[MS-GRVPROT]: Groove Protocols Overview

Abstract

This document describes the intended functionality of the Microsoft® Groove® and Microsoft SharePoint® Workspace system and how the protocols within this system interact. It also provides examples of some common user scenarios. It does not restate the processing rules and other details that are specific to each protocol. Those details are described in the protocol specifications for each of the protocols and data structures that make up this system.

The Groove and SharePoint Workspace system is designed for Internet-based collaboration. The system consists of protocol clients that communicate with each other and with supporting protocol servers. The system protocols, as described in [MS-GRVRDB], [MS-GRVDYNM], [MS-GRVWDPP], [MS-GRVSSTPS], [MS-GRVSSTP], [MS-GRVHENC], [MS-GRVSPCM], and [MS-GRVSPMR], are designed to facilitate and help secure communications, data synchronization, and supporting services and management.

Revision Summary

Date	Revision History	Revision Class	Comments
04/04/2008	0.01	Major	Initial Availability
06/27/2008	1.0	Minor	Revised and edited technical content
12/12/2008	1.01	Editorial	Revised and edited technical content
07/13/2009	1.02	Major	Revised and edited the technical content
08/28/2009	1.03	Editorial	Revised and edited the technical content
11/06/2009	1.04	Editorial	Revised and edited the technical content
02/19/2010	2.0	Minor	Updated the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Editorial	Changed language and formatting in the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.05	Minor	Clarified the meaning of the technical content.
11/15/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	2.05	No change	No changes to the meaning, language, or formatting of the technical content.
06/10/2011	2.05	No change	No changes to the meaning, language, or formatting of

Date	Revision History	Revision Class	Comments
			the technical content.
01/20/2012	2.05	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

The Microsoft® Groove® and Microsoft SharePoint® Workspace system is designed to facilitate Internet-based collaboration. The system consists of protocol clients and supporting management, relay, and data bridge servers. The following diagram provides a high-level overview of the system.

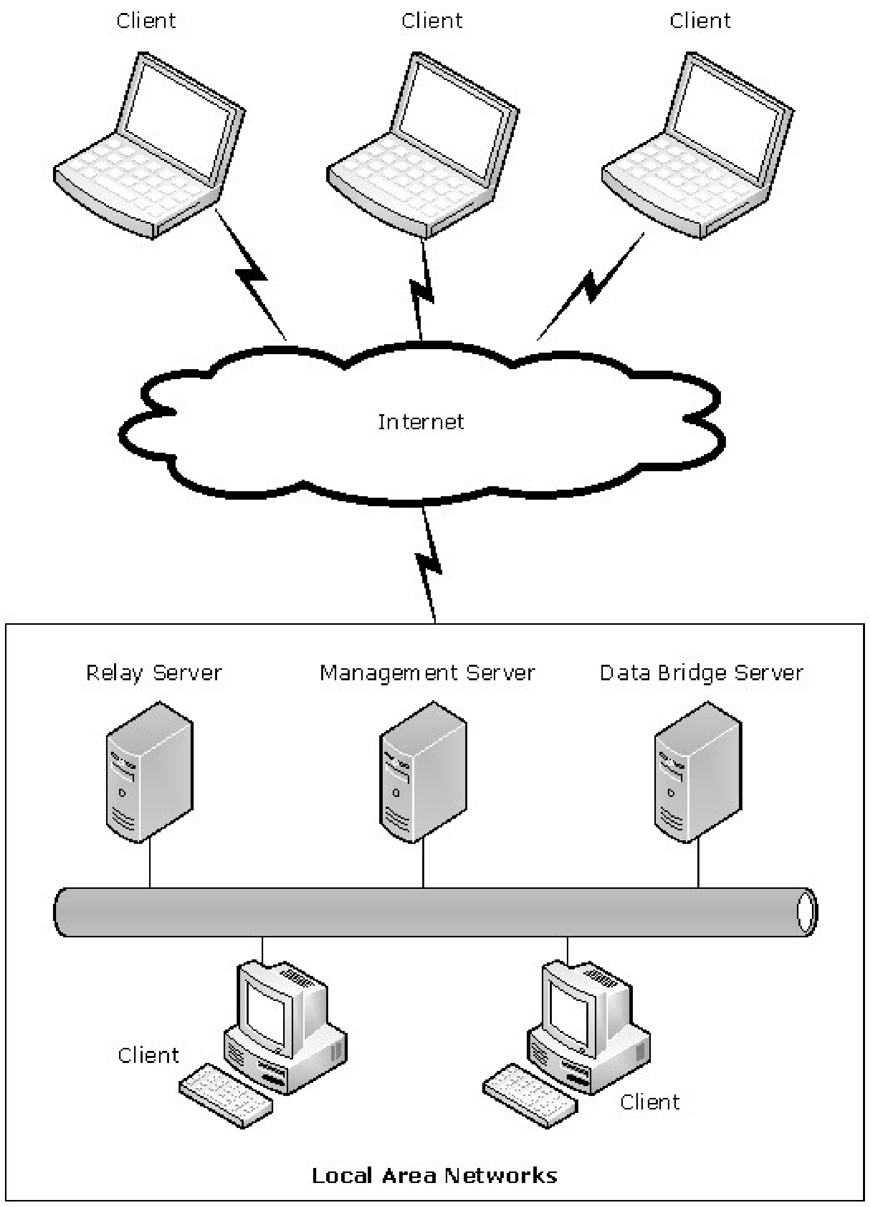

Figure 1: Overview of the Groove and SharePoint Workspace system

Protocol clients within the system can connect to each other through local area networks (LANs) or the Internet. The **management server** provides services for managing users of the system. The **relay server** provides services for protocol clients when direct peer-to-peer communications are not possible. The **data bridge server** enables integration with external applications.

Users of the system can create **shared spaces** and share those spaces with other users. A shared space can contain documents and tools, such as calendar, discussion, and meeting tools, and it is synchronized automatically between protocol clients. Members of a collaboration team can be part of the same organization or multiple, independent organizations. In addition, users can work either online or offline.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Domain Name System (DNS)
Hypertext Transfer Protocol (HTTP)
Secure Sockets Layer (SSL)

The following terms are defined in [MS-OFCGLOS]:

account configuration code
data bridge server
fanout
identity
management server
peer
presence
relay server
shared space
Simple Object Access Protocol (SOAP)
Simple Symmetric Transport Protocol (SSTP)
TCP/IP
Web Services Description Language (WSDL)

1.2 References

References to Microsoft Open Specification documents do not include a publishing year because links are to the latest version of the documents, which are updated frequently. References to other documents include a publishing year when one is available.

We conduct frequent surveys of the informative references to assure their continued availability. If you have any issue with finding an informative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[MSDN-GWSDF] Microsoft Corporation, "Microsoft Office Groove 2007 Web Services Developer Reference", http://msdn.microsoft.com/en-us/library/bb403118.aspx

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary".

[MS-GRVDYNM] Microsoft Corporation, "Groove Dynamics Protocol Specification".

[MS-GRVHENC] Microsoft Corporation, "HTTP Encapsulation of Simple Symmetric Transport Protocol (SSTP) Protocol Specification".

[MS-GRVRDB] Microsoft Corporation, "Groove RDB Commands Protocol Specification".

[MS-GRVSPCM] Microsoft Corporation, "Client to Management Server Groove SOAP Protocol Specification".

[MS-GRVSPMR] Microsoft Corporation, "Management Server to Relay Server Groove SOAP Protocol Specification".

[MS-GRVSSTP] Microsoft Corporation, "Simple Symmetric Transport Protocol (SSTP) Specification".

[MS-GRVSSTPS] Microsoft Corporation, "Simple Symmetric Transport Protocol (SSTP) Security Protocol Specification".

[MS-GRVWDPP] Microsoft Corporation, "Wide Area Network Device Presence Protocol (WAN DPP) Specification".

[MS-OCPROTO] Microsoft Corporation, "Office Client Protocols Overview".

[MS-OFCGLOS] Microsoft Corporation, "Microsoft Office Master Glossary".

2 Functional Architecture

The following sections describe the functional architecture of the Microsoft® Groove® and Microsoft SharePoint® Workspace system.

2.1 Overview

The Microsoft® Groove® and Microsoft SharePoint® Workspace system is designed to facilitate collaboration over the Internet, primarily through the use of shared spaces.

The protocols within the system are designed to facilitate and help secure communications between protocol clients and protocol servers, which includes data bridge servers, management servers, and relay servers. A data bridge server is a peer to protocol clients and it enables integration between external applications and the system. Such integration is possible through the use of Groove Web Services, which are described in [MSDN-GWSDF] and are based on the **Web Services Description Language (WSDL)**.

The following diagram provides a high-level overview of communications between protocol clients and protocol servers that are part of the system.

