Contents

Legal Statements and Policies

- Disclaimer of Warranty ................................................................. 3
- Copyright and Trademarks................................................................. 3
- Government Rights Legend .............................................................. 3
- Virus-Free Software Policy ............................................................. 3

Architecture Guide ........................................................................ 4

Implementation Architecture ............................................................ 5
- AgilePoint NX OnDemand ................................................................. 5
- AgilePoint NX PrivateCloud ............................................................. 6
- AgilePoint NX OnPremises ............................................................... 7

Platform Architecture .................................................................... 8

Software Architecture ..................................................................... 10

Deployment Architecture ................................................................. 15
- Deployment/Installation Architecture Model .................................... 15
  - Single Server Deployment Architecture ...................................... 16
  - NLB Deployment Architecture ....................................................... 17
  - Regional Deployment Architecture .............................................. 17
  - Multi-tenant Architecture ........................................................... 21
- Strategy for Development, Testing and Staging ............................... 21
  - Development, Testing, Staging and Production Environment ........... 21
- Disaster Recovery ................................................................. 23
  - Single AgilePoint Server Instance ............................................... 23
  - Multiple AgilePoint Server Instances ......................................... 24

Security ...................................................................................... 25
- Authentication .............................................................................. 25
  - User Interface Authentication ...................................................... 25
  - Server Process Authentication .................................................... 26
  - Authentication for Third-Party Applications ................................... 26
  - Database Authentication ............................................................ 26
- Authorization .............................................................................. 26
- Active Directory Integration ......................................................... 27
This section provides legal statements and general policies for AgilePoint software and documentation.

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 and Trademarks

© 2019 AgilePoint, Inc. All rights reserved.

AgilePoint is a registered trademark of AgilePoint, Inc. AgilePoint's products, including NX, 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. AgilePoint takes the following measures to make sure 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 in is fully isolated and protected, and anti-virus software is installed and running during packaging.

- The compiled and packaged software files are scanned by a variety of anti-virus software services before they are made available for customer download. An official disclosure document regarding the findings from our virus scanning activities is available upon request.
Architecture Guide

This document describes the hardware, software, and deployment architectures for AgilePoint NX.
Implementation Architecture

This section describes the overall architecture for the 3 main AgilePoint NX implementation types.

AgilePoint NX OnDemand

AgilePoint OnDemand is a version of AgilePoint NX that runs entirely in the cloud with no software installed in your local environment.

The AgilePoint NX OnDemand platform as a service (PaaS) can connect with other cloud-based services and platforms, such as Salesforce or Microsoft Office 365. It can also connect to services inside your company's firewall, such as SAP or on-premises SharePoint.

Figure 1: AgilePoint NX OnDemand Implementation Architecture

These features that are available in AgilePoint NX OnPremises and AgilePoint NX PrivateCloud are not available in AgilePoint NX OnDemand:

- AgilePoint NX Developer.
- Data Services.
- Event Services.
- Active Directory integration (ADSyncModule Extension).
- SAP integration (SAP Integration AgileConnector).
- Skype for Business integration.
- SMS notifications.
- Synchronize with a database user store (DBSyncModule Extension).
- Synchronize SharePoint users (SPSyncModule Extension).
- These authentication providers:
  - ADFS
  - Okta
• Identity Server 4
• CA SiteMinder
• Administrator access to the AgilePoint Server configuration.
• VPN connection to systems that are installed behind your firewall.
• Local file system as a data source in an application.
• Create reports against a custom database in Report Center.
• These activities in the Process Builder:
  • External Command activity
  • PowerShell Command activity
  • SharePoint PowerShell Command activity
  • Begin SAP Transaction activity
  • Call SAP Function activity
  • End SAP Transaction activity
  • .NET Proxy activity
  • Create User (Active Directory) activity
• Loop a process more than 100 times.
• Work calendar.
• Upload files as e-mail attachments in e-mail templates or e-mail notifications.
  For more information, refer to Why Are E-mail Attachments Disabled in AgilePoint NX OnDemand?
• In AgilePoint NX OnDemand, bulk migration requires help from AgilePoint Customer Support. For more information, refer to How Do I Migrate More than One Process Instance to a New Process Model Version?
• AgilePoint NX App for Salesforce.
  For v7.0 and higher, this feature is available only for AgilePoint NX OnPremises or AgilePoint NX PrivateCloud.
  • Salesforce as a primary data source for a process-based app.
    For v7.0 and higher, this feature is available only for AgilePoint NX OnPremises or AgilePoint NX PrivateCloud.

