



Architecture Guide

AgilePoint BPMS v5.0 SP2

Document Revision r5.2.5

November 2011

Contents

- Preface.....4**
 - Disclaimer of Warranty.....4
 - Copyright.....4
 - Trademarks.....4
 - Government Rights Legend.....4
 - Virus-free software policy.....4
 - Document Revision Numbers.....5
 - AgilePoint Documentation in PDF and HTML.....5
 - Contacting AgilePoint Sales.....5
 - Contacting Customer Support.....6
- Introduction.....7**
 - Relationships between Entities.....7
 - AgilePoint Business Process Data Flow.....8
 - Build Time.....9
 - Run Time (AgilePoint Server).....9
 - Run Time (Application).....10
- Software Architecture.....12**
 - Presentation Layer.....12
 - Logic Layer (AgilePoint Server).....13
 - Process Engine.....13
 - Session Manager.....13
 - Request Queue Manager.....14
 - Process Template Class Loader.....14
 - Process Template Management.....14
 - Archive Manager.....14
 - Process Swapper.....14
 - Server Coordinator.....15
 - Escalation Monitor.....15
 - Notification Preprocessor.....15
 - Notification Deliver.....15
 - Exception Handling.....16
 - AgilePart.....16
 - Data Layer.....16
 - Server (Business Components) Interactions.....17
- Implementation Architecture.....18**
 - Autonomous BPMS.....18
 - Embedded BPMS.....18
 - Rich Client Implementation.....19
- Deployment Architecture.....20**
 - Deployment/Installation Architecture Model.....20
 - Small Scale Model.....21
 - Medium Scale Model.....22
 - Large Scale Model.....23
 - AgilePoint and Your Existing Environment.....24

Network Performance.....	24
Domain Name Service (DNS).....	25
Authentication, Active Directory and Kerberos.....	25
Hardware and Software Configuration.....	25
Strategy for Development, Testing and Staging.....	25
Nonfunctional Requirements for AgilePoint Application.....	25
Development, Testing, Staging and Production Environment.....	26
Disaster Recovery.....	28
Single AgilePoint Server Instance.....	28
Multiple AgilePoint Server Instances.....	29
Performance.....	31
Security.....	32
Authentication.....	32
User Interface Authentication.....	32
Server Process Authentication.....	32
Database Authentication.....	32
Authorization.....	33
User Access Security.....	33
Active Directory Integration.....	33

Preface

Disclaimer of Warranty

AgilePoint, Inc. makes no representations or warranties, either express or implied, by or with respect to anything in this document, and shall not be liable for any implied warranties of merchantability or fitness for a particular purpose or for any indirect, special or consequential damages.

Copyright

Copyright © 2011 AgilePoint, Inc. All rights reserved.

Trademarks

AgilePoint, Inc. and AgilePoint's products are trademarks of AgilePoint Inc. References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

Government Rights Legend

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the applicable license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1995), DFARS 252.227-7013(c)(1)(ii) (Oct 1988), FAR 12.212(a) (1995), FAR 52.227-19, or FAR 52.227-14, as applicable.

Virus-free software policy

AgilePoint recognizes that viruses are a significant security consideration for our customers. To date, we have had no report of AgilePoint BPMS carries any virus. AgilePoint takes the following measures to ensure our software is free of viruses upon delivery:

- AgilePoint is built on top of Microsoft .NET framework. The pre-compiled executable is a .NET Common Language Runtime (CLR) application, not a native machine binary. As far as is known at this time, there are no viruses that infect .NET CLR executables.
- The virtual environment for the product packaging process is fully isolated and protected, and anti-virus software is installed and running during packaging.
- The deliverable package is scanned by anti-virus software before upload to our customer download site.

Document Revision Numbers

AgilePoint documentation uses the revision number format **rX.Y.Z**. The letters and numbers in this revision number can be interpreted as follows:

- **r** - Indicates "revision." This helps to differentiate the document *version* numbers, which start with **v**.
- **X** - The major version number for AgilePoint BPMS to which this document refers. For example, AgilePoint releases 5.0, 5.0 SP1, and 5.5 would all have an **X** value of **5**.
- **Y** - The major document revision number. This number typically changes only when either there is a new AgilePoint release, or there are major changes to the document.
- **Z** - The minor document revision number. This number is incremented each time the document is republished.

AgilePoint Documentation in PDF and HTML

AgilePoint documentation is provided in both print-friendly (PDF) and web-based (HTML) formats.

Advantages of HTML Documentation

- HTML is the **primary delivery format** for AgilePoint documentation.
- Unified, global **search** across all documentation. PDF documents allow you to search only within the context of a given PDF file.
- **All hyperlinks supported**. Links in PDFs are only supported in certain contexts.
- "One-stop shopping" for all information related to AgilePoint BPMS.
- The HTML documentation is updated more frequently than the PDF documentation. Web-based documentation is updated periodically between AgilePoint releases to address errors and omissions, but the PDF documentation is updated only at the time of a software release.

Advantages of PDF Documentation

PDFs can be more easily **printed**, **archived**, and **transferred** (such as by FTP or email) than HTML documentation.

For more information, see [Downloading Files and Sharing Links from the Documentation Library](#) on the [AgilePoint Support Portal](#).

Contacting AgilePoint Sales

AgilePoint is a leading Business Process Management System (BPMS) provider created by a team of driven people who strive to incorporate the principles of relentless innovation for the benefit of our customers. Our mission is to help companies of any size attain and sustain operational success through process excellence.

Headquarters: AgilePoint Corporation 1916C Old Middlefield Way Mountain View, CA 94043, USA

Tel: (650) 968 - 6789

Fax: (650) 968 - 6785

Email: info@agilepoint.com

Web site: www.agilepoint.com

International: For AgilePoint EMEA and AgilePoint Asia Pacific, please call the AgilePoint Corporate Office for contact information.

Contacting Customer Support

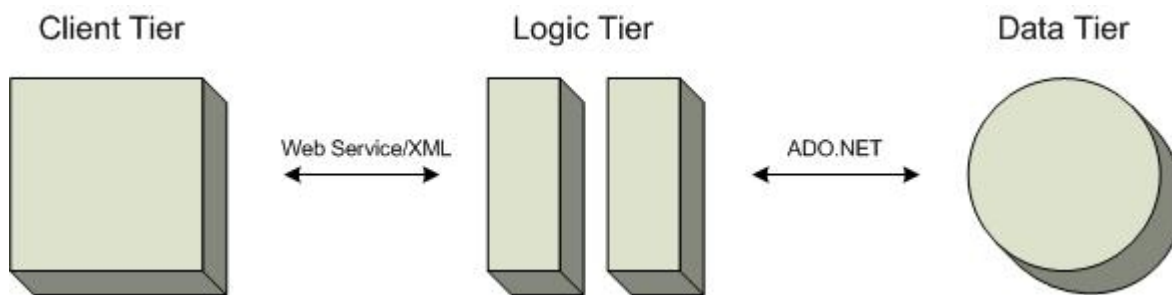
To contact AgilePoint Support, please submit a ticket on the AgilePoint Support Portal: <http://support.agilepoint.com/SupportPortal/>

If you do not have a Support Portal account, you can send an email to request one: support@agilepoint.com

Introduction

This document describes the architecture of the AgilePoint BPMS (Business Process Management System). AgilePoint is a Microsoft .NET based BPMS that provides a robust and scalable Business Process Management engine to support both human workflow and automated processes through reusable business process rules and process assets. AgilePoint's process engine is fully WfMC (Workflow Management Coalition) compliant with support for XPDL and ingrained XML. The AgilePoint XML based process description is fully translatable into other emerging standards such as BPEL and BPML.