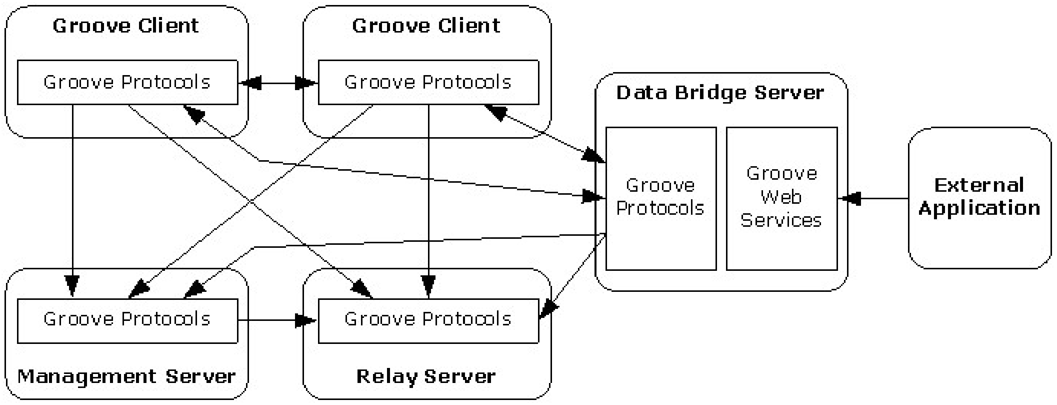

Figure 2: Overview of system communications

As shown in the preceding diagram, protocol clients and a data bridge server are **peers** that can communicate with each other. They act as clients of a management server and a relay server, and they use request/response-based communications to interact with those servers. In addition, a management server is a client of a relay server and it can be used to manage a relay server. External applications use the Groove Web Services protocol, as described in [MSDN-GWSDF], and a data bridge server to communicate with the system.

The protocols within the system are Internet-based and they use standard Internet protocols for transport. The following figure illustrates the relationship between the system and standard Internet protocols.

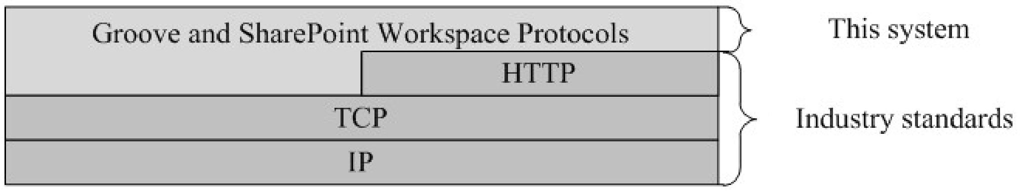

Figure 3: Relationship between the system and standard Internet protocols

System protocols consist of:

- Client to Management Server Groove SOAP Protocol, as described in [MS-GRVSPCM]
- Groove Dynamics Protocol, as described in [MS-GRVDYNM]
- Groove RDB Commands Protocol, as described in [MS-GRVRDB]
- HTTP Encapsulation of SSTP, as described in [MS-GRVHENC]
- Management Server to Relay Server Groove SOAP Protocol, as described in [MS-GRVSPMR]
- Simple Symmetric Transport Protocol (SSTP), as described in [MS-GRVSSTP]
- SSTP Security Protocol, as described in [MS-GRVSSTPS]
- Wide Area Network Device Presence Protocol (WAN DPP), as described in [MS-GRVWDPP]

The following figure illustrates the relationships between system protocols and standard Internet protocols.

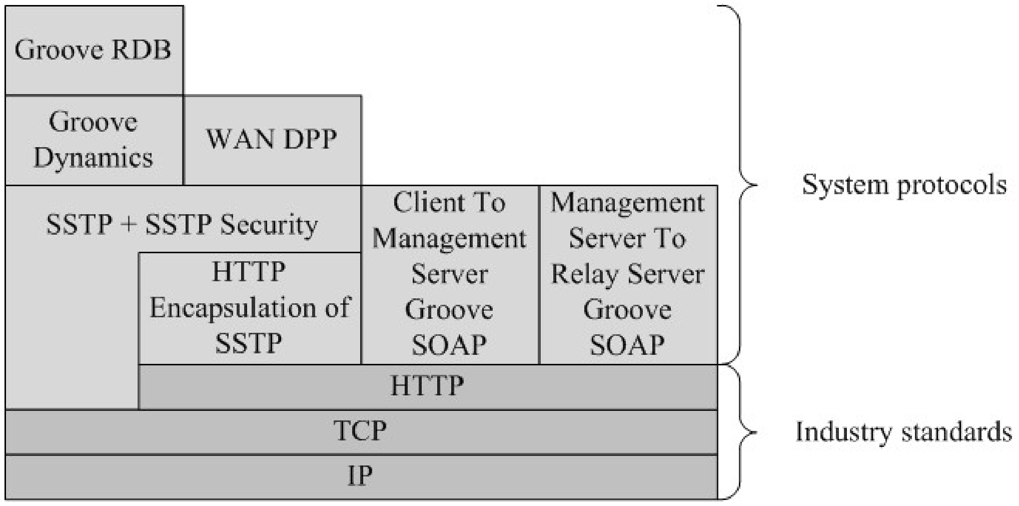
Groove RDB
Groove Dynamics
WAN DPP
SSTP + SSTP Security
HTTP Encapsulation of SSTP
Client To Management Server Groove SOAP
Management Server To Relay Server Groove SOAP
HTTP
TCP
IP
System protocols
Industry standards

Figure 4: Relationships between system protocols and standard Internet protocols

The Groove RDB Commands Protocol uses the Groove Dynamics Protocol as its transport. The Groove Dynamics Protocol and the Wide Area Network Device Presence Protocol (WAN DPP) use the Simple Symmetric Transport Protocol (SSTP) as their transport. The SSTP Security Protocol is a subset of SSTP. SSTP can use either TCP or HTTP Encapsulation of SSTP as its transport. The HTTP Encapsulation of SSTP, the Client to Management Server Groove SOAP Protocol, and the Management Server to Relay Server Groove SOAP Protocol use HTTP as their transport.

The Groove RDB Commands and the Groove Dynamics protocols are designed to synchronize shared spaces. The WAN DPP is designed to enable protocol clients to discover the **presence (1)** of other protocol clients. The SSTP is designed as a data transport protocol. If direct SSTP connections are not possible because of firewalls and proxy servers, the HTTP Encapsulation of SSTP can be used to pass through those firewalls and proxy servers. The Client to Management Server Groove SOAP Protocol is designed for user management. The Management Server to Relay Server Groove SOAP Protocol is designed for relay service management.

2.1.1 Protocol Client

A protocol client provides a more secure environment for users to share data and work with team members by connecting through local area networks (LANs) or over the Internet. Users can create shared spaces and invite other users to participate in those shared spaces. Members of a shared space can then create or edit documents that are part of that shared space and add new tools to the shared space as necessary. Members of a shared space can also have different roles, such as manager or participant, within that shared space. Shared spaces are synchronized automatically for all members.

Each user is associated with a user account and has an **identity** that is associated with that user account. An identity is a digital persona for the user. It contains the user's contact information, such as name, e-mail address, phone number, and organization name. A user's name, as it appears in shared spaces and contact lists, is based on the user's identity.

A protocol client can communicate with other protocol clients or a data bridge server directly. If a protocol client cannot communicate directly with another protocol client or it needs to use relay services, such as client presence (1) and data synchronization **fanout** services, it can communicate with relay servers directly. A protocol client knows which relay servers to use based on information that is provided when a user account is created and configured by a management server.

A protocol client can also communicate with an associated management server for user management. The protocol client obtains the URL for a management server from user input or the user's configuration data.

Protocol clients use the following system protocols:

- Client to Management Server SOAP Protocol, as described in [MS-GRVSPCM]
- Groove Dynamics Protocol, as described in [MS-GRVDYNM]
- Groove RDB Commands Protocol, as described in [MS-GRVRDB]
- HTTP Encapsulation of SSTP, as described in [MS-GRVHENC]
- Simple Symmetric Transport Protocol (SSTP), as described in [MS-GRVSSTP]
- SSTP Security Protocol, as described in [MS-GRVSSTPS]
- Wide Area Network Device Presence Protocol (WAN DPP), as described in [MS-GRVWDPP]

Protocol clients communicate with a relay server by using the following system protocols:

- HTTP Encapsulation of SSTP, as described in [MS-GRVHENC]
- Simple Symmetric Transport Protocol (SSTP), as described in [MS-GRVSSTP]
- SSTP Security Protocol, as described in [MS-GRVSSTPS]
- Wide Area Network Device Presence Protocol (WAN DPP), as described in [MS-GRVWDPP]

The following figure shows the protocol stack for protocol clients.