AgilePoint NX PrivateCloud

AgilePoint NX PrivateCloud is a version of AgilePoint NX where the platform is hosted with a cloud subscription in a dedicated virtual machine. This subscription can be managed by the AgilePoint Infrastructure team (referred to as AgilePoint-hosted PrivateCloud), or you can use your own subscription (referred to as customer-hosted PrivateCloud).

Like AgilePoint NX OnDemand, you have a self-contained environment. The AgilePoint NX OnDemand platform as a service (PaaS) can connect with other cloud-based services and platforms, such as Salesforce or Microsoft Office 365. It can also connect to services inside your company’s firewall, such as SAP or on-premises SharePoint. Unlike AgilePoint NX OnDemand, you can connect to the services inside your firewall using a virtual private network (VPN).
AgilePoint NX OnPremises

With AgilePoint NX OnPremises, the server-side components of the AgilePoint NX platform are hosted on servers inside your organization's firewall. Like AgilePoint NX PrivateCloud or AgilePoint NX OnDemand, AgilePoint NX OnPremises can operate in a multi-tenant configuration. AgilePoint NX OnPremises operates as a platform as a service (PaaS). You can connect to services inside or outside your firewall, and you maintain complete control of the connections and security.
Platform Architecture

This diagram shows the AgilePoint NX platform architecture.

NX Portal
The NX Portal is a software component that shows other AgilePoint NX applications, such as Process Builder and Manage Center. The Portal enables authentication to AgilePoint NX, and provides the overall navigation.

AgilePoint Developer
AgilePoint NX Developer is a AgilePoint NX software component that lets you create custom assets, such as activities, AgileConnectors and AgilePoint NX web applications. AgilePoint NX Developer is an extension for Microsoft Visual Studio.

Windows Service (WCF)
WCF is a Microsoft framework for building service-oriented applications.

RESTful Services
AgilePoint NX offers a full set of REST APIs, and can integrate with RESTful services directly from an application with out-of-the-box activities.
**AgilePoint Server**

*AgilePoint Server* is a software engine that runs *AgilePoint NX applications* behind the scenes.

**Database**

*AgilePoint NX* uses several *databases* to store data.

For more information, refer to *System Requirements for AgilePoint Server*.

**Abstraction Extension Framework**

The Abstraction Extension Framework lets you extend the *AgilePoint NX* in many different ways:

- Use out-of-the-box integrations for more than 30 different third-party services.
- Create your own *AgileConnectors* to connect to even more third-party services.
- Create your own *activities*.
- Create a custom UI for your *applications*.
- Create a custom UI for *NX Portal*, or any of its component applications.
Software Architecture

This diagram shows the AgilePoint NX software architecture.

**eForm Builder**

The eForm Builder is an integrated visual tool that lets you build anything from simple forms with just a few form controls to complex, multi-tabbed forms. The eForm Builder provides more than 50 control types, and lets you build business logic into the form, enabling dynamic restructuring of the form based on form user input. You can use your eForms within an application, share your eForms across applications, or even use eForms with external applications.

**Process Builder**

The Process Builder is a visual tool where you can create workflows and the logic for processes and applications with a simple drag-and-drop interface.

**Application Test and Deployment**

You can export or import your application as a set of JSON files, or publish your application. For more information, refer to:

- Export an Application
- Import an Application
- Publish an Application
**Team Work Management**

The **Work Center** is a component where you can access to the information about the work you care about, and the **applications** that are available for your use. This component also lets supervisors manage the work of their subordinates.

**Application Versioning**

This component lets you **check in, check out, or publish** an **applications** in the AgilePoint system and manage the versions.

**Data Modeling**

This component lets you configure the different **data sources** and **data source types** for your **application**. The data source for an **eForm** is created dynamically as you add **form controls**. In addition, you can add external data sources to supported third-party systems.

**Report Center**

The **Report Center** can create, run, and configure **reports** and reports settings related to data in your **applications**.

**User, Group, and Role Authorization**

AgilePoint NX uses an **authorization** system based on **access rights** that can be applied to a **user**, **group**, or **role**.