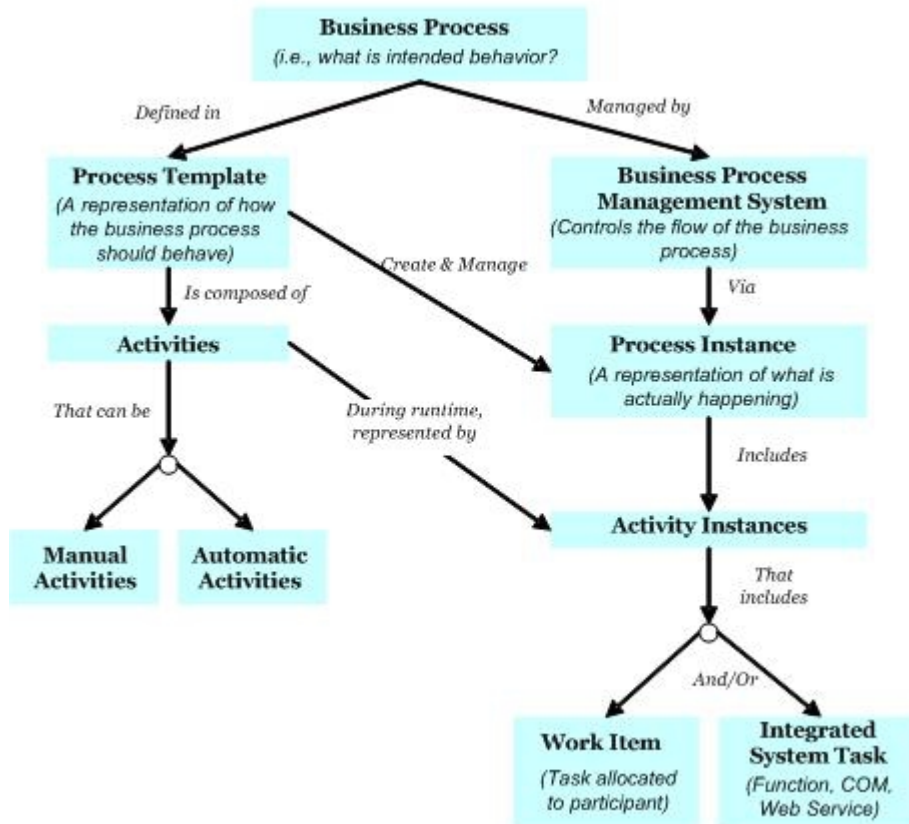
AgilePoint has adopted 3-tier architecture which is highly scalable (3-tier which can be scaled to n-tier), portable (through Web Service) and flexible (process component reusability). Data persistence is managed in the database. This layered architecture is used to isolate changes and to provide for easy integration among different technologies. The following diagram provides a simple view of AgilePoint's 3-tier architecture. Detail description of components and its overall architecture is presented later in this document.



Relationships between Entities

A process is defined in a process template # an XML representation of the process. The process template is deployed and managed by the AgilePoint BPMS. The XML entities within the process template define the flow of the business process. AgilePoint BPMS will initiate process instances and their corresponding activities based on the entity definitions within the process template.

The following diagram represents how the above entities are related in AgilePoint BPMS.

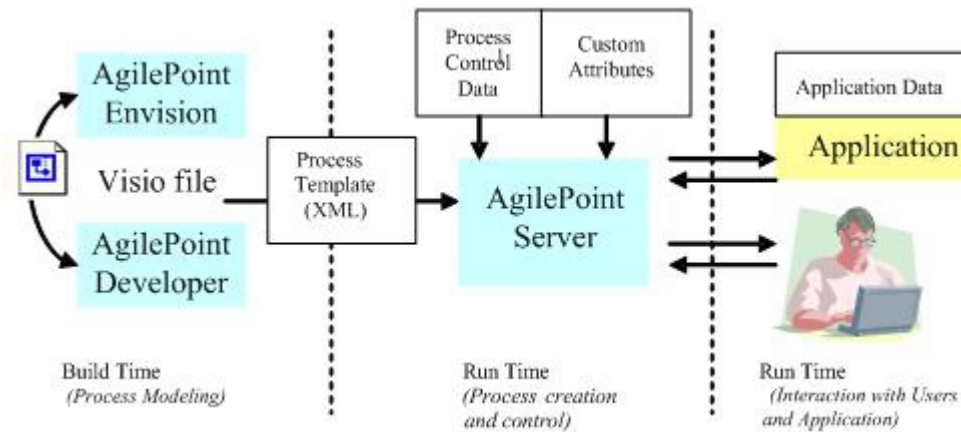


AgilePoint Business Process Data Flow

In an AgilePoint-enabled application, various types of data will flow through AgilePoint Server:

- **Process Definition** – Process template that defines the process and will be deployed to AgilePoint Server in XML format.
- **Process Control Data and Custom Attributes** – Process-related data that AgilePoint Server uses to manage, monitor, and audit the processes.
- **Application Data** – Application-related data that AgilePoint Server will carry and pass to the application to perform further application logic.

The following diagram represents the overall process data flow:



Build Time

Users build an AgilePoint process model using AgilePoint Envision, which is built on Microsoft Visio. AgilePoint is the first BPMS to extend Visio into a full lifecycle BPM productivity tool, from design to production.

- The created process model can be saved in Microsoft Visio format to facilitate sharing and collaboration throughout the design process.
- Process models created in AgilePoint Envision can be deployed directly to AgilePoint Server, the BPM engine, for execution. To facilitate enterprise deployment methodology, AgilePoint Envision supports the deployment to multiple target server environments, such as development, testing, staging, and production.
- As part of the deployment process to AgilePoint Server for execution, the process model is converted into XML format.

When customization and/or IT abstraction is need, application developers can then use Microsoft Visual Studio .NET based AgilePoint Developer to perform the following:

- Create a standard Visual Studio .NET based AgilePoint project to add custom application code and integrate the application logic with the associated process models.
- Create AgileParts to abstract IT functionality for direct leverage and reuse at the process level.

By enabling full customization through associating a Visual Studio .NET project with a Visio based process model, AgilePoint separates the process definition from underlying code and technologies. This is a compelling advance over competing BPMS products. Like its tiered architecture, this design facilitates flexibility, reusability and manageability that are key to power adaptive BPM process-based applications.

Run Time (AgilePoint Server)

Once an AgilePoint process model is deployed to AgilePoint Server and stored in XML format, AgilePoint Server responds to user requests through the user interface of AgilePoint enabled application, programmatically, or via external events to initiate an instance of the process mode, execute and control the created running process instance based on the definition in the process model.

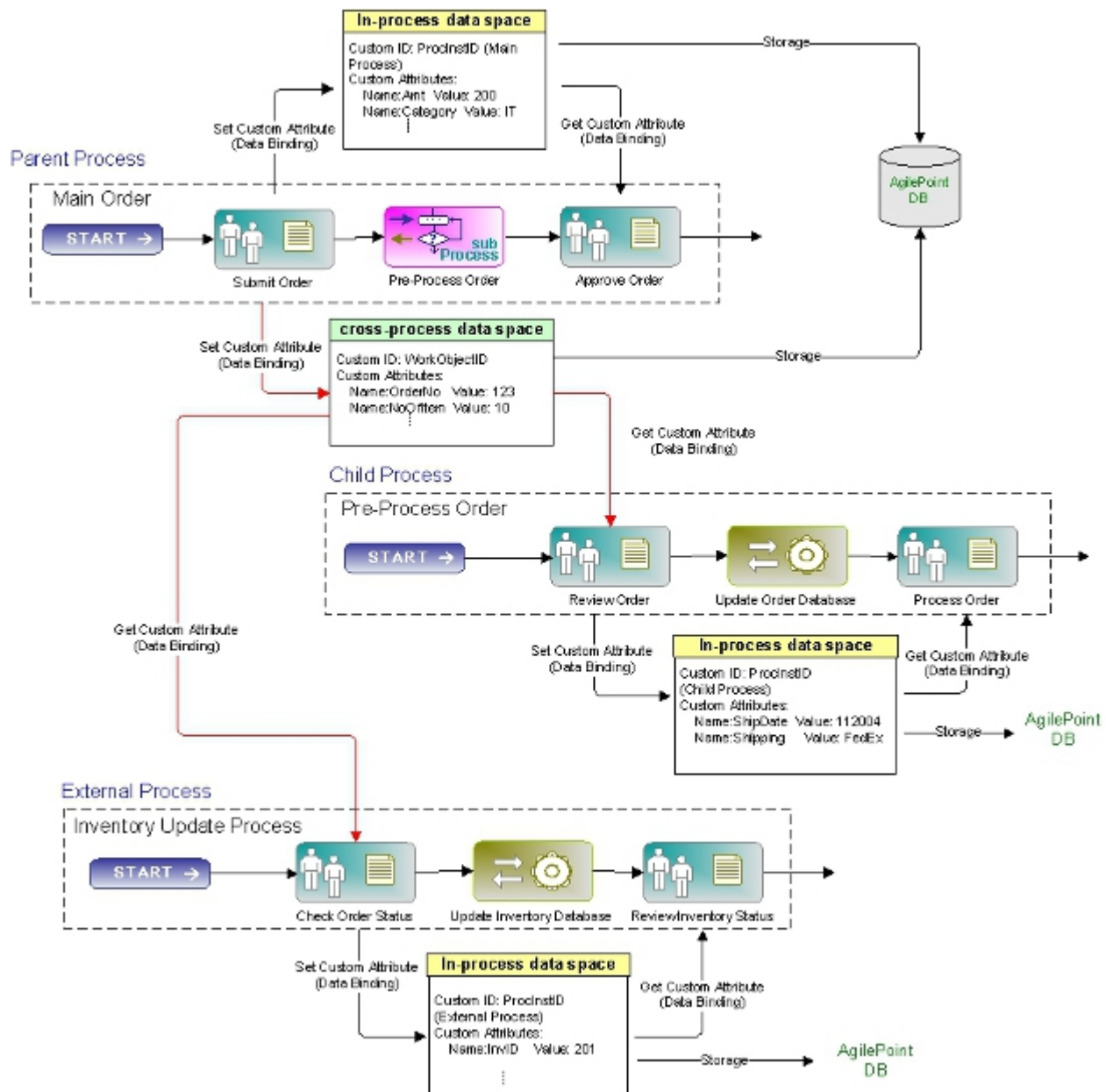
- Process control data is maintained and stored in the database. This data is used in process management, monitoring, and reporting.
- Custom attributes are used to maintain process related data within a process instance or across multiple process instances. This data is used by AgilePoint to perform specific data driven business logic such as decision making and conditional branching.