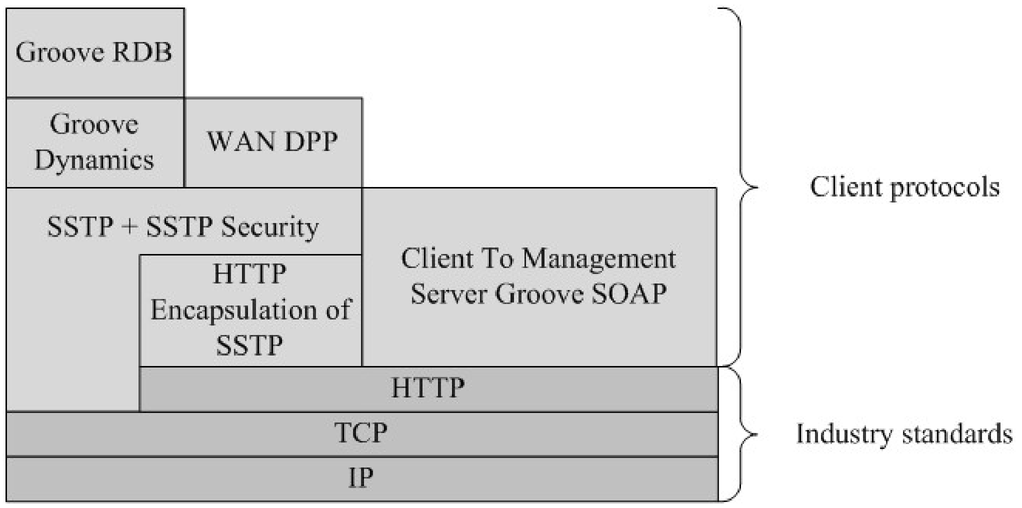

Figure 5: Protocol stack for protocol clients

2.1.2 Management Server

A management server provides the following client management services, as described in [MS-GRVSPCM]:

- Creation, configuration, activation, password reset, and backup of user accounts
- Domain and policy management for users
- User directory
- Usage statistics and auditing of users

In an enterprise deployment, a management server is used to manage protocol clients; a management server supports tasks such as creating domains and groups, and applying specific usage rules.

Domain management, policy management, and auditing services are designed primarily for enterprise deployments. Policies can be used to enforce certain usage rules, such as password strength and back-up schedules. In addition, policies can be set for domains, groups, and individual users. Policies are sent to protocol clients when accounts are created and, subsequently, when protocol clients check for updates with the server.

User directory services enable users to publish their identities to a management server, which also enables users to search for and find identities by using protocol clients.

A management server is also used to manage a relay server. It can be used to add users to relay servers, enable and disable relay services, and purge user data on relay servers. A management server communicates with a relay server by using the Management Server to Relay Server Groove SOAP Protocol, as described in [MS-GRVSPMR]. In this case, the management server is a client of the relay server.

A management server needs to be configured to know which relay servers it manages. A management server communicates with a relay server when it is used to configure relay server settings, create or update the user database on the relay server, or change relay services for a user.

The following figure shows the protocol stack for the management server:

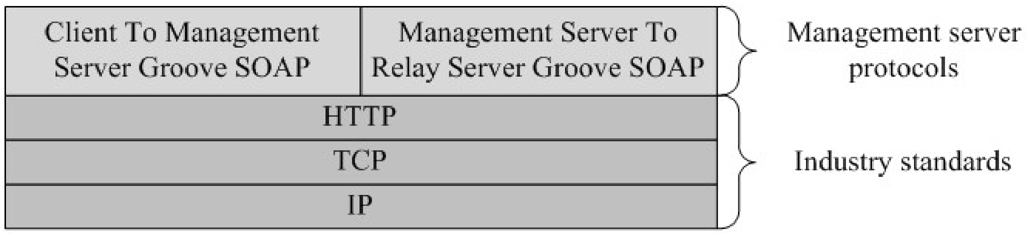
Client To Management Server Groove SOAP
Management Server To Relay Server Groove SOAP
HTTP
TCP
IP
Management server protocols
Industry standards

Figure 6: Protocol stack for the management server

2.1.3 Relay Server

Before it sends other relay-service management requests, a management server registers with a relay server. After a management server registers with a relay server, it can use configuration management services to configure the relay server properly and to use additional services that are provided by the relay server.

A relay server provides the following services:

- Server registration and configuration management
- User database and data management
- Storage and forwarding of user data
- Synchronization fanout of user data
- Client presence

User database and data management services support tasks such as adding users, enabling and disabling users, and purging user data on the relay server.

The store-and-forward service enables indirect, peer-to-peer data communications. If a protocol client cannot communicate with another protocol client for any reason, such as the presence of firewalls or connection states that block communications, the data for the unreachable client is sent to and stored on a relay server. The stored data is then forwarded to the destination protocol client when that client contacts the relay server. The data that is sent to the relay server is encrypted by the protocol client and is opaque to the relay server. Undeliverable data is purged based on purge settings for the relay server.

The synchronization fanout service enables protocol clients to send user data to a relay server and it enables a relay server to send that data to all of the protocol clients that require the data. This reduces the required bandwidth for data synchronization by protocol clients.

The client presence service enables protocol clients to publish and subscribe to presence (1) information, which indicates the online or offline status of each protocol client.

The following figure shows the protocol stack for a relay server.

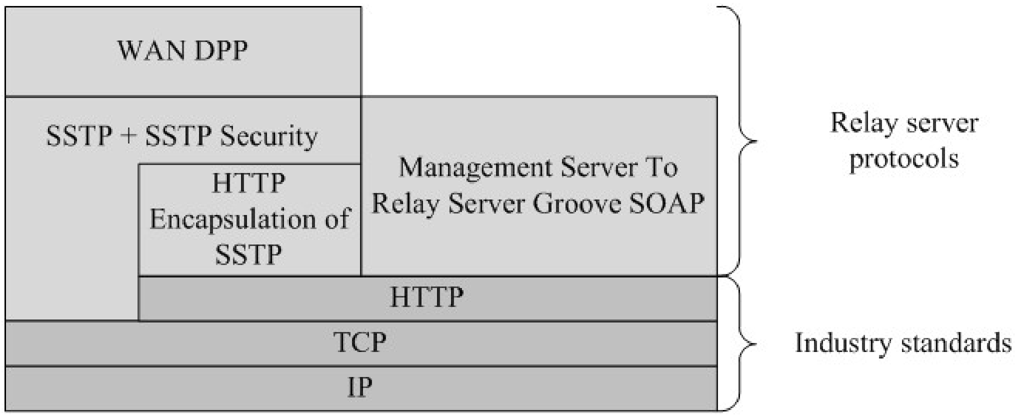

Figure 7: Protocol stack for the relay server

2.1.4 Data Bridge Server

A data bridge server enables external applications to integrate with the system. It provides services that enable external applications to create or access shared spaces and to import or export data.

These integration services are provided by Groove Web Services, which are based on the Web Services Description Language (WSDL). For more information about Groove Web Services, see [MSDN-GWSDF].

The following figure illustrates how external applications can integrate with the system by using a data bridge server.

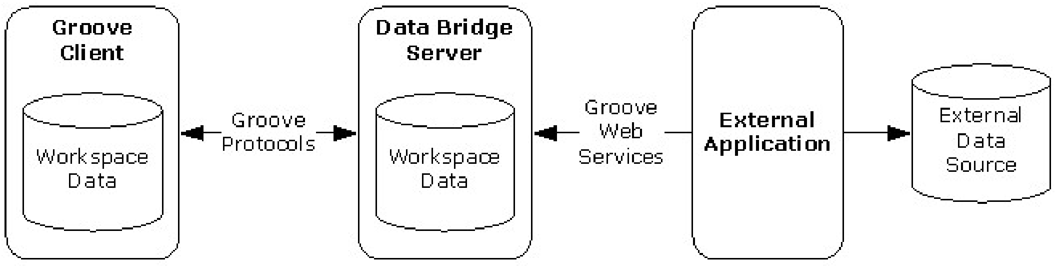

Figure 8: Integration with external applications

A data bridge server is a peer to protocol clients and it has the same protocol stack as a protocol client, as described in section 2.1.1.

2.2 Protocol Summary

The following table provides a comprehensive list of the member protocols of the Microsoft® Groove® and Microsoft SharePoint® Workspace system.

Protocol name	Description	Short name
Groove RDB Commands Protocol	An application-layer distributed protocol that uses encoded XML messages to specify database operations. It is used to create or update database records for shared spaces.	[MS-GRVRDB]

Protocol name	Description	Short name
Groove Dynamics Protocol	An application-layer distributed protocol that uses encoded XML messages to sequence operations consistently for an arbitrary number of peers. It is used to sequence create or update commands for shared spaces and to group those commands into operational units for transactional consistency.	[MS-GRVDYNM]
Wide Area Network Device Presence Protocol (WAN DPP)	A session-based message protocol that enables protocol clients to publish and subscribe to presence (1) information about the online and offline status of protocol clients.	[MS-GRVWDPP]
Simple Symmetric Transport Protocol (SSTP)	A TCP-based, message-oriented, application-layer binary protocol that enables two higher-level applications to engage in bidirectional communication and to exchange data over multiple logical sessions on a single network connection.	[MS-GRVSSTP]
Simple Symmetric Transport Protocol (SSTP) Security Protocol	A subset of the SSTP, as described in [MS-GRVSSTP], that enables a protocol client and a relay server to exchange secret keys when authenticating with each other through a challenge/response message sequence.	[MS-GRVSSTPS]
HTTP Encapsulation of Simple Symmetric Transport Protocol	A collection of protocols and methodologies that is used to route SSTP operations through firewalls and proxy servers. It is used when SSTP, as described in [MS-GRVSSTP], cannot use TCP directly as its transport.	[MS-GRVHENC]
Client to Management Server Groove SOAP Protocol	A request/response message-exchange protocol that is based on a custom implementation of an early version of **Simple Object Access Protocol (SOAP)**. It provides interfaces for managing protocol clients.	[MS-GRVSPCM]
Management Server to Relay Server Groove SOAP Protocol	A request/response message-exchange protocol that is based on a custom implementation of an early version of SOAP. It provides interfaces for managing relay servers.	[MS-GRVSPMR]

2.3 Environment

The following sections identify the context in which the Microsoft® Groove® and Microsoft SharePoint® Workspace system exists. This includes systems that use the interfaces that are provided by this system, other systems that this system depends upon, and, as appropriate, how system components communicate with each other.