**Process Model Management**

This component lets you manage the **process models** in your environment.

**Tenant Provisioning**

This component lets you **provision** a **tenant** in a multi-tenant AgilePoint NX environment.

**Process and Activity Monitoring**

You can see information about your **process instances**, such as template name, **status**, start date, due date, and completion date and about the **system activities** in your **process**, such as the start date, completed date, and status.

**Application Configuration**

Lets you manage **application permissions**, create **access tokens**, and configure the settings for the **App Builder**.

**Dashboard (System Management)**

Shows a summary of your system data and business data..

For more information, refer to **System Overview**.

**Form Engine**

The runtime engine for the **eForm Builder**. It uses HTML5 and jQuery to render the **eForm**.
Process Engine
The main component of AgilePoint Server that lets you run, pause, stop, and change processes live without writing software code.

Multi-Tenant Management
You can configure AgilePoint NX to run in a multi-tenant environment.
For more information, refer to Multi-Tenant Configuration

License Management
An internal service manages which components of AgilePoint NX to which you have access, based on your license key.

Exception Handler
You can create a custom AgileConnector to handle exceptions and take specific actions when an exception occurs. Each activity also has configuration properties that let you specify what happens when an exception occurs. You can specify the action to take when an exception occurs, the amount of detailed process-related data that is recorded to the AgilePoint workflow database, and a process data variable to store information about the error.

Process Swapping Management
Lets you configure the amount of time an system activity is loaded in memory. This is to improve the efficiency of the memory usage.
For more information, refer to Configuring the Swap Out Time

Process Instance Management
Lets you start, pause, cancel or move a process instance in a process.

Caching and Clustering, Process Load Balance Management
The AgilePoint clustering functionality within the AgilePoint Server Windows service works as an event dispatcher using .NET Remoting. This service records information from the AgilePoint Servers track which process instances are running on each of the AgilePoint Server machines. When a process instance is created, the first event is load balanced by the NLB. After that, the clustering feature dispatches events to each of the AgilePoint Servers in the cluster, ensuring that all events associated with the same process instance lifecycle are processed on the same AgilePoint Server machine. When a new database event is triggered, the AgilePoint Server machine picks up the event, it uses the clustering feature to determine whether the event is currently being handled by any of the AgilePoint Servers in the NLB based on the associated process instance ID. If the event is associated with a Process Instance ID that is already running on a different server, the AgilePoint Server clustering component will re-route the event to be processed on the AgilePoint Server that is currently dedicated to processing events associated with that process instance.

Security Service (Authentication)
The service that manages the authentication in AgilePoint Server.
For more information, refer to Authentication.
Logging Service
The logging service lets you see errors and events that are recorded in log files in various locations. Logs show informational, warning, and error messages about the AgilePoint Server, the process engine, execution of activities, and errors that occur in network connection.
For more information, refer to I Need Help with AgilePoint Logs.

Event Service
The event service is a set of options that let you an process-based app when a specified event occurs.

Integration Service
You connect AgilePoint NX applications to over 30 different third-party services out of the box. You can also add your own integrations with custom AgileConnector.

Timer Service
You can set the time for different actions in your process. For example, you can pause for a specified time before process continues, specify a time limit for the session to complete an activity, or set a limit on your expiration to your hours of operation.

Process Archiving Service
You can archive a process instance and all its related data to an archive database when the completed or canceled date for the process instance exceeds a specified number of days.

Data Population Service
The data population service lets you store the values for specified process data variables in a database. You can use this data for reports.

Data Tracking Service
The data tracking service monitors process data variables based on specified conditions. You can see the audit history report of the tracked data in the Manage Center on the Data Tracking tab if the value of the tracked variable meets the condition.

Notification Service
This services sends a notification when an event occurs in a process. Notifications can send through applications such as e-mail, Skype for Business, Yammer, or Salesforce Chatter.
For more information, refer to Notification.

Multi-Tenant Architecture
The AgilePoint NX multi-tenant architecture is a platform that can run more than one tenant from one AgilePoint Server instance. In a multi-tenant environment, each tenant has its own, self-enclosed database.

Embedded Hosted Application
You can use AgilePoint NX components like eForm Builder and Manage Center in your own portal. It is not necessary to use the NX Portal application.
**Windows Service (WCF)**

WCF is a Microsoft framework for building service-oriented applications.