Activities are created where manual tasks are distributed to participant(s) to perform the specific tasks through the application interface and automatic system tasks will be triggered through the application integration.

Run Time (Application)

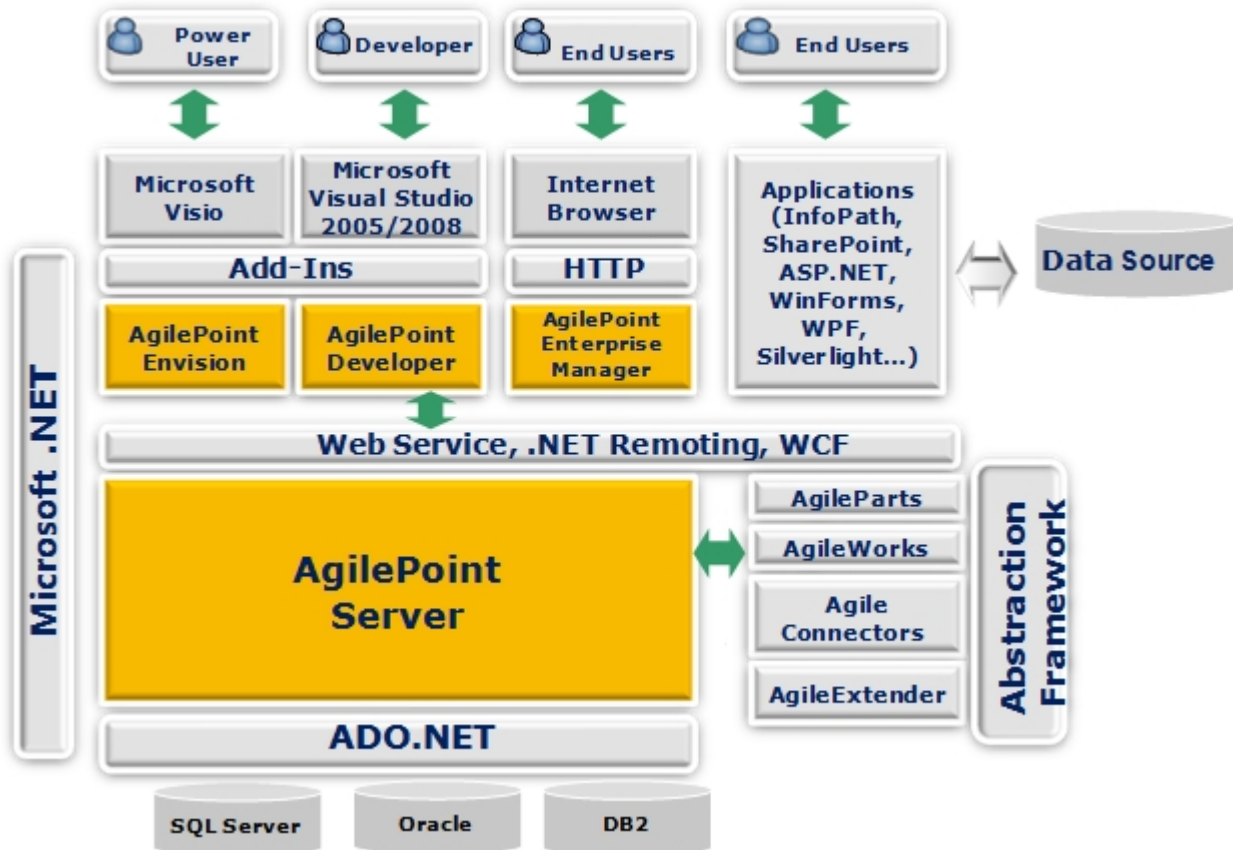
- AgilePoint-enabled applications provide interfaces for users to perform the manual activities. They also manage, monitor, and audit the processes.
- Application data can be stored in the database or other format, depending on the architecture and technology of the application. When database is used, the application data can be stored in a separate database from the AgilePoint Server database. This application data will be used by the application to perform specific application logic.

Data Management in AgilePoint (Logical View)



Software Architecture

The AgilePoint BPMS 3-tier architecture is detailed in the following diagram.



Presentation Layer

There are two types of presentation layers: AgilePoint's own presentation layer and an AgilePoint-enabled application's presentation layer.

- The AgilePoint Presentation Layer provides the interfaces for BPMS implementers to design, develop, execute, manage, and monitor the process and its instances. AgilePoint has three presentation components:
 - **AgilePoint Envision** – A Microsoft Visio add-in through COM technology where process designers can define their process templates with the Microsoft Visio interface.
 - **AgilePoint Developer** – A Microsoft Visual Studio .NET add-in through COM technology where application developers can implement the process and application logic through the Microsoft Visual Studio .NET.

- **AgilePoint Enterprise Manager** – A web-based interface that power users or system administrators can manage, monitor, and audit the AgilePoint BPMS and its runtime processes.
- **AgilePoint Server Configuration** – A desktop-based configuration tool for AgilePoint system administrator to configure the various system parameters such as SMTP server and database settings.
- The Application Presentation Layer provides the interfaces for business process end users to initiate processes and/or perform the activities defined in the processes. AgilePoint provides flexible architecture that various types of application presentation can be implemented through the AgilePoint Web Service Client API.

Logic Layer (AgilePoint Server)

AgilePoint's Logic layer provides the core functionality and BPMS logic for AgilePoint Server. There are multiple components, each of which provides its own set of features and functionalities.

The logic layer can be accessed through the client-side Web Service API or the Server-side API.

Process Engine

The Process Engine provides the core engine to handle the process flow.

- The Process Engine is the core of the AgilePoint Server. It is a WfMC compliance process execution engine with an XPDL extension and AgilePoint's own ingrained XML support. It is capable of consuming the XML-based process definitions designed and deployed through AgilePoint Envision.
- The engine executes process instances and performs the corresponding activities based on the definitions and business rules in the process templates. The activities include manual activities that require human intervention, automatic activities that trigger corresponding system tasks, and conditional activities that determine the branching and flow dynamics of the processes. The engine is responsible for routing manual tasks to appropriate participants and associate with corresponding tasks.
- The engine is capable of performing dynamic binding and loading during execution at run-time offering the maximum flexibility and agility.

Session Manager

The Session Manager handles the client application connection session including security authentication and authorization. This component is also responsible for monitoring the behavior of the activities throughout the process and keeping track of audit trail of the process. The related information is maintained in the AgilePoint database and can be used for reporting.

Request Queue Manager

The RequestQueue Manager component manages task requests. It maintains the multi-threaded asynchronous execution of a series of task requests, which can be delegated, reassigned, suspend, resumed, and canceled. These operations can be preset through programs, predefined instructions such as delegation rules, or they can be triggered manually through the Enterprise Manager.

Process Template Class Loader

- This component handles the loading of the process template and its execution 'instructions.' For example, the 'instructions' could be the Visual Studio.NET project containing the custom code to control the runtime behavior of the process instance. The process template definition and instructions will be fed to the process engine to initiate and manage the execution of the process instance.
- This feature enables the separation of process definition from underlying code providing new levels of runtime control, flexibility, and manageability.

Process Template Management

The Process Template Management component manages the lifecycle of the process templates. AgilePoint supports built-in versioning control for the lifecycle development of process template. A process template has different stages throughout its life-cycle such as: created, released, checked-out, checked-in, and retired. Only a "released" process template can be executed while the previous versions will be "retired". Process templates are maintained with unique process template names along with version and status. System Administrators or users with authorized permissions have full controls of the deployment and versioning of all the process templates. Process templates be easily managed with AgilePoint Enterprise Manager.

Archive Manager

The Archive Manager module handles the versioning and data backup of process-related data. This component allows system administrator to back up, restore, and archive process-related data.