2.3.1 Dependencies on This System

None.

2.3.2 Dependencies on Other Systems/Components

The Microsoft® Groove® and Microsoft SharePoint® Workspace system depends on the following protocols and components:

- **HTTP**
- **TCP/IP**
- TCP/IP networks

- **Domain Name System (DNS)** servers
- Local area network and Internet routers
- Computer operating systems that support TCP/IP networking

2.4 Assumptions and Preconditions

The assumptions and preconditions that are described in this section apply to the Microsoft® Groove® and Microsoft SharePoint® Workspace system overall. For information about assumptions and preconditions that apply to a specific protocol within this system, see the specification for that protocol.

The Groove and SharePoint Workspace system is comprised of Internet-based collaboration software and protocols. It is designed to work with TCP/IP networks. Protocol clients and protocol servers in the system need to be able to communicate with each other through local area networks (LANs) and over the Internet. In addition, IP addresses are required at runtime for protocol client and protocol server computers. The system depends on a Domain Name System (DNS) to translate domain names to IP addresses. It is assumed that DNS servers are available on LANs and over the Internet.

If a protocol server is behind a firewall or proxy server, the firewall or proxy server needs to be configured to connect or proxy data from protocol clients to protocol servers by using at least one of the mechanisms supported by the Simple Symmetric Transport Protocol (SSTP), as described in [MS-GRVSSTP], HTTP Encapsulation of SSTP, as described in [MS-GRVHENC], or the Client to Management Server Groove SOAP Protocol, as described in [MS-GRVSPCM]. The firewall or proxy server also needs to be able to resolve server names to IP addresses and to establish connections to protocol servers by using TCP/IP. Finally, the firewall or proxy server needs to be configured to allow connections to at least one well-known port (**SSTP**, HTTP, or **SSL**) on the protocol server.

2.5 Use Cases

The following use cases are provided to facilitate understanding of the Microsoft® Groove® and Microsoft SharePoint® Workspace system overall:

- Create an account
- Publish presence status
- Create a shared space
- Display presence information
- Search for users
- Add contacts
- Invite users to join a shared space
- Update a shared space
- Create a domain
- Add users to a domain
- Add a policy to a domain
- Delete policies from a domain

- Add a relay server
- Add users to a relay server

These use cases are not intended to provide a thorough and complete model of the system for any implementation.

The following diagram illustrates the primary actors in and relationships between the uses cases.

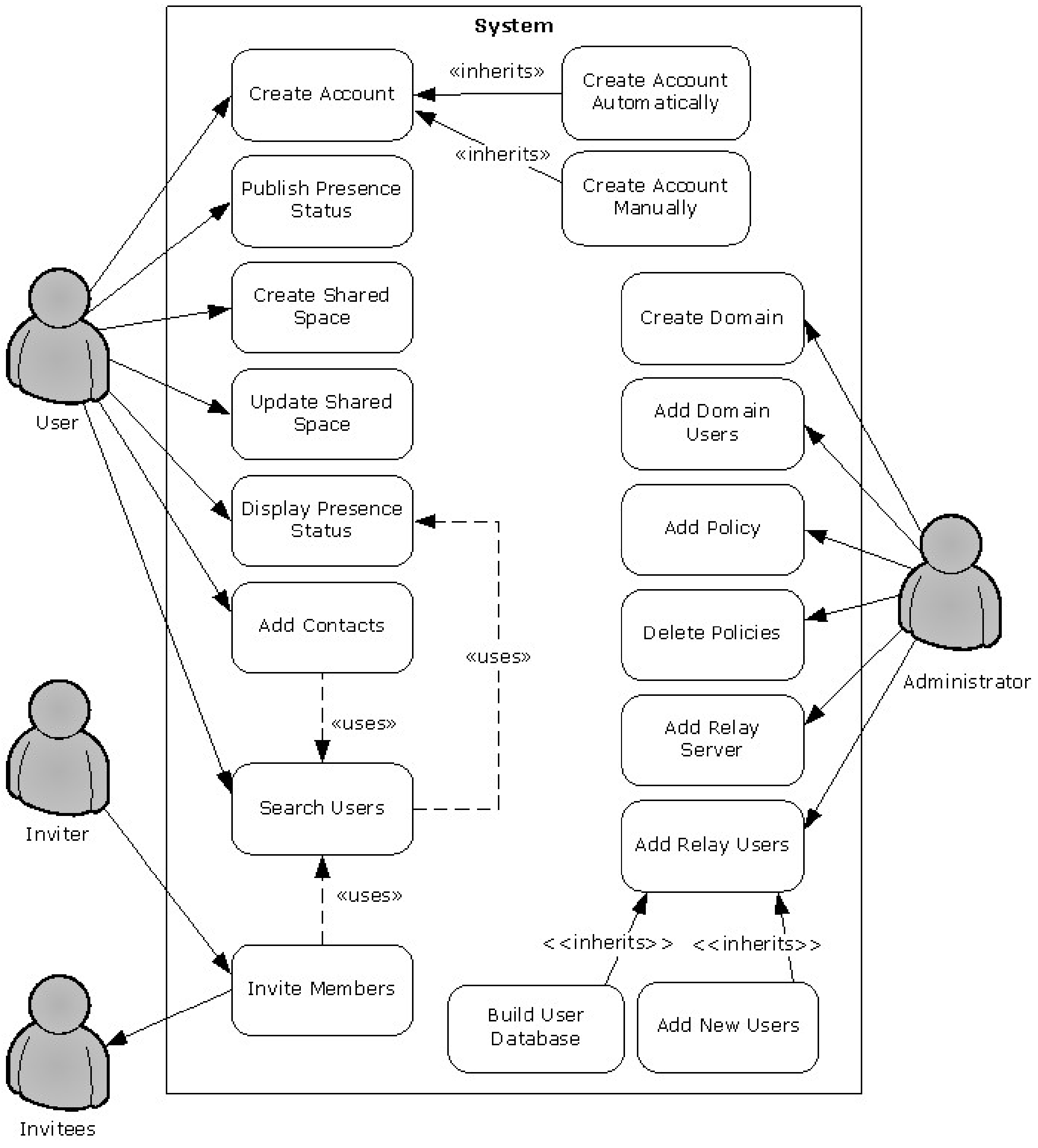

Figure 9: Use cases for the system

2.5.1 Create an Account

This use case describes how a user account is created. A user account can be created manually or automatically. For manual account creation, the user enters an **account configuration code** and the URL of the management server. For automatic account creation, the protocol client receives the URL for the management server from the user's configuration data and the management server needs to be able to authenticate users securely. For either type of account creation, the actor is a user.

The following diagram illustrates the account creation process.

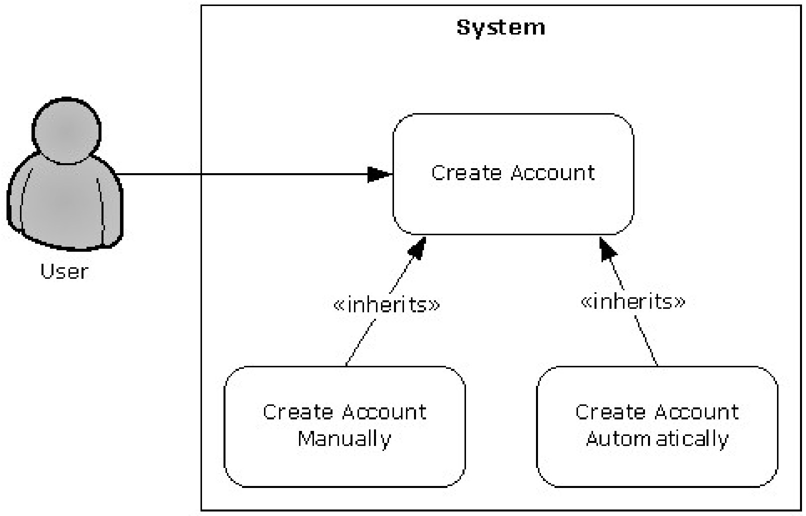

Figure 10: Process for creating an account

Preconditions

- The management server database contains the user data for the account.
- The protocol client can connect to the management server.
- The user is configured for automatic account configuration or has received an account configuration code and the URL of the management server.
- For automatic account configuration, the management server can both receive authenticated user information from an authenticated connection and use that information to identify the user in its database.

Steps

1. The user starts a new installation of a protocol client.
2. The protocol client detects that an account has not been created for the user.
3. The protocol client checks if the user is configured for automatic account configuration.
4. The protocol client gets account creation data:
   - For automatic account creation, the protocol client receives the URL of the management server from the user's configuration data.
   - For manual account creation, the protocol client receives an account configuration code and the URL of the management server from the user.
5. The protocol client sends an account configuration request to the management server:

- For automatic account creation, the protocol client uses an authenticated, secure channel to communicate with the management server.
- For manual account creation, the protocol client sends the **account configuration** code to the management server.

