make necessary namespace and/or library name changes.
28. Include only modified and necessary code; the rest can refer to the original library.
29. Sign with the new key.
30. Deploy in the GAC.
31. Create an entry in the safe controls list in the Web.config.
32. Deploy in the MOSS Web Part gallery.
33. Use the new Web Part in pages.

Commerce Server Configuration File Settings

Commerce Server architecture provides for configuration across the layers for controlling the functional and non-functional behavior of the system. The key configuration files related to the development and deployment of a solution based on this architecture are:

- Web.config – a regular ASP.NET Web site configuration file supplemented with sections specific to SharePoint and Commerce Server
- ChannelConfiguration.config – Configuration information related to commerce foundation services operation sequences.
- Metadatadefinitions.xml – Configuration of commerce entities exposed and used by commerce foundation services.
- OrderObjectMappings.xml – Entity mapping for order objects.
- Order PipelineMappings.xml – Entity dictionary mapping for pipeline access.

Channelconfiguration.config and MetadataDefinitions.xml are new files introduced in Commerce Sever 2009.

Channelconfiguration.config caters to:

- Operation sequences and the components in each sequence (per channel)
- Translators – from Commerce entity to Commerce Server Core entity
- Link to metadata definitions
- Authorization store for securing operation sequences related catalog, profiles, and orders

For more information, see [Commerce Foundation Operation Sequences](http://msdn.microsoft.com/en-us/library/dd442376(CS.90).aspx) ([http://msdn.microsoft.com/en-us/library/dd442376\(CS.90\).aspx](http://msdn.microsoft.com/en-us/library/dd442376(CS.90).aspx)).

MetadataDefinitions.xml caters to:

- Claims that are required to execute operation sequences
- For each commerce entity, specifies the various properties, relationships, and the claims that are needed to access them

For more information, see [Commerce Foundation Metadata Repository Design](http://msdn.microsoft.com/en-us/library/dd452304(CS.90).aspx) ([http://msdn.microsoft.com/en-us/library/dd452304\(CS.90\).aspx](http://msdn.microsoft.com/en-us/library/dd452304(CS.90).aspx)).

For more information about the other configuration files, see [Understanding Commerce Server Configuration](http://msdn.microsoft.com/en-us/library/aa545252(CS.70).aspx) ([http://msdn.microsoft.com/en-us/library/aa545252\(CS.70\).aspx](http://msdn.microsoft.com/en-us/library/aa545252(CS.70).aspx)).

For a typical Commerce Server deployment, the default values could be used—to some extent. However, specific changes have to be done based on the design and deployment needs. The typical configuration areas that need to be addressed for the e-commerce implementation are listed in Table 9.

Table 9: Typical Configuration Value Changes

Configuration area	Web.config	Channel configuration	Metadata configuration	Order object mappings	Order pipeline mappings
Web site security	Authentication mode and membership provider configuration Role provider configuration				
Service Security	Service binding and security configuration (This will be the Web.config of the WCF hosting)				
Performance	ASP.NET and Office SharePoint Services performance tuning				

	configurations				
Operation Sequence		Operation sequence components configuration Translator configuration Authorization repository connection			
Foundation Service invocation	Commerce Operation Service configuration to indicate usage of WCF Client end point (WCF) configuration in case WCF is used				
Deploying a new Web Part	Safe Control Entry	Related updates to operation sequence configuration if the Web Part introduces a new functionality			
Enabling diagnostics	Appropriate trace listener configuration for system diagnostics Enable Commerce Services global tracing Enable pipeline diagnostics Configure debug level				
Commerce server connectivity	Commerce Server site name configuration				
Pipeline	*.pcf files that are to be configured	Configure references of			Update to dictionary

		pipelines used by specific operation sequences in respective sections			mapping if any
Caching	Configure SharePoint cache Configure Commerce Server cache				
Data capture for reporting	Configuration to enable commerce events				
Commerce entity Definition Ref: http://msdn.microsoft.com/en-us/library/dd442315(CS.90).aspx			Configure commerce entities, properties, and relationships to be exposed and used by Commerce Foundation Services Access security for commerce entities, properties, and relationships		
Changes to order structure or its mappings				Tables, classes, and mapping	
Changes to order pipeline Dictionary structure					Classes, properties and mapping to dictionary keys

In addition to the configuration files, the following files can be used to manage configurations related to the Commerce Server Core Web services. These are the Web services used by the Commerce Server Manager applications and the Commerce Server adapters.

- Regular Web.config entries related to ASMX Web service hosting.

- Configuration in xx_PresentationInfo.xml. A variation of this file exists for each language. (for example, en_PresentationInfo.xml, fr_PresentationInfo.xml)

This configuration needs to be updated for customizations to properties that must be handled via the Web services from the management applications, Commerce Server adapters, or any other client that is using these services.

Search Integration

The technology stack provides for multiple options for deploying search capability in the solution. The key search types required for the solution are catalog search and content search. Additionally, search across user profiles and user transactions also may be required as part of end user and management user functionality.

Catalog and transactional searches will need to scan and search data primarily from the commerce repository and optionally from back-end systems. The content search will be mainly across the Web site content and any other supplementary source that is related to the catalog data – such as product manuals.

The search against the commerce repository is supported by the foundation services. The service will fetch the right records based on the input search criteria and the catalog schema configuration.

The search against the site content can be done using one of three technologies—Office SharePoint Services 2007 search, Microsoft Search Server, or Microsoft FAST—depending on the specific search needs.