Process Swapper

The Process Swapper component handles process swapping that enhances performance and resource handling. AgilePoint Server can maximize the capacity of the physical server by the process swapping technology. Many long-running processes may take hours, days, or weeks to complete between steps. AgilePoint Server exploits an intelligent algorithm to swap these processes out from system memory to maximize the usage of system resources. This module monitors the status of the process instances, and it releases and allocates the resources of idle process instances to active process instances. Though swapped out, the idle processes (in "sleep" state), are still controlled by AgilePoint timing sensitive modules such as timeout activities and email notifications. When external or internal events to which idle processes listen occur, these processes can be swapped back into the memory instantly.

Server Coordinator

This component manages the server and system configuration. A user interface **Server Configuration** is provided for System Administrator to specify the system and server configurations. These configurations included:

- Database Server configuration including database name, database authentication mechanism, and database connection pool setting.
- System administrator setting and authentication mechanism.
- SMTP Server and related settings such as email format and sender information.

The configuration is stored in an XML file that contains the above information.

In a multi-server deployment scenario, this component also manages the clustering and load balancing relationships and configurations among multiple servers.

Escalation Monitor

The Escalation Monitor module keeps track of whether an activity or a process is overdue and whether it requires escalation. In each process template, the process and each activities are defined with due dates which will be monitored by the Escalation Monitor. Specific instructions or activities can be implemented and triggered when the activities or the process are overdue.

Notification Preprocessor

- The Notification Preprocessor module preprocesses the email notifications based on the mail template definition and the runtime data of the process. For example, AgilePoint lets users define dynamic email templates containing macros that will generate dynamic contents based on runtime conditions.
- An activity can contain multiple email templates that correspond to different stages of the activity. The mail template for the email notification can contain dynamic runtime data such as the activity's participant information or the unique process instance name. These dynamic values will be incorporated into the final email notification before sending.

Notification Deliver

The Notification Deliver component manages the delivery session of email notifications. AgilePoint Server provides different types of email notifications depending on the stage and status of the activities. For example, multiple email notifications can be triggered depending on the stage of the execution of an activity such as "entering", "in-processing", or "exiting" the activity. Similar email notifications could also occur when the activity is overdue, or is being reassigned. The email notifications are sent through the SMTP server as specified during the server configuration. This module also keeps track of failed notifications that could be caused by temporarily unavailability of the SMTP server. The system will resend the failed notification in a predefined schedule or System administrators can also explicitly resend the failed notifications through the Enterprise Manager interface.

Exception Handling

This component contains the default exception handling module that is executed when a runtime exception occurs. Part of the default exception handling includes notifying system administrators of the exception through email. AgilePoint also lets users to replace the default exception handling with their custom exception handling routines to gain 100 percent control of handling run-time exceptions.

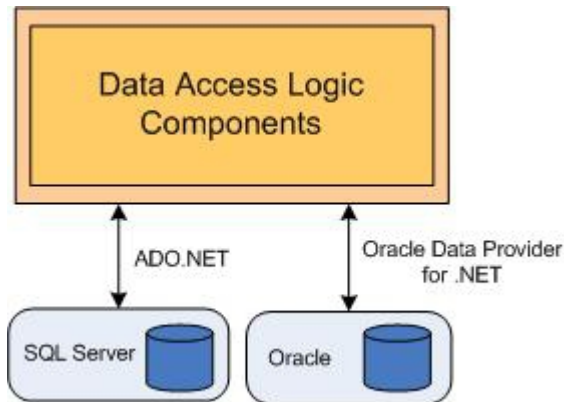
AgilePart

AgileParts enable the abstraction of IT functionality from the code level to the process level. In short, AgileParts are custom and reusable IT components that can be leveraged within or across AgilePoint processes. AgileParts contain custom code that performs specific tasks.

- AgileParts are exposed at the process level for business user consumption through AgilePart Registration support in AgilePoint Envision. They can be used as custom activities during the process template designing and modeling. This is a unique and powerful mechanism to extend and expose reusable custom code from the IT level into the process level to empower business users and further enable business agility. This advancement also significantly increases IT utilization and return of IT investment.
- To maximize the flexibility of reuse, the behavior of AgileParts can be easily configured by user input through the property interface in AgilePoint Envision.
- AgileParts can be used to encapsulate past (pre .NET) developments such as VB, COM, C++, or even Java and leverage them immediately to build .NET powered BPM applications in AgilePoint.
- AgileParts can be executed by the same system running AgilePoint Server. Though AgilePoint can "scale up" through clustering support to handle increasing load of AgilePart execution, AgilePoint also supports "scale out" to deploy AgileParts to be executed by its own dedicated server to offload AgilePoint Server for dedicated execution of process instances.
- When exposing IT functionality of different systems into AgilePart, it offers the advancement of the creation of a "virtual IT layer" to enable technology independent user interface and operations.

Data Layer

AgilePoint's Data layer handles database interaction where all process control and related data are stored. In AgilePoint, the logic necessary to access data is abstracted in a separate layer of data access logic components where different database access packages can be applied to access different types of data sources. AgilePoint currently supports both Microsoft SQL Server 2000 and Oracle databases through ADO.NET and Oracle Data Provider for .NET respectively. This architecture can be easily extended to support other data sources if needed. The following illustrates the AgilePoint's data layer:



The AgilePoint Data Access Logic component provides two types of interfaces:

- Web Service API that allows user interface (client) component
- Server side API that allows logic layer or business components

Server (Business Components) Interactions

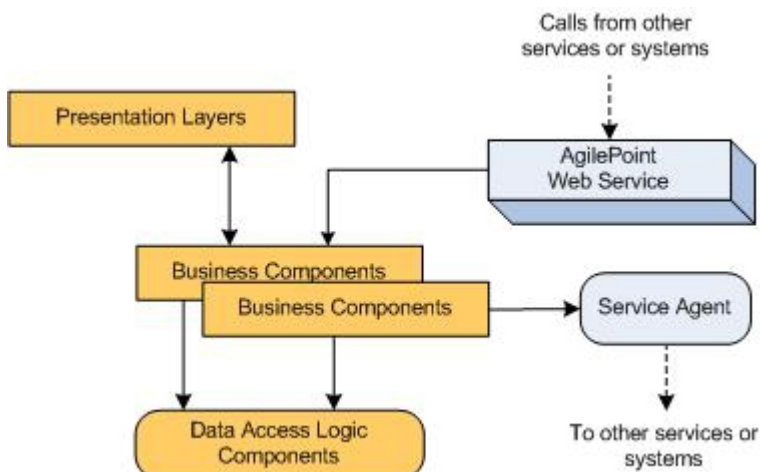
AgilePoint's Business Logic Layer contains business components that encapsulate the business logic. As part of the business logic layer, business components interact with both the Presentation and Data layers. There are 2 ways that AgilePoint's business components can be invoked:

- Through components in the Presentation Layer
- Through the AgilePoint Web Service interface

AgilePoint's business components call AgilePoint's data access logic components to insert, update, and retrieve data from the AgilePoint database.

For integration, AgilePoint's business components can invoke service agents to call the other services. This provides an AgilePoint enabled application flexibility to integrate with other systems or services.

The following diagram illustrates the interaction between AgilePoint business components and the Presentation and Data layers.



Implementation Architecture

AgilePoint BPMS is designed to support various types of integration and implementation, including web-based, WinForm based, and third-party integration. There are two main types of implementation approaches:

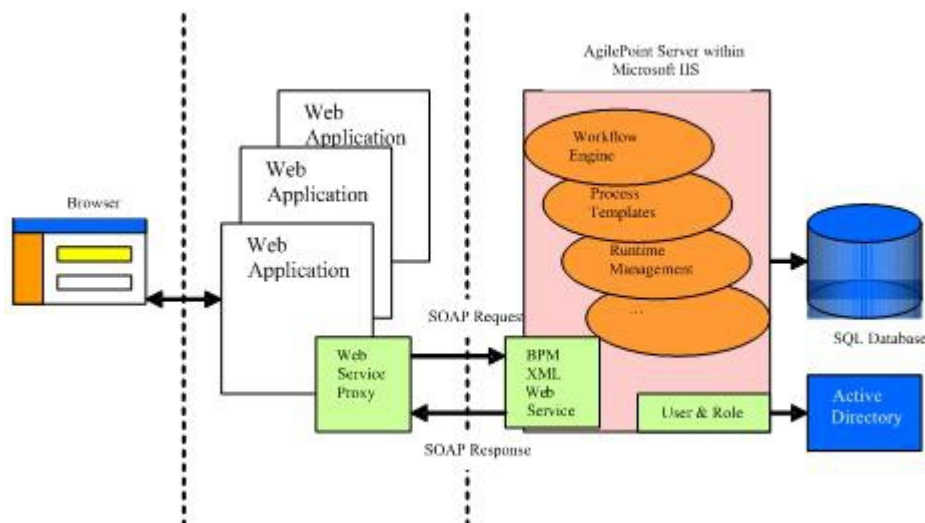
- Autonomous BPMS
- Embedded BPMS

Autonomous BPMS

In an Autonomous BPMS, an AgilePoint web service awaits a client to make a request. It then processes the request and sends a response message to the client. The client could be a typical end-user application # for example, Microsoft Internet Explorer (IE) for a web application.