6. The management server receives and processes the request from the protocol client:
   - For automatic account creation, the management server receives the user information through the authenticated communication channel.
   - For manual account creation, the management server receives the account configuration code.
7. The management server responds to the protocol client with the following information about the user:
   - Identity
   - Domain
   - Usage policies
8. The protocol client receives the response from the management server, creates the user's account, and associates the user's account with the user's identity, domain, and usage policies.
9. The protocol client sends an account registration request and the GUID for the user's account to the management server.
10. The management server registers the user's account in its database.
11. The management server returns a successful response to the protocol client.
12. The protocol client notifies the management server that the identity and policies that it received from the management server have been installed successfully on the protocol client.
13. The management server returns a successful response to the protocol client.
14. The protocol client sends a domain enrollment request to the management server to activate the user in the domain.
15. The management server receives the request and processes it by activating the user in the user's domain and signing the user's identity data with the domain certificate.
16. The management server responds to the protocol client with the signed identity data.
17. The protocol client receives the response from the management server and updates the user's identity with the signed data.
18. The protocol client notifies the user that the account was created successfully and is ready for use.

Alternative scenarios

The user can cancel the account creation process before submitting the account creation request to the management server.

Errors

- Cannot connect to the management server

1. The protocol client notifies the user that it cannot connect to the server.
2. The protocol client provides the user with options for updating input, trying again, or cancelling the account creation process.
3. If the user chooses to try again, the protocol client tries to connect to the server again and with any updated input. If the user chooses to cancel the process, the protocol client ends the account creation process.

- Invalid request from the protocol client

1. The management server discards the invalid request and returns a fault response to the protocol client.
2. The protocol client sends the request again or ends the account creation process and notifies the user of the error.

- Invalid response from the management server

1. The protocol client discards the invalid server response.
2. The protocol client either sends the request again or ends the account creation process and notifies the user of the error.

- Server errors

1. The management server discards the request and returns a fault response, if possible, or no response.
2. The protocol client sends the request again or ends the account creation process and notifies the user of the error.

- Client errors – The protocol client ends the account creation process and notifies the user of the error.

Post-conditions

- The user account is created and the user is notified that the protocol client is ready for use.
- The user account is not created and the user is notified of the error.

2.5.2 Publish Presence Status

This use case describes how a user’s presence (1) information is published. The actor is a user.

The following diagram illustrates this process.

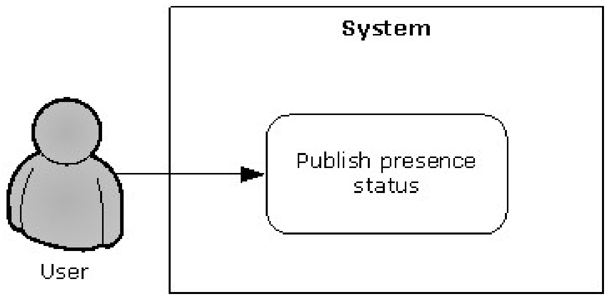

Figure 11: Process for publishing presence information

Preconditions

- The user configured an account on the protocol client, either manually or automatically, as described in section 2.5.1.
- The protocol client can communicate with the relay servers that are associated with the user.

Steps

1. The user starts the protocol client.
2. The protocol client sends the user's online status to the relay server that is associated with the user.
3. The relay server updates the presence (1) information records and notifies all subscribers of the user's presence (1) information.
4. The user selects the option to work offline or exits the protocol client.
5. The protocol client publishes the user's offline status to the relay server that is associated with the user.
6. The relay server updates the presence (1) information records for the user and notifies all subscribers of the user's presence (1) information.
7. The protocol client closes.

Errors

- Cannot connect to the relay server – The protocol client tries again, when it can connect to the relay server.
- Relay server errors – The protocol client tries again when possible.
- Client errors – The protocol client notifies the user of the error and tries again when possible.

Post-conditions

- The user's presence (1) information is published to the relay server.
- The user's presence (1) information is not published to the relay server because an error occurred.

2.5.3 Create a Shared Space

This use case describes how a user creates a shared space by using a protocol client. The actor is a user.

The following diagram illustrates this process.

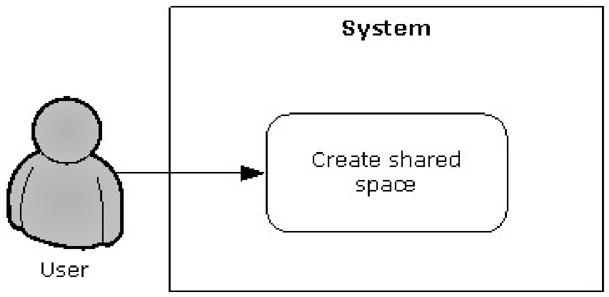

Figure 12: Process for creating a shared space

Preconditions

- The user configured an account on the protocol client, either manually or automatically, as described in section 2.5.1.
- The protocol client can communicate with the relay servers that are associated with the user.
- The user can create shared spaces by using the protocol client.

Steps

1. The user selects the option in the protocol client UI to create a shared space.
2. The protocol client creates the shared space and opens it for the user.

Errors

The protocol client ends the process and notifies the user of the error.

Post-conditions

- A shared space is created for the user and opened in the protocol client.
- A shared space is not created and the user is notified that an error occurred.

2.5.4 Display Presence Information

This use case describes how a user obtains and displays presence (1) information for other users. The actor is a user.

The following diagram illustrates this process.

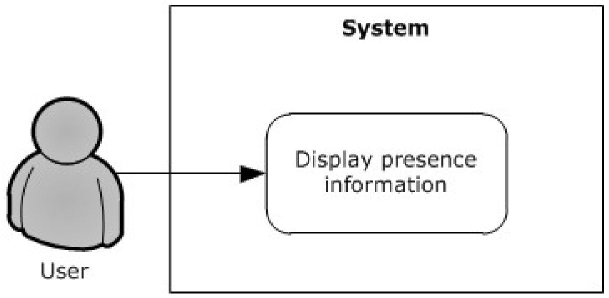

Figure 13: Process for displaying presence information

Preconditions

- The protocol client can communicate with the relay servers that are associated with the user.
- The user can see presence (1) information for other users.

Steps

1. The user selects an option in the protocol client UI to display a list of other users.
2. The protocol client subscribes to presence (1) information from the relay servers that are associated with the users to display.
3. The relay server updates the presence (1) subscription records and notifies the protocol client of the presence (1) information for the subscribed users.
4. The protocol client receives the presence (1) information for the users, and then displays the list of users and presence (1) information for each of those users.
5. The user closes the UI that displays the list of users.
6. The protocol client unsubscribes from the presence (1) information for the other users.
7. The relay server updates the presence (1) subscription records and removes the corresponding presence subscriptions for the protocol client.

Errors

- Cannot connect to the relay server – The protocol client sets the presence (1) information to a status of unknown and tries again when it can connect to the relay server.
- Relay server errors – The protocol client sets the presence (1) information to a status of unknown and tries again.
- Client errors – The protocol client notifies the user of the error, sets the presence (1) information to a status of unknown, and tries again.

Post-conditions

- The user sees presence (1) information for other users.
- The user cannot see presence (1) information for other users because an error occurred.

2.5.5 Search for Users

This use case describes how a user searches for other users and displays presence (1) information for those users, as described in section 2.5.4. The actor is a user.

The following diagram illustrates this process.

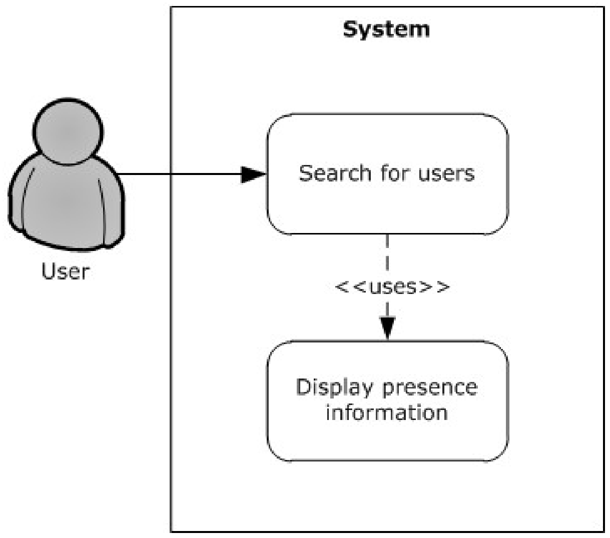

Figure 14: Process for searching for users

Preconditions

- The protocol client can communicate with its associated management server.
- The user can search for users by using the protocol client.

Steps

1. The user selects an option in the protocol client UI to search for users.
2. The protocol client displays a search menu and waits for user input.
3. The user enters the search criteria and then submits it.
4. The protocol client sends the search request to the management server.
5. The management server uses the search criteria to search for users in the user database on the management server.
6. The management server returns the search results to the protocol client.
7. The protocol client displays the search results to the user and performs the steps outlined in section 2.5.4 to display information about the users who are listed in the search results.
8. The user selects users from the search results to view contact information for them, send messages to them, or add them to a contact list.
9. The user closes the search UI and ends the search.

Alternative scenarios

The user can cancel the search before submitting it.