A typical full deployment will use Office SharePoint Server 2007 for content management and delivery. In this case, Office SharePoint Services 2007 search can be employed within the same environment. Office SharePoint Services provides enterprise scale crawling, indexing, and query services that can be deployed in different scalable configurations. This supports search federation as well as search across multiple other business applications using standard interfaces.

If a different platform component is used for content management and delivery (for example, a third-party component), then it depends on whether the platform itself provides an effective content search mechanism. An appropriate solution in this case, or in cases where the front end is built from plain ASP.NET, is a solution based on Microsoft FAST ESP or Microsoft Search Server. Microsoft Search Server provides enterprise search capabilities with indexing, search federation, connectors, content filters, and query services that can be deployed in scalable configurations. The Microsoft FAST ESP can be used in scenarios similar to either of the above, but with additional and tighter search, performance, and availability requirements.

For more information about Microsoft search solutions, and when to use which solution, see the [Enterprise Search Product Overview](http://www.microsoft.com/enterprisesearch/en/us/products.aspx) (<http://www.microsoft.com/enterprisesearch/en/us/products.aspx>).

The search Web Parts in the SharePoint Commerce Services performs content search against a specific, single search scope. This search scope must be configured to point to the relevant content rather than index the whole site. The SharePoint Commerce Services search Web Parts can be customized to change the display look and feel of the search experience. If more advanced

configuration involving multiple scopes and searches across multiple scope is required, then this will require extending the SharePoint Commerce Services Web Parts.

To enable effective search in the solution implementation, complete the following steps:

34. Ensure appropriate catalog configuration to enable search across the required properties.
35. Deploy the chosen indexing and search solution (Office SharePoint Services 2007, FAST ESP, or other).
36. Plan and configure appropriate search scope(s) within the search solution to control indexing to focus on relevant content.
37. Use the SharePoint Commerce Services Search Web Parts (Search box, Search paging, and Search Results) to provide search functionality.
38. Customize or extend the search Web Parts to provide additional functionality or to search across different sources or scope or support search solutions other than Office SharePoint Services.

Commerce Server Adapters

Commerce Server 2009 provides adapters that can be used with BizTalk Server to exchange data. This mechanism can be used to integrate line-of-business and legacy systems with your digital marketing or e-commerce solution.

The adapters are designed to work in a loosely coupled architecture. The receive adapters are configured to pull data from Commerce Server into BizTalk processing, and the send adapters are used for pushing data from BizTalk processing back into Commerce Server. Based on your integration needs, the specific orchestrations required are to be identified and designed. The receiving and sending ends of the orchestration should then be configured to use the appropriate Commerce Server adapter.

Sending Data from Commerce Server

Receive adapters are used for sending data from Commerce Server. Receive adapters work in an asynchronous scheduled mode. The adapters can periodically check for updates by using the Commerce Server Web services, and pull data in bulk or incrementally, based on the configuration. An orchestration should be configured with the corresponding Commerce Server receive adapter scheduled for the appropriate time interval. The operations involved in the adapter and further processing is isolated from the Web site transactions. For example, a Web site transaction to update order information completes successfully by writing the data into the Commerce Server repository. The next scheduled run involving the Commerce Server adapter will pick the order information and send it to BizTalk for further processing, while the Web site operations will be proceeding without waiting for this processing.

If the Commerce Server Web services are not available when the adapter tries to invoke them, the specific attempt fails, but the data is processed in the next successful attempt.

If a synchronous transactional flow-through to an external system is required, the integration should deploy a more synchronous mechanism such as calling the external system services directly or through an orchestration that is published as a service. A good approach is to have the appropriate

customization done in the respective operation sequence to invoke the relevant Web service – either from the external system or through the BizTalk orchestration. A direct service call to the external system is simpler and can be employed when there is no complex transformation involved, the system to be interfaced supports standard Web service interfaces already, and the data updates are not always transactional. BizTalk Server orchestration–based integration provides the additional benefits of transactional support, transformation, legacy adapters, tracking, and reporting.

Sending Data to Commerce Server

The send adapters can be used for sending data to Commerce Server from BizTalk message processing or orchestration. The send adapters call into the Commerce Server Web services and support transactions via the Web services.

Due to the synchronous nature of the send adapters, synchronous flow from an external system can be processed within a BizTalk orchestration that uses the adapter within the rules of BizTalk processing. The transactional capabilities are enabled (and limited) by what the Commerce Server Web services support.

If the Commerce Server Web services are unavailable, then the adapter fails and so does the orchestration. A proper retry mechanism must be put in place as part of the design either within the BizTalk environment or on the other end of the orchestration.

Data Processing by the Commerce Server Adapters

The different commerce server adapters provide built-in support for processing records in different modes. For example, the profile adapter (receive) supports an incremental and a bulk fetch mode. In an incremental fetch mode, the receive adapters fetches only the relative changes since the previous fetch. In a full fetch mode, the adapter fetches the full data related to the subsystem.

Some of the adapters also can process data in individual or bulk mode. For example, in individual processing mode, an order receive adapter sends one record at a time to BizTalk processing, and in the bulk record processing mode, the receive adapter sends all orders in one message to BizTalk processing.

As part of the integration design, you must determine which mode to enable based on nature of the data and the processing requirements. The other factors to consider are the size of the data and the level of transaction required.

The adapters can be used in standard BizTalk orchestration configurations to support long running transactions and employ correlations to process staggered responses.