Regardless of the type of client interface, one common theme is that a piece of code, called a proxy, is used for receiving requests and returning responses between client and server. Microsoft Visual Studio .NET provides a way to generate and update a proxy for a web service by a given server URL. AgilePoint Web Application Project type available through its integration with Visual Studio .NET facilitates the creation of an AgilePoint powered web application.

The following diagram illustrates a typical autonomous BPMS implementation.

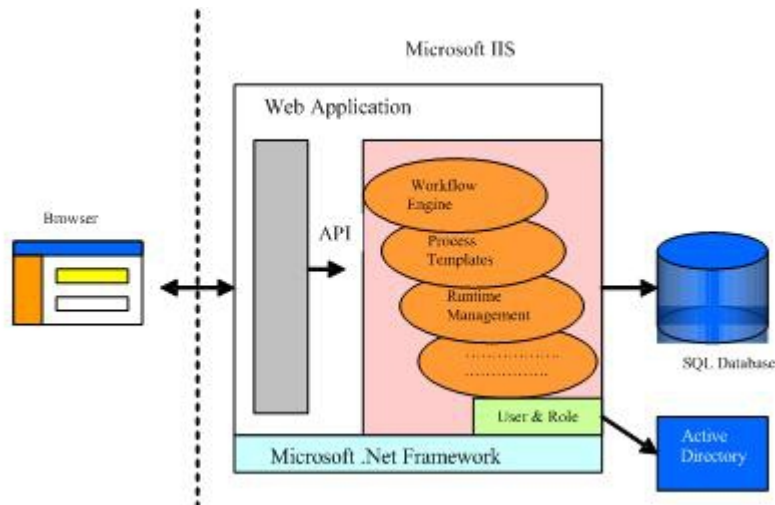


Embedded BPMS

In an embedded BPMS application, AgilePoint Server runs with the same process of the embedding application as a component. AgilePoint workflow application programming interface (WAPI) and AgilePoint Embedded Server Project type in Visual Studio .NET makes implementation and integration

of this type of application straightforward. An embedded BPMS application obtains more and direct controls on AgilePoint BPMS.

The following diagram illustrates a typical Embedded BPMS implementation:



Rich Client Implementation

Besides the web-based client, AgilePoint also supports rich client application interfaces. You can either create a rich client interface based on Windows form (WinForm) or integrate with existing an Microsoft application such as Microsoft Office.

In most enterprise application deployments, using rich client interfaces implies the needs of the following:

- The ability to authenticate users through Microsoft Active Directory
- The ability to work offline
- An interface with richer functionalities and session related state management.

AgilePoint's flexible architecture supports the rich client interface implementation in addition to the web-based application implementation.

Deployment Architecture

There are three basic AgilePoint deployment architecture models. Each model can be tuned to scale up and out for overall performance of the AgilePoint system. It guides the starting point of planning the deployment architecture of AgilePoint installation. Notice, the models described in this document may not be exactly the same as the actual system you are trying to build, but taking the model as a template will help the configuration of system environment.

Typically, AgilePoint deployment or installation falls into one of the following three architecture models:

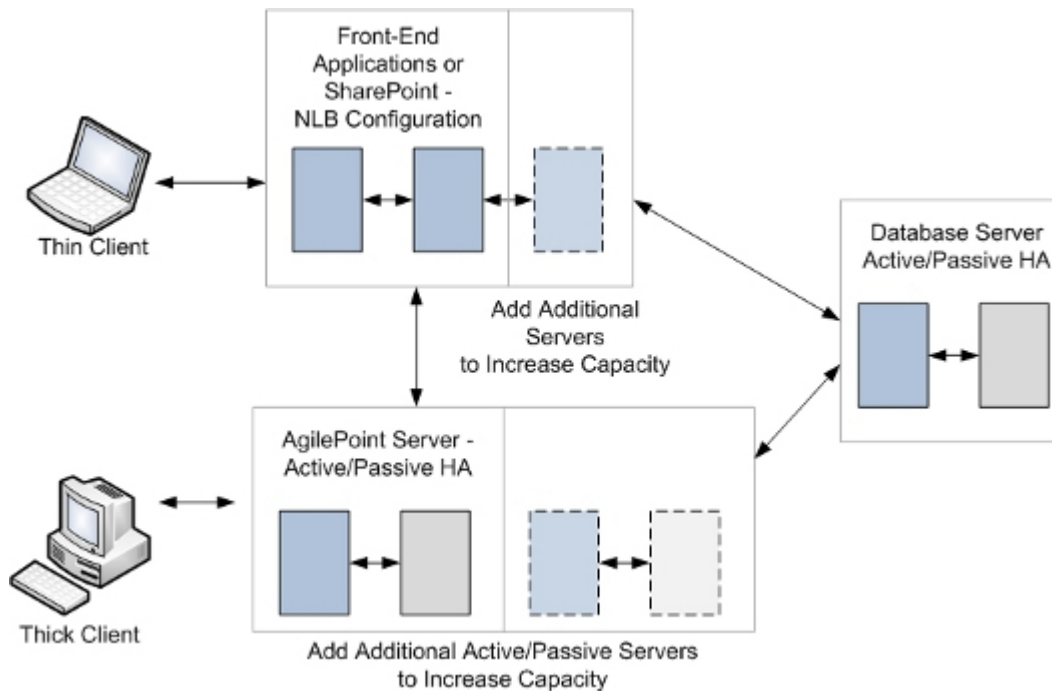
- Small Scale Model
- Medium Scale Model
- Large Scale Model

It covers most of deployments, but your system configuration may be slightly different. Meantime, you may take similar approaches to plan your staging environment.

Deployment/Installation Architecture Model

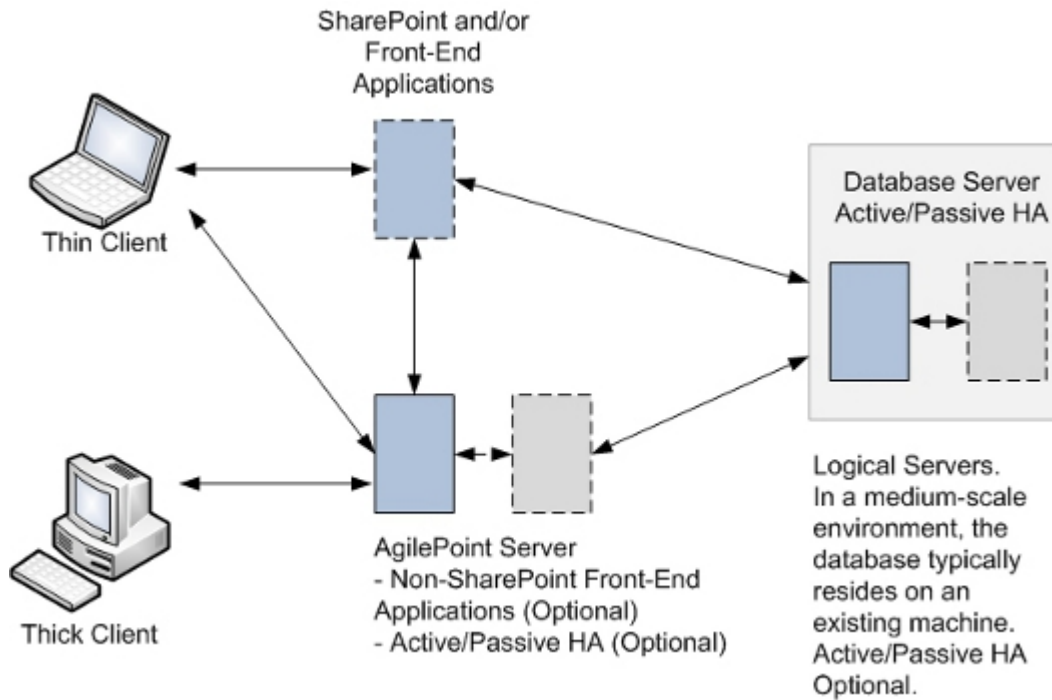
The AgilePoint deployment architecture varies based on the specific requirements for individual organizations. However, all deployment architecture recommendations are based on the same basic standards:

- **Front end application server (Presentation Layer)** - As the scale increases, presentation layer servers are typically added and load balanced using NLB.
- **AgilePoint servers (Business Logic Layer)** - Failover servers are added to the business logic layer to improve reliability, and additional servers are added to handle additional load.
- **Database server (Data Layer)** - In most small- to medium-scale AgilePoint implementations, the database resides on a database server that serves other applications as well. However, this can be a dedicated server, and it often is in enterprise implementations. Active/passive failover is recommended.