Errors

- Cannot connect to the management server – The protocol client ends the search and notifies the user of the error.
- Invalid client request – The management server discards the invalid request and returns a fault response to the protocol client. The protocol client then tries the request again or ends the search and notifies the user of the error.
- Invalid server response – The protocol client discards the invalid server response, and then either tries the request again or ends the search and notifies the user of the error.
- Server errors – The management server discards the request and returns either a fault response or no response. The protocol client then tries the request again or ends the search and notifies the user of the error.
- Client errors – The protocol client ends the search and notifies the user of the error.

Post-conditions

- The search results are displayed to the user and the user ends the search.
- The search results are not displayed to the user and the protocol client ends the search.

2.5.6 Add Contacts

This use case describes how a user searches for users, as described in section 2.5.5, and adds users from a list of search results to a contact list in the protocol client. The actor is a user.

The following diagram illustrates this process.

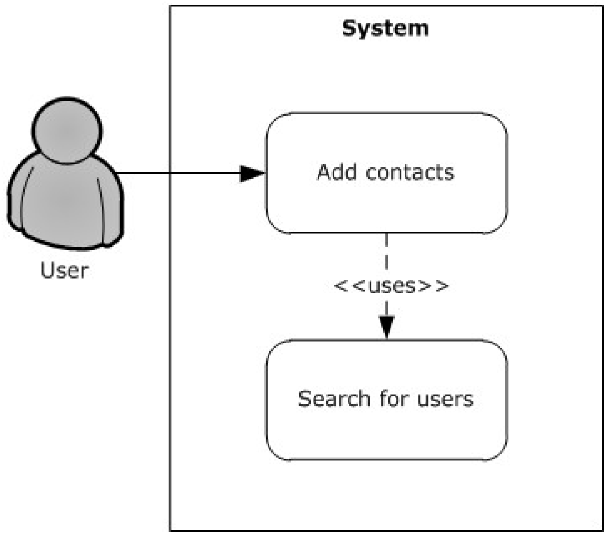

Figure 15: Process for adding users to a contact list

Preconditions

The user can add users to a contact list by using the protocol client.

Steps

1. In the protocol client UI, the user selects the option to add contacts to the contact list.

2. The protocol client displays the add contacts UI and then waits for user input.
3. The user searches for contacts, as described in section 2.5.5.
4. The protocol client displays the search results to the user, as described in section 2.5.5.
5. The user selects users from the search results and adds them to the contact list.
6. The user closes the UI for adding contacts.

Alternative scenarios

The user can cancel the request to add contacts before submitting it.

Errors

- Search errors – The protocol client notifies the user of the error.
- Client errors – The protocol client closes the UI for adding contacts and notifies the user of the error.

Post-conditions

- Users are added to the contact list in the protocol client.
- The contact list is not changed because an error occurred or no users are found.

2.5.7 Invite Users to Join a Shared Space

This use case describes how a user searches for users, as described in section 2.5.5, and invites other users to become members of a shared space. The actors are the inviter, which is the user who invites other users to join the shared space, and one or more invitees, which are the users who receive the invitation from the inviter.

The following diagram illustrates this process

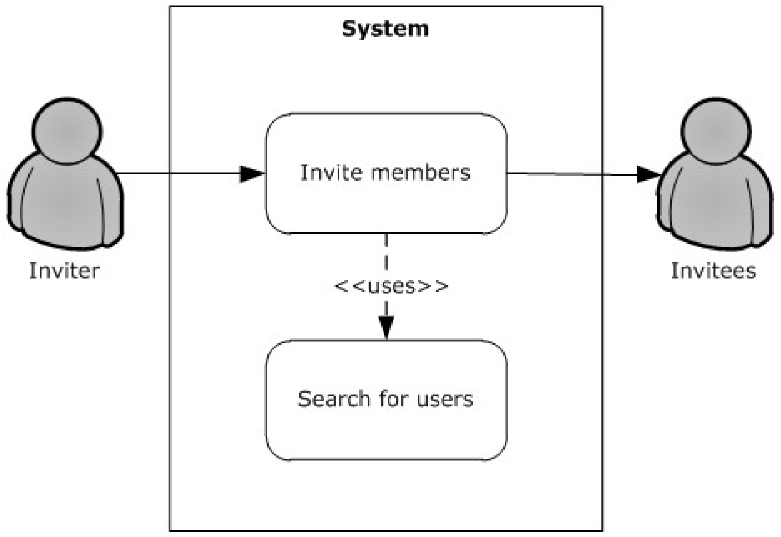

Figure 16: Process for inviting users to join a shared space

Preconditions

The inviter can invite other users to become members of a shared space by using a protocol client.

Steps

1. In the protocol client, the inviter opens the shared space.
2. In the shared space, the inviter selects an option to invite others to join that shared space.
3. The protocol client displays an invitation UI that enables the inviter to select invitees from a contact list or to search for invitees, as described in section 2.5.5.
4. The inviter adds invitees from the contact list or a list of search results, and optionally enters an invitation message and chooses to require acceptance confirmations from invitees.
5. The inviter submits the invitation.
6. The protocol client creates the invitation and sends it to the invitees. If an invitee's protocol client cannot be reached, the inviter's protocol client sends the invitation to a relay server for delivery.
7. An invitee's protocol client receives the invitation and notifies the invitee.
8. The invitee opens the invitation and accepts or declines the invitation.
9. The invitee's protocol client sends the response to the inviter.
10. The inviter's protocol client processes the invitee's response:
11. The invitee accepted the invitation. If confirmation is needed, an invitation confirmation message is displayed. The inviter can verify the invitee's contact information and confirm or deny the invitee's acceptance. If the inviter confirms the acceptance, the shared space needs to be sent to the invitee. If confirmation is not needed, the shared space needs to be sent to the invitee.
12. The invitee declined the invitation. The shared space does not need to be sent to the invitee. Optionally, notification of the declined invitation is displayed to the inviter.
13. For each member that accepted the invitation, the inviter's protocol client adds that member to the shared space and sends the shared space to that member. If a relay server is available, the protocol client can send the shared space to the relay server for synchronization with all members.
14. An invitee's protocol client receives the shared space and notifies the invitee that the shared space is ready for use. The invitee's protocol client also acknowledges the inviter's protocol client.

Alternative scenarios

- Invitation file – In the shared space UI, the inviter uses an invitation option to create an invitation file in a location that can be accessed by invitees.
- Invitation e-mail – In the shared space UI, the inviter uses an invitation option to send an invitation as an e-mail message to invitees.
- Cancellation – The user can cancel the invitation before sending it.

Errors

- Invalid invitation – The invitation is discarded by invitee protocol clients.

- Invalid response to an invitation – The inviter's protocol client discards the response.
- Relay server errors – The relay server discards the request and returns an error response or no response. The protocol client then notifies the user of the error and tries again when possible.
- Client errors – The protocol client tries again when possible and notifies the user of the error.

Post-conditions

- The shared space is sent to invitees who accepted the invitation and were confirmed by the inviter. The list of shared space members is updated with information about the new members.
- The shared space is not sent to any invitees because an error occurred. The list of shared space members is not changed.

2.5.8 Update a Shared Space

This use case describes how a shared space is updated and how those updates are synchronized across members of that shared space. The actor is a user who updates a shared space by using a protocol client.

The following diagram illustrates this process.

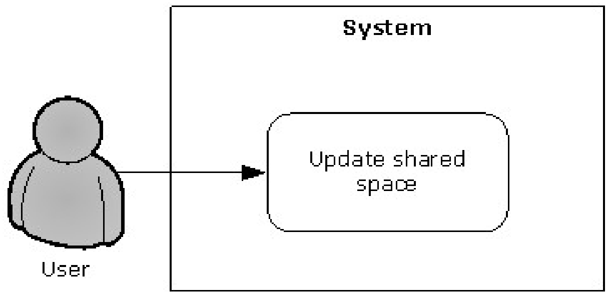

Figure 17: Process for updating a shared space

Preconditions

The user is a member of the shared space and can update that shared space by using a protocol client.

Steps

1. The user updates components, such as documents and tools, of a shared space.
2. The protocol client creates an ordered set of update commands for the shared space. The set of commands corresponds to the updates that the user wants to make.
3. The protocol client sends the ordered set of update commands to all members of the shared space. If a relay server is available, the protocol client can send the update commands to the relay server for delivery to all members.
4. Each member's protocol client receives the set of update commands and sends an acknowledgement to the sender.

5. Each member's protocol client executes the update commands in the specified order. If necessary, the protocol clients also back up previous changes to a point where the update commands can be rearranged in the correct execution order for synchronizing the shared space.
6. The shared space is updated and members of the shared space are notified of the changes.

Errors

- Cannot connect to member protocol clients or relay servers – The protocol client queues the update commands and sends them when possible.
- Invalid synchronization commands – The commands are rejected by the protocol clients of other members.
- Invalid synchronization command acknowledgements – The acknowledgements are discarded by the protocol clients of other members.
- Relay server errors - The relay server discards the request and returns either an error response or no response. The protocol client tries the request again or communicates directly with the protocol clients of other members.
- Client errors – If an error occurred while the user was changing the shared space, the protocol client ends the process and notifies the user of the error. The shared space is not changed.

Post-conditions

The shared space is synchronized across all members of the shared space.