**SQL Server**

AgilePoint Server stores data in Microsoft SQL Server.
For more information, refer to System Requirements for AgilePoint Server.

**Microsoft .NET Framework**

AgilePoint NX is built on the Microsoft .NET Framework.
For more information, refer to System Requirements for AgilePoint Server.

**IIS**

Some web-based components of AgilePoint NX, such as NX Portal, require IIS.
For more information, refer to System Requirements for AgilePoint Server.

**Messaging Service**

The Messaging Service sends messages from applications with SMS.

**Microsoft Operating System**

AgilePoint Server runs on a Microsoft Operating System.
For more information, refer to System Requirements for AgilePoint Server.
Deployment Architecture

There are three basic AgilePoint deployment architecture models. Each model can be tuned to scale up and out for overall of 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 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 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.
Single Server Deployment Architecture

Single-server AgilePoint NX deployment is mainly used for entry-level, department-specific, and small business deployments.

Main Points

- The external front end server is not necessary if your application uses eForms as the user interface.
- If the AgilePoint NX application load is small, the database can be shared with other applications.
- AgilePoint Server is database-intensive. Therefore, a faster database server increases AgilePoint Server performance.
NLB Deployment Architecture

An network load balancing (NLB) deployment uses a 3-tier architecture. An NLB configuration 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 web forms, SharePoint libraries, and reports.

Main Points

• The external front end server is not necessary if your application uses eForms as the user interface.

• The architecture can be scaled out by adding nodes to existing NLB clusters.

• The architecture can be scaled up by upgrading existing nodes (for example, increasing CPU and memory) and adding more instances of AgilePoint Server and web applications on each server.

• AgilePoint Server is database-intensive. Therefore, a faster database server increases AgilePoint Server performance.

Regional Deployment Architecture

Regional deployment is used for organizations that keep their AgilePoint instances isolated from one regional team to the next — for example, where an AgilePoint instance in one country is isolated from the AgilePoint instance in another country.

As process load increases, a regional deployment model is necessary for the enterprise or any organization. It provides not only layer isolation, but also 3-tier high availability and performance by clustered database, AgilePoint and front-tier applications.
**Main Points**

- The external front end server is not necessary if your application uses eForms as the user interface.
- The architecture can be scaled out by adding nodes to existing NLB clusters.
- The architecture can be scaled up by upgrading existing nodes (for example, increasing CPU and memory) and adding more instances of AgilePoint Server and web applications on each server.
- AgilePoint Server is database-intensive. Therefore, a faster database server increases AgilePoint Server performance.
- Users from other regional teams do not have access to your region's processes since each instance gets its own database instance. This gives you strict control over your processes.
- Upgrades can be planned independently without having to rely on other teams.
- Your AgilePoint Server machine can be scaled independently of other regional team servers.
- Better performance can be achieved since the physical server is located at the same location so network bandwidth does not become a constraint.
- Because each team has its own AgilePoint Server, business time calculation and holiday schedule can be effectively controlled for each region. As such, you will not have a task due on a holiday.
- Physical hardware cost increases because each team must buy its own hardware.
- License costs increase because each server license is charged separately.
- Each server must be upgraded separately, instead of centralized upgrades.
Centrally Hosted Regional Deployment, More than One Instance of AgilePoint Server

This diagram shows an example of a regional AgilePoint NX deployment with a more than one instance of AgilePoint Server.

Main Points

• Users from other regional teams do not have access to your region’s processes since each instance gets its own database instance. This gives you strict control over your processes.

• Because each team has its own AgilePoint Server, business time calculation and holiday schedule can be effectively controlled for each region. As such, you will not have a task due on a holiday.

• Physical hardware cost can be controlled since all teams can share same physical hardware but at same time maintain their own database instance.

• In case you prefer to have centralized upgrades for all teams then this is a good option since you can run upgrade once on the server and all instances get upgraded.

• Upgrades cannot be planned independently so all teams need to plan for upgrades together.

• Offshore teams can notice some level of performance decrease if the network connection to remote server is not good.
Centrally Hosted Regional Deployment, Shared AgilePoint Server Instance

This diagram shows an example of a regional AgilePoint NX deployment with a shared instance of AgilePoint Server.