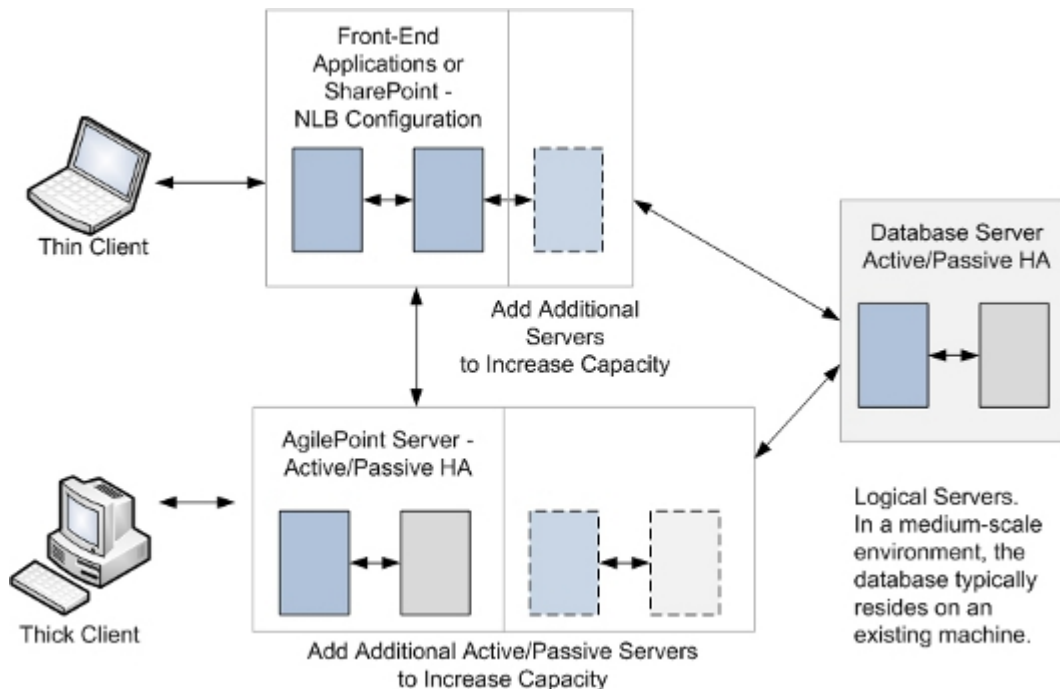
Small Scale Model

In a small-scale deployment architecture, the front-end application and AgilePoint Server are on one server, and the database server can be shared with other applications if process loading not heavy. However, since AgilePoint is a database-intensive server, a dedicated and faster database server will increase overall AgilePoint performance. (Note that in systems with minimum requirements of 4 GB memory, you may also want a separate machine to run SharePoint.) Small-scale architecture should be considered to be entry-level and departmental small business deployment



Medium Scale Model

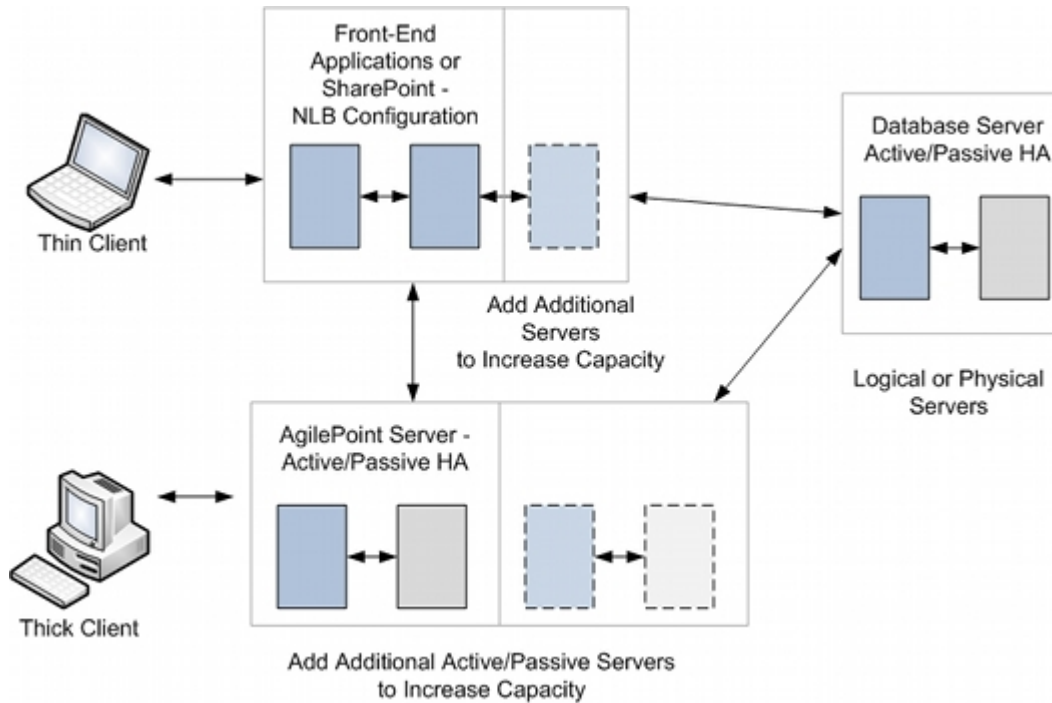
The medium-scale model is designed as 3-tier architecture. It provides data tier redundancy, front tier performance, and high availability. This model is considered the entry-level enterprise deployment architecture. Also, the architecture is quite easy to scale and handles a heavy loading by end user interaction, such as ASP.NET forms, SharePoint libraries, and reports.



Large Scale Model

For business-critical process-driven applications, as process load increasing, the large scale deployment model will be necessary to be planned and implemented for the enterprise or any organization. It provides not only layer isolation, but also three-tier high availability and performance by clustered database, AgilePoint and front-tier applications.

The architecture can be scaled out by simply adding nodes to existing NLB clusters. It can be scaled up by upgrading existing nodes (for example, increasing CPU and memory) and adding more instances of AgilePoint and web applications on each server.



AgilePoint and Your Existing Environment

Once you have selected AgilePoint deployment architecture model, next you must analyze and configure your existing infrastructure in order to deploy AgilePoint BPMS in your environment. There are many aspects you must take in account, such as: network, DNS, authentication/Kerberos/Active directory, IIS configuration, and database configuration.

Network Performance

Network performance is an important factor in preparing your existing environment for AgilePoint deployment.

Connectivity between Servers

From network topology point of view, the connections between servers are critical to the AgilePoint BPMS system. It is also important that the server uses a fast network interface card (NIC). Often AgilePoint Server requires a large amount of data communication from database server, so good network connectivity is required between the nodes. If SharePoint is installed, the SharePoint farm needs a fast network connection to the database. Understanding network connection constraints and server location should be part of AgilePoint deployment plan.

Connectivity between the Server and Workstation

Besides server performance, it is also very important to optimize the system for end user to have best possible experience. For large enterprise deployments, intranet users may have good connectivity, but Internet users do not. Typical issues would be network latency, user locations and network bandwidth. Usually, these issues follow a time pattern # for example, network traffic from 9 a.m. to 11 a.m.

Domain Name Service (DNS)

The Domain Name Service (DNS) is a hierarchical naming system for computers, services, or any resource participating in the intranet and internet. From distributed computing perspective, it plays important role as well to help with performance, reducing the amount of time for translating name to address. For example, host headers or aliases could be an issue if a DNS administrator does not register the qualified name. It is beneficial to consider DNS as part of AgilePoint deployment project.

Authentication, Active Directory and Kerberos

AgilePoint BPMS supports Microsoft Active Directory for authentication or non-Windows authentication. If AgilePoint is using Active Directory authentication, Active Directory server calls a domain controller assigned to the same site as AgilePoint Server. If AgilePoint uses non-Windows authentication, Active Directory is not required.

Kerberos is a computer network authentication protocol that allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner. It is not an additional component. It is there by default on Windows. Kerberos is one of way to solve **double-hop** authentication issues that are common in n-tier applications environments.

Hardware and Software Configuration

For basic hardware and software requirements for AgilePoint BPMS, see [System Requirements for .NET 3.5](#) on the [AgilePoint Support Portal](#).