2.5.9 Create a Domain

This use case describes how a domain is created on a management server. The actor is an administrator.

The following diagram illustrates this process.

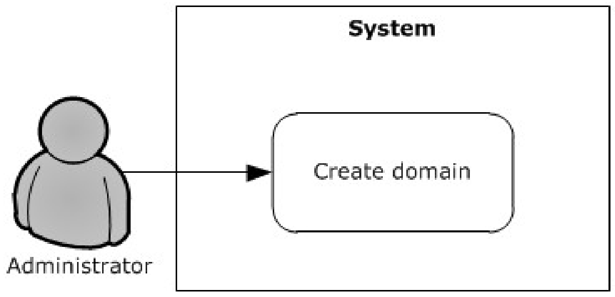

Figure 18: Process for creating a domain

Preconditions

The management server is installed and configured correctly.

Steps

1. The administrator starts the administration application on the management server.
2. The administration application displays the administration UI for the management server.

3. The administrator selects the domain creation option in the administration application UI.
4. The administration application displays the domain creation UI.
5. The administrator enters the required information in the domain creation UI.
6. The administrator submits the information about the domain.
7. The administration application validates the information. If the information is not valid, it notifies the administrator and waits for valid information.
8. The administration application creates the domain and displays it to the administrator.

Alternative scenarios

The administrator can cancel the process before submitting data about the domain.

Errors

- Cannot connect to the management server – The administration application notifies the administrator of the error and the domain is not created.
- Management server errors – The administration application notifies the administrator of the error and the domain is not created.
- Administration application errors – The administration application notifies the administrator of the error and the domain is not created.

Post-conditions

- The domain is created and displayed to the administrator.
- The domain is not created and the administrator is notified of the error.

2.5.10 Add Users to a Domain

This use case describes how users are added to a domain on a management server. The actor is an administrator.

The following diagram illustrates this process.

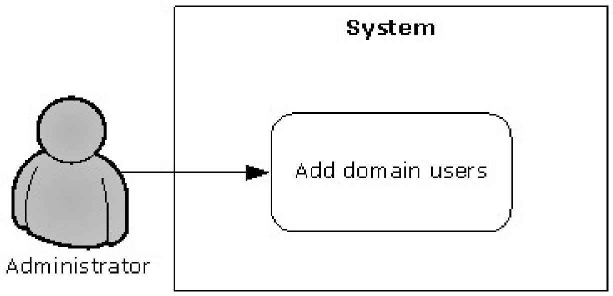

Figure 19: Process for adding users to a domain

Preconditions

A domain was created on the management server, as described in section 2.5.9.

Steps

1. The administrator starts the administration application on the management server.
2. The administration application displays the administration UI for the management server.
3. The administrator selects the domain to add members to.
4. The administration application displays the domain management UI.
5. The administrator selects the option to add users.
6. The administration application displays the UI for adding domain users.
7. The administrator enters the required information for either a single user or, by using a defined format, multiple users.
8. The administrator submits the required information.
9. The administration application validates the information. If the information is not valid, it notifies the administrator and waits for valid information.
10. The administration application adds the specified user or users to the domain and displays an updated list of domain users to the administrator.

Alternative scenarios

The administrator can cancel the process before submitting information about one or more users.

Errors

- Cannot connect to the management server – The administration application notifies the administrator of the error and the users are not added.
- Management server errors – The administration application notifies the administrator of the error and the users are not added.
- Administration application errors – The administration application notifies the administrator of the error and the users are not added.

Post-conditions

- Users are added to the domain and displayed in the list of domain users.
- No users are added to the domain and the administrator is notified of the error.

2.5.11 Add a Policy to a Domain

This use case describes how a policy is added to a domain on a management server. The policy is then installed on protocol clients when those clients checks for updates with the management server. The actor is an administrator.

The following diagram illustrates the process of adding a policy to a domain.

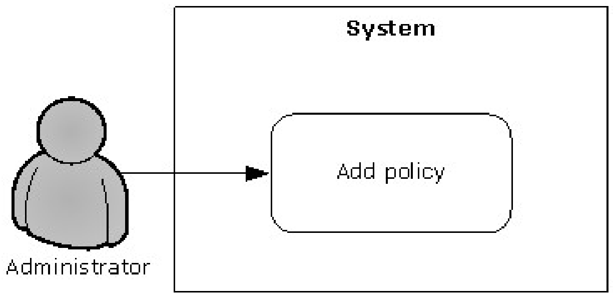

Figure 20: Process for adding a policy to a domain

Preconditions

A domain was created on the management server, as described in section 2.5.9.

Steps

1. The administrator starts the administration application on the management server.
2. The administration application displays the administration UI for the management server.
3. The administrator selects the domain to add a policy to.
4. The administration application displays the domain management UI.
5. The administrator selects the policy management option.
6. The administration application displays the policy management UI.
7. The administrator selects the option to add a policy.
8. The administration application displays the add policy UI.
9. The administrator enters the required information about the policy and submits that information.
10. The administration application validates the information. If the information is not valid, the administration application notifies the administrator and waits for valid information.
11. The administration application creates the policy and displays it to the administrator.

Alternative scenarios

The administrator can cancel the process before submitting policy information.

Errors

- Cannot connect to the management server – The administration application notifies the administrator of the error and the policy is not added.
- Management server errors – The server discards the request and returns an error response or no response. The administration application notifies the administrator of the error and the policy is not added.
- Administration application errors – The administration application notifies the administrator of the error and the policy is not added.

Post-conditions

- A policy is added to the domain and displayed in the list of domain policies.
- A policy is not added to the domain and the administrator is notified of the error.

2.5.12 Delete Policies from a Domain

This use case describes how an administrator deletes policies from a domain on a management server. The deleted policies are removed from protocol clients when those clients check for updates with the management server. The actor is an administrator.

The following diagram illustrates the process of deleting policies from a domain.

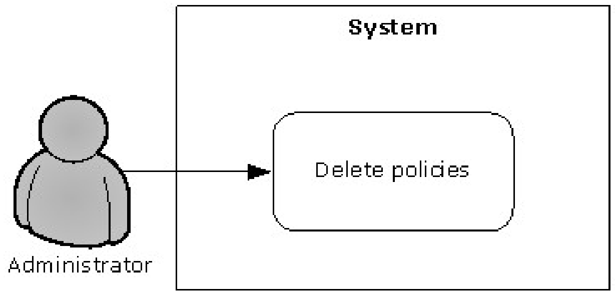

Figure 21: Process for deleting policies from a domain

Preconditions

One or more policies were added to the domain on the management server, as described in section 2.5.11.

Steps

1. The administrator starts the administration application on the management server.
2. The administration application displays the administration UI for the management server.
3. The administrator selects the domain to delete policies from.
4. The administration application displays the domain management UI.
5. The administrator selects the policy management option.
6. The administration application displays the policy management UI.
7. The administrator selects the policies to delete and submits the deletion request.
8. The administration application deletes the selected policies and displays the result to the administrator.

Alternative scenarios

The administrator can cancel the deletion process before submitting the deletion request.

Errors

- Cannot connect to the management server – The administration application notifies the administrator of the error and the policies are not deleted.
- Management server errors – The administration application notifies the administrator of the error and the policies are not deleted.
- Administration application errors – The administration application notifies the administrator of the error and the policy is not deleted.

Post-conditions

- The specified policies are deleted from the domain and the result is displayed to the administrator.
- The specified policies are not deleted from the domain and the administrator is notified of the error.

2.5.13 Add a Relay Server

This use case describes how an administrator adds a relay server to a domain on a management server. The actor is an administrator.

The following diagram illustrates this process.

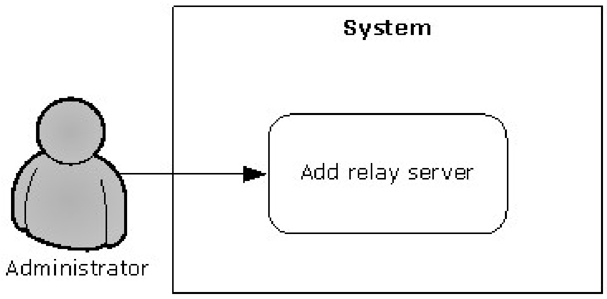

Figure 22: Process for adding a relay server

Preconditions

- The relay server is configured and ready to be managed by the management server.
- A domain exists on the management server.

Steps

1. The administrator starts the administration application on the management server.
2. The administration application displays the administration UI for the management server.
3. The administrator selects the domain to add a relay server to.
4. The administration application displays the domain management UI.
5. The administrator selects the relay server management option.
6. The administration application displays the relay server management UI.
7. The administrator selects the option to add a relay server.

8. The administration application displays the add relay server UI.
9. The administrator enters the required information about the relay server and submits that information.
10. The administration application validates the information about the relay server. If the information is not valid, the administration application notifies the administrator and waits for the administrator to submit valid information.
11. The administration application adds the relay server to the domain and displays it to the administrator.

Alternative scenarios

The administrator can cancel the process of adding a relay server before submitting the request.

Errors

- Cannot connect to the management server – The administration application notifies the administrator of the error and the relay server is not added to the domain.
- Server errors – The administration application notifies the administrator of the error and the relay server is not added to the domain.
- Administration application errors – The administration application notifies the administrator of the error and the relay server is not added to the domain.