Main Points

- Centralized server can use same AgilePoint license for all teams.
- Physical hardware cost can be controlled because all teams can share same physical.
- If you prefer to have centralized upgrades for all teams then this is a good option since you can run upgrade once on the server and all teams get upgraded.
- Upgrades cannot be planned independently so all teams need to plan for upgrades together.
- Offshore teams can notice some level of performance degrade if the network connection to remote server is not good.
- The database instance is shared between different regional teams therefore power users from other team who have higher privileges can access information from processes belonging to other teams.
- Teams would be sharing the same work calendar which would mean that business time calculations cannot be used for other teams since that would result in task becoming Overdue during their non working hrs. Also holiday schedule cannot be used either due to shared calendar.
• Offshore teams can notice some level of performance decrease if the network connection to remote server is not good.

Multi-tenant Architecture

AgilePoint NX can be installed in a multi-tenant architecture where each tenant has its own isolated database.

Main Points

• Users from other regional teams do not have access to your region’s processes since each instance gets its own database instance which lets you to have strict control over your processes.
• Physical hardware cost can be controlled since all teams can share same physical hardware but at same time maintain their own database instance.
• In case you prefer to have centralized upgrades for all teams then this is a good option since you can run upgrade once on the server and all instances get upgraded.
• Upgrades cannot be planned independently so all teams need to plan for upgrades together.
• Offshore teams can notice some level of performance degrade if the network connection to remote server is not good.

Strategy for Development, Testing and Staging

Planning, implementing and deploying an AgilePoint 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.

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.

**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 make sure 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), make sure 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.
Security

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

Authentication

AgilePoint NX implements authentication in multiple 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 a username and password. AgilePoint NX prompts users to enter credentials when they sign into the application. Sign-in credentials can come from any of these sources:

<table>
<thead>
<tr>
<th>Authentication Type</th>
<th>Supported In</th>
</tr>
</thead>
<tbody>
<tr>
<td>AgilePoint ID</td>
<td>• AgilePoint NX OnDemand (public cloud)</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX PrivateCloud</td>
</tr>
<tr>
<td>Salesforce</td>
<td>• AgilePoint NX OnDemand (public cloud)</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX OnPremises</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX PrivateCloud</td>
</tr>
<tr>
<td>Windows Azure Active Directory</td>
<td>• AgilePoint NX OnDemand (public cloud)</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX OnPremises</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX PrivateCloud</td>
</tr>
<tr>
<td>Office 365 SharePoint Online</td>
<td>• AgilePoint NX OnDemand (public cloud)</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX OnPremises</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX PrivateCloud</td>
</tr>
<tr>
<td>Google</td>
<td>• AgilePoint NX OnDemand (public cloud)</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX OnPremises</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX PrivateCloud</td>
</tr>
<tr>
<td>ADFS</td>
<td>• AgilePoint NX OnPremises</td>
</tr>
<tr>
<td></td>
<td>• AgilePoint NX PrivateCloud</td>
</tr>
</tbody>
</table>
Authentication Type | Supported In
---|---
Okta | • AgilePoint NX PrivateCloud or AgilePoint NX OnPremises v7.0 Software Update 1 or higher
Identity Server 4 | • AgilePoint NX PrivateCloud or AgilePoint NX OnPremises v7.0 Software Update 1 or higher
CA SiteMinder | • AgilePoint NX PrivateCloud or AgilePoint NX OnPremises v7.0 Software Update 1 or higher

To use a custom database authentication, contact AgilePoint Professional Services.
For more information, refer to Authentication.

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. AgilePoint Server and its associated process use the AgilePoint System Account identity.

Integrated applications use the concept of an encrypted access token that passes the authentication credentials for the application.

Authentication for Third-Party Applications

Integrated third-party applications use the concept of an encrypted access token that passes the authentication credentials for the application.

Database Authentication

In order to further separate the authentication between the applications and data, it is recommended to use a separate authentication mechanism or credentials 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 based on their company's policy, system, and application requirements, etc.

Authorization

Once an identity is authenticated, it must be authorized.
AgilePoint NX uses an authorization system based on access rights that can be applied to a user, group, or role.
- Individual users or groups can be members of a role.
- Each member of a role possesses all the privileges granted to the role. If a person is a member of multiple roles, he or she will possess the sum of all the access rights associated with 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 NX supports application permissions. These permissions provide built-in roles, such as system administrator or application designer, that give authorization at the application level. 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 App Builder 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.