Strategy for Development, Testing and Staging

Planning, implementing and deploying a AgilePoint BPMS solution is not only about having a running AgilePoint server and application, but also continual support, maintenance, improvement. It is also about optimization, development and policy. It requires to be supported by an effective environment configuration and infrastructure.

Nonfunctional Requirements for AgilePoint Application

Nonfunctional requirements directly impact your design and deployment strategy. However, many customers do not consider these requirements until they arise after the deployment. As part of your AgilePoint deployment plan, the following aspects are important to take in account:

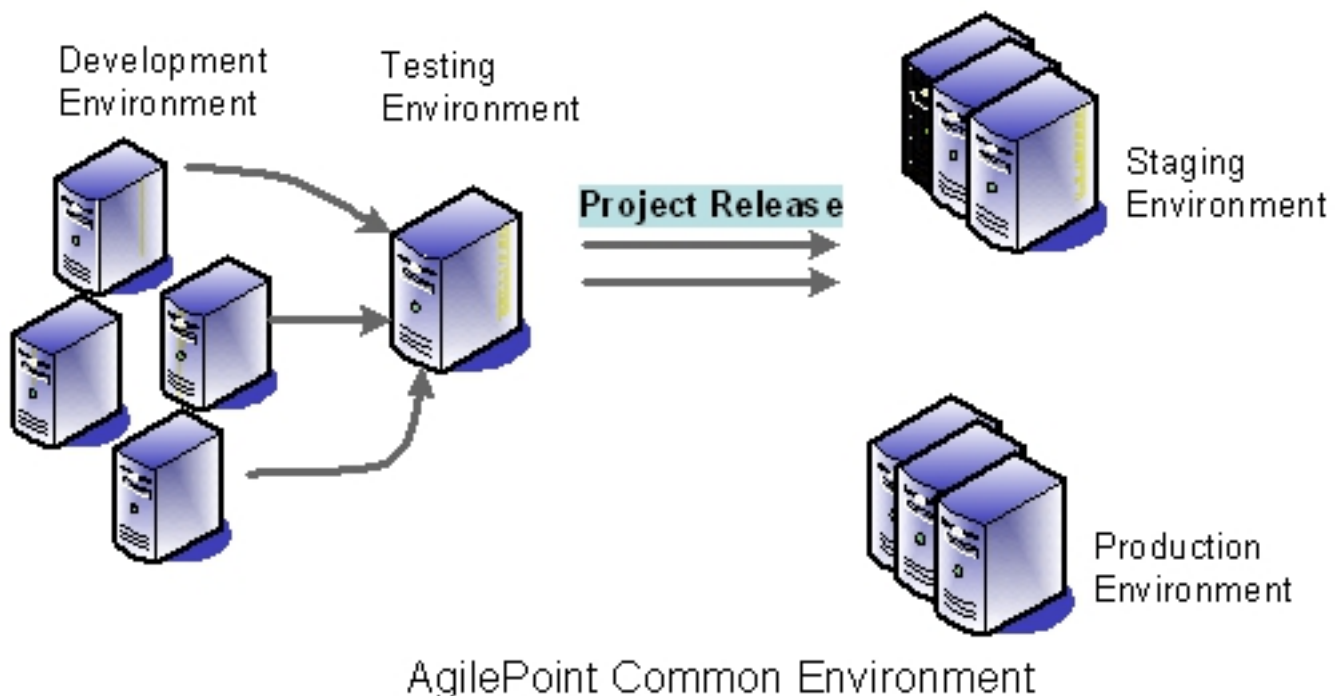
- Expected system performance and its testing environment and measurement.
- Data growth capacity, archive and reporting of historical data.
- Security policy and authorization, especially for Internet access.
- Daily maintenance and support after deployment.
- Version control of AgilePoint components and process models.
- Windows workstation version, browser version, and .Net framework version.
- Network bandwidth restrictions.

Development, Testing, Staging and Production Environment

Typically, four environments are needed for continual AgilePoint development and maintenance:

- Development
- Testing
- Staging
- Production

However, depending upon your project and deployment scenario, you may have more or fewer environments. For example, a process model minor change may require approval, but not a formal development process. You may add more sophisticated test environments for testing performance and security.



Virtual Environments

In order to save money, it is possible to run one or more of your non-production environments as virtual environments. This is much cheaper, but it tends to run slower than a physical environment. It is possible to run functional tests in a virtual environment, but not load tests.

If load testing is not required in a particular environment, AgilePoint recommends using a virtual environment. This is an especially effective strategy for development environments.

Note that even if you use a virtual environment, physical database servers are recommended.

Development in Virtual Environments

In particular, virtualization is advantageous for development environments because it enables isolation (sandboxing) between development environments to prevent them from impacting one another. Using virtualization technology also speeds up the process of environment configuration. It is a best practice to keep copies of each version of development environment for future enhancement.

Support for Virtual Environments

AgilePoint is committed to fully supporting AgilePoint BPMS running on virtualization technologies.

AgilePoint recommends installing AgilePoint on a physical server machine, but AgilePoint can be supported in a virtual machine (including NLB) given the virtual machine can support Windows Server 2003 or 2008, and the .NET Framework 3.5. AgilePoint recommends Windows Server® 2008 Hyper-V™, but other Microsoft and non-Microsoft virtualization products are also supported as discussed here. In addition to Windows Server® 2008 Hyper-V™, AgilePoint can also be deployed using Microsoft Virtual Server and Microsoft Virtual PC virtualization technologies. Only virtualization products that have specifically passed Microsoft's requirements for virtualization support are also officially supported for running AgilePoint. See the following Microsoft Web site that provides a list of non-Microsoft virtualization products that have passed the requirements for Windows Server 2003 and 2008:

<http://www.windowsservercatalog.com/results.aspx?&bCatID=1521&cplID=0&avc=0&ava=0&avq=0&OR=1&PGS=25&ready=0>

AgilePoint recommends the following configurations to ensure optimal performance of the AgilePoint BPMS in a virtual environment:

Host OS Architecture	X64
Host Machine Processors	4
Memory	4 GB allocated to each virtual machine
Bandwidth	1 Gb Network connection to the host machine
Virtual Memory	Memory allocation cannot always be guaranteed (i.e. if 4 GB is set for the virtual machine, it does not mean that the virtual machine will be using the entire 4 GB of memory at all times). The memory is shared amongst all the virtual machines.

AgilePoint Database	The AgilePoint Database should be on a physical machine.
---------------------	--

Testing Environment

The testing environment is logically a clean copy of development environment that allows developers and testers to do integration testing. The test environment should be isolated from development with a blank environment configuration to ensure most of the nonfunctional requirements would be tested so that the solution would not have major issues to deploy into staging and production environment.

The test environment may be virtual, unless load testing is required.

Staging Environment

This staging environment is for some of the nonfunctional requirements that may not be covered by development and testing environments, such as security, load-balancing, redundancy, and scalability. Usually, staging environment configuration is similar to the production environment. Planning a staging environment as part of production system would reduce the risk of introducing problems on the production environment.

Disaster Recovery

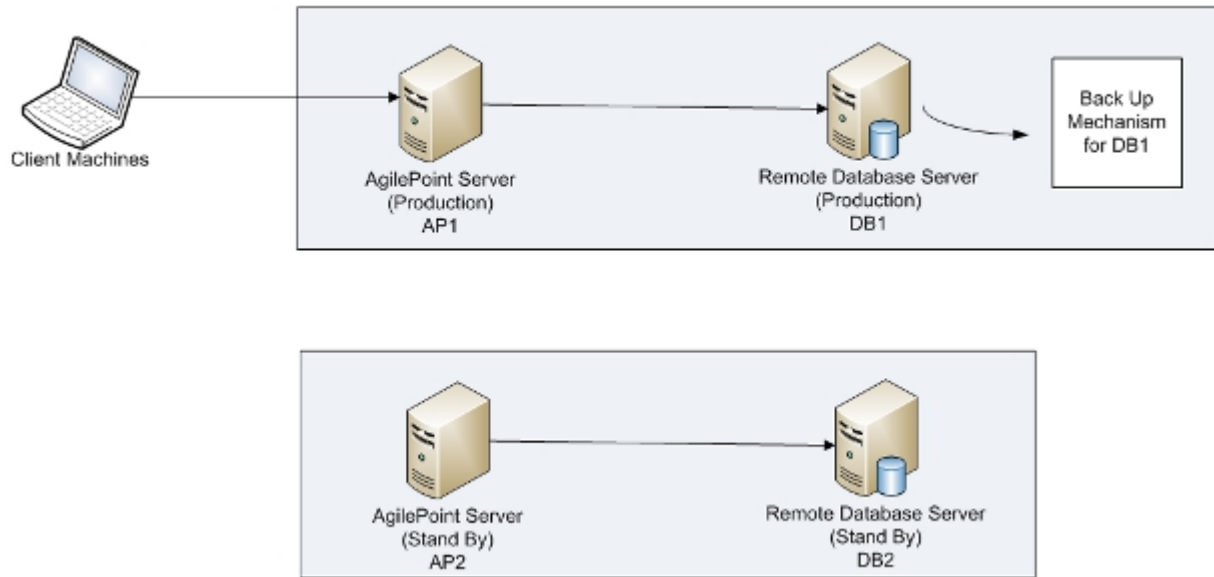
The information provides a brief overview of best practices for AgilePoint disaster recovery for environments that use a single AgilePoint Server instance, or multiple instances.

In either case, it is important to create a backup of the following components:

- The AgilePoint database
- web.config files
- netflow.cfg
- The archived AgilePoint database, if archiving is enabled
- The data services databases for data population and tracking, if data population and tracking are in use

Single AgilePoint Server Instance

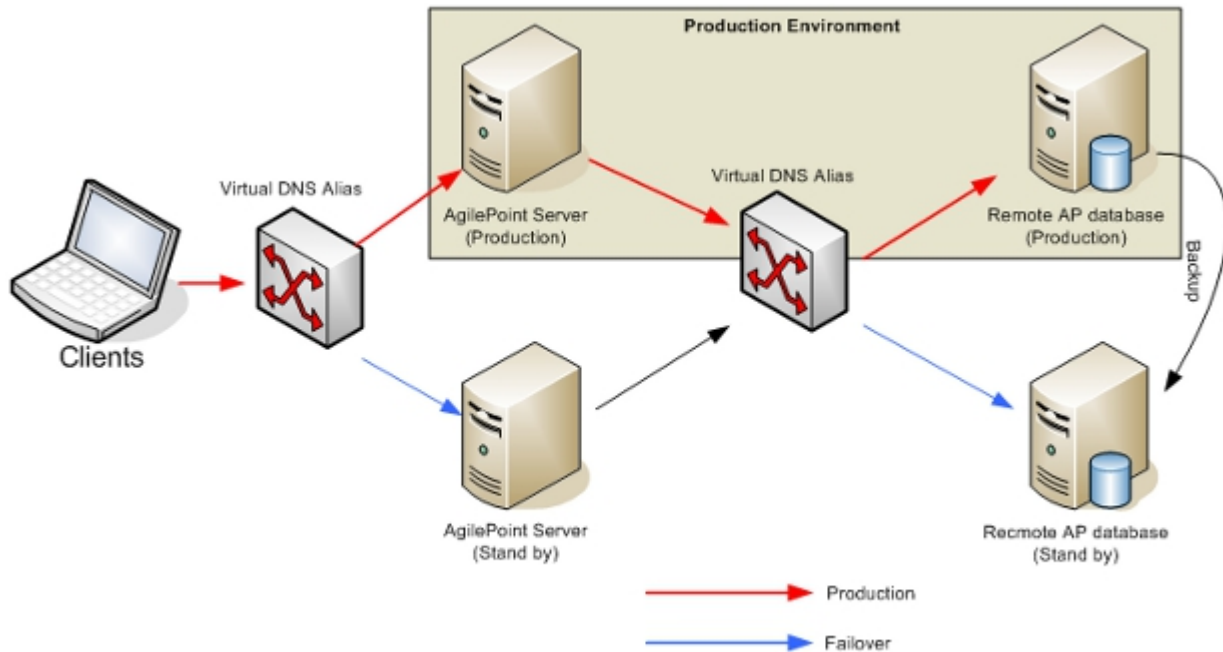
The following guidelines apply to environments where there is only one AgilePoint Server instance running at a time.



- The Standby AgilePoint Server (AP2) does not point to the Production Database (DB1). This would interrupt the live data in production.
- A Standby Database (DB2) is set up for the Standby AgilePoint Server (AP2). You can make use of your database backups to synchronize the data from your Production Database (DB1) to your Standby Database (DB2). The frequency to synchronize the data is based on your data growth and also your application tolerance. (For example, if you cannot afford to lose the data for 30 minutes, synchronize at a shorter interval.)
- The Standby AgilePoint Server (AP2) must have exactly the same software versions as the Production AgilePoint Server (AP1) for all the software installed on those two servers. This also means, whenever you apply upgrades or hotfixes to the Production AgilePoint Server (AP1), ensure those upgrades or hotfixes are applied to the Standby AgilePoint Server (AP2) as well. We strongly recommend you to keep documentation that records each software upgrade/hot fixes for these two AgilePoint Servers (AP1 and AP2).
- When a failure occurs and the production environment is not available, the AgilePoint Server URL (the one used by your clients to access AgilePoint) mapping is switched to point to the Standby environment. All the events for workflow are recorded in the database. So, if you switch the front-end web server from the production AgilePoint Server to the Standby AgilePoint Server, the Standby AgilePoint Server can pick up the events in the database and continue to process those events.

Multiple AgilePoint Server Instances

The following guidelines apply if you have more than one AgilePoint Server instance running at one time. In this situation, the standby systems for disaster recovery are handled similarly to standby systems for failover.



- The virtual DNS alias points to the production AgilePoint Server. (This production AgilePoint Server receives and processes the requests from the clients, and it also runs a thread to read the events from the database, processes them, and write them back to database.)
- The Standby AgilePoint Server instance runs continuously. This Standby AgilePoint Server does not receive and process the requests from the clients, but it does read and write to the database for event processing.

Performance

Powered by Microsoft .NET technology, AgilePoint leverages standard Microsoft and web-based technologies and the C#, ASP.NET language APIs. AgilePoint Server is fully developed in C# code that is a key factor to deliver high performance. Because of its web service architecture, AgilePoint clients can be either browser-based or Windows client-based while delivering roughly equivalent performance.

AgilePoint is designed with performance-enhanced features, such as process swapping, database indexing, and data caching. AgilePoint has gone through workload simulations with thousands of concurrent users with good results. AgilePoint can be further optimized with load balancing, application data indexing, and additional resources (CPU, etc.).

Security

Security is an important component of any enterprise application. AgilePoint leverages Microsoft security model and provides a secure framework for BPMS development.

Authentication

Authentication is defined as secure identification which is a mechanism for securely identifying user's identity that matches the security requirements of the application. Authentication must be implemented in different layers to provide higher level of security.

User Interface Authentication

Authentication needs to be implemented in the user interface layer to provide authorization, auditing, and personalization capabilities. Authentication typically involves user credentials such as username and password. A wide range of authentication mechanisms are available for web-based user interfaces. For example, AgilePoint Enterprise Manager prompts users to enter credentials (username and password) for logon. However, AgilePoint supports other mechanisms such as Windows authentication which based on the Windows logon credentials.

Server Process Authentication

Besides the user interface authentication, the server logic layer must also authenticate the identity that calls the service or runs the process. For AgilePoint Server, the credentials are based on the setting of the IIS identity. The running of all processes will be using this identity. For application integration, AgilePoint supports the notion of impersonation. Each integrated application will require an impersonator as part of the application settings for authentication. You can provide different impersonators for different integrated applications to separate the authentication credentials to provide higher level of security and control.

Database Authentication

In order to further separate the authentication identity between the application and the data, it is recommended to use a separate authentication mechanism or credential for your database access. AgilePoint supports both SQL Authentication (which requires username and password explicitly) and Windows authentication (which based on the Windows logon credentials). For example, AgilePoint Server Configuration supports both authentications through which users can determine what mechanism would best fit their needs base on their company's policy, system and application requirements, etc.

Authorization

Once an identity is authenticated, it must be authorized. Authorization is the process of verifying permissions and rights for an authenticated user.

User Access Security

AgilePoint uses a role-based authorization mechanism to handle user access security. In AgilePoint, a role is defined as a collection of privileges or access rights.

- Individuals or groups of individuals can be added to a role as the member of the role.
- Each member of a role possesses all the privileges granted to the role. If a person is a member of multiple roles, he/she will possess the sum of all the access rights associate to the different roles.
- Once users are authenticated to AgilePoint, AgilePoint Server checks their rights based on their roles to determine if they have permission specific tasks.
- AgilePoint provides built-in roles such as Administrator or Process Template Designer. Additional administrator-defined roles can be created to meet the needs of the application.

Active Directory Integration

AgilePoint supports Active Directory integration where Windows authentication can occur directly against the Active Directory. AgilePoint also provides integration with Active Directory where organization and user credentials information can be directly retrieved or synchronized with Active Directory. AgilePoint Envision also provides direct access to Active Directory and can leverage Active Directory's user group setting directly as part of the business rules. This Active Directory integration provides you flexibility to tie AgilePoint usage as part of your overall security policy through the use of Active Directory.