Post-conditions

- A relay server is added to the domain on the management server and the result is displayed to the administrator.
- A relay server is not added to the domain and the administrator is notified of the error.

2.5.14 Add Users to a Relay Server

This use case describes how users are added to a relay server. The actor is an administrator.

The following diagram illustrates this process.

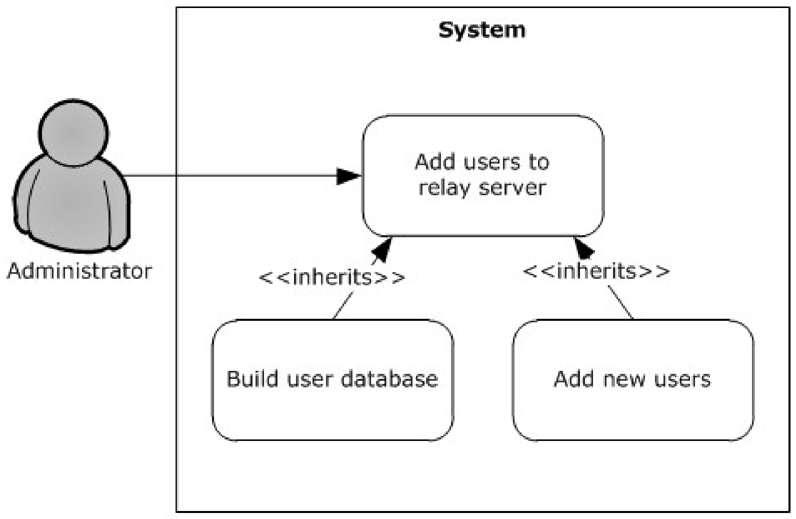

Figure 23: Process for adding users to a relay server

Preconditions

The relay server is configured and ready to be managed by a management server.

Steps

1. The administrator adds users to a domain on the management server, as described in section 2.5.10, by using the administration application.

2. The management server sends an add-user request to the relay server. If all of the users cannot be added in one request, the management server sends additional add-user requests to the relay server.

3. The relay server receives the add-user request and processes it.

If the user database for the relay server exists and does not need to be rebuilt, the relay server adds the users to the existing user database and returns a response to the management server indicating that the users were added successfully.

If the user database for the relay server needs to be built or rebuilt, the following steps also occur:

1. The relay server returns a response to the management server indicating that the user database needs to be built.

2. The management server sends a request to put the relay server in an inactive state before building the user database.

3. The relay server receives the request, enters the inactive state, and returns a successful response.

4. The management server receives the successful response and sends add-user requests to build the user database on the relay server.

5. The relay server receives the add-user requests, builds or rebuilds the user database, adds users to that user database, and returns successful responses.

6. The management server sends a request to put the relay server in an active state.

7. The relay server receives the request, enters the active state, and returns a successful response.

Errors

- Cannot connect to the relay server – The administration application notifies the administrator of the error.
- Relay server errors – The administration application notifies the administrator of the error.
- Management server errors – The administration application notifies the administrator of the error.

Post-conditions

- Users are added to the relay server.
- Users are not added to the relay server and the administrator is notified of the error.

2.6 Versioning, Capability Negotiation, and Extensibility

Protocol servers and protocol clients that are part of the Microsoft® Groove® and Microsoft SharePoint® Workspace system can be different versions with different features and capabilities. However, different versions need to be compatible with each other. Product version numbers and capability specifications can be used to determine the capabilities of components that are part of the system.

For information about versioning, capability negotiations, and the extensibility of a specific protocol within the system, see the specification for that protocol.

2.7 Error Handling

The following types of errors might occur within the Microsoft® Groove® and Microsoft SharePoint® Workspace system:

- Client errors
- Server errors
- Communication errors between protocol clients
- Communication errors between protocol clients and protocol servers

Errors that occur while a user is using a protocol client need to be reported to the user appropriately. If a critical error occurs and leaves a protocol client in a corrupted state, the protocol client needs to terminate itself. Otherwise, the security of the system is at risk.

Server errors need to be logged and reported to an administrator appropriately. If a critical error occurs and leaves a protocol server in a corrupted state, the protocol server needs to be restarted before it processes any additional requests. Otherwise, the security of the system is at risk.

For information about errors and error handling for a specific protocol within the system, see the specification for that protocol

2.8 Coherency Requirements

None.

2.9 Security

As an Internet-based, collaboration software system, security is an important aspect of the Microsoft® Groove® and Microsoft SharePoint® Workspace system. Security issues are addressed at both the protocol level and the application level. At both levels, sensitive data needs to be encrypted.

2.9.1 Protocol Security

The following sections describe protocol-level security within the system.

2.9.1.1 Groove Dynamics Protocol

The Groove Dynamics Protocol is used to synchronize shared spaces between protocol clients. To help secure data, the protocol encrypts and signs outgoing messages and decrypts and verifies the signatures of incoming messages. For more information, see [MS-GRVDYNM].

2.9.1.2 SSTP Security Protocol

The Simple Symmetric Transport Protocol (SSTP) Security Protocol enables a relay server to authenticate a protocol client before providing any service to that protocol client. The authentication data in the protocol message is encrypted. For more information, see [MS-GRVSSTPS].

2.9.1.3 Groove SOAP Security

There are two SOAP-based protocols within the system, the Client to Management Server Groove SOAP Protocol and the Management Server to Relay Server Groove SOAP Protocol. Both of these protocols require encryption of sensitive data, which helps ensure the confidentiality and integrity of the data. For more information, see [MS-GRVSPCM] and [MS-GRVSPMR].

2.9.2 Application Security

The following sections describe application-level security within the system.

2.9.2.1 Protocol Clients

In an enterprise deployment, sign-on operations for a protocol client can be integrated with a user authentication mechanism for the enterprise. A password can also be used for sign-on operations.

In addition, sensitive data that is stored on disks can be encrypted. User identities within the system are signed and verifiable, and update commands for shared spaces include a sender's identity, which can be used to authenticate the sender.

2.9.2.2 Protocol Servers

Protocol servers need to be deployed and managed in environments that are configured to help prevent attacks and unauthorized access to those servers. In addition, management servers need to be registered with the appropriate relay servers to help secure communications. For more information, see [MS-GRVSPMR].

2.9.2.3 Server Administration Applications

Administration applications on protocol servers need to be managed in environments that are configured to help prevent unauthorized access to those applications and servers. In addition, server administration data that is sent over a network needs to be encrypted.

2.10 Additional Considerations

None.

3 Examples

The examples in the following sections provide more information about use and operation of the Microsoft® Groove® and Microsoft SharePoint® Workspace system, especially interactions between system components. This document provides the following examples:

- Create an account
- Add contacts
- Update a shared space
- Add a policy to a domain
- Add users to a relay server

Each example includes information such as system-level processing that occurs, roles and components that interact with each other, and the state of the system. Each example also provides information about the following types of common errors that might occur:

- Connection errors
- Invalid requests from protocol clients – A request from a protocol client is invalid if the protocol server cannot parse the request message successfully. This includes security checks. In such cases, the protocol server can discard the request or return a fault response.
- Invalid responses from protocol servers – A response from a protocol server is invalid if the protocol client cannot parse the response message successfully. This includes security checks. In such cases, the protocol client discards the response, displays an error message to the user if appropriate, and ends the operation.
- Server fault responses – Possible causes for this type of response are an invalid request from a protocol client, invalid data such as an invalid account configuration code in a request, and a server-side error that occurs while processing a request.
- Protocol client errors – If the error is not critical, the protocol client can display an error message to the user. If the error is critical, the protocol client can display an error message to the user and additionally terminate itself.
- Protocol server errors – This type of error can occur while a protocol server is processing a request. If the error is not critical, the protocol server can return a fault response to the protocol client. If the error is critical, the protocol server cannot respond to the protocol client because the server is in an unstable state.

For examples of how a specific protocol can be used, see the specification for that protocol.

3.1 Example 1: Create an Account

This example describes the process for creating a user account manually. It corresponds to the "Create an Account" use case that is described in section 2.5.1. After the account is created, the protocol client publishes presence (1) information for the account to the associated relay server, which corresponds to the "Publish Presence Status" use case that is described in section 2.5.2.

In this example, an administrator has added the user to a domain on the management server. The user has an account configuration code and the URL of the management server.

The protocol client communicates with the management server by using the Client to Management Server Groove SOAP Protocol, as described in [MS-GRVSPCM]. The protocol client publishes presence (1) information to the relay server by using the Wide Area Network Device Presence Protocol (WAN DPP), as described in [MS-GRVWDPP].

The following steps describe the process for creating an account manually, and assume that no errors occur during the process:

1. The user installs and starts the protocol client.
2. The protocol client detects that the user is not configured to create an account automatically and displays the UI for creating an account manually.
3. The user enters the account configuration code and the URL of the management server.
4. The protocol client receives and validates the information entered by the user. If there are any errors in the information, the protocol client prompts the user for the correct information.
5. The protocol client sends a **KeyActivation** request, as described in [MS-GRVSPCM], to the management server. The request contains the user's account configuration code.
6. The management server receives the request and uses the account configuration code to find the user in the user database.
7. The management server returns a **KeyActivationResponse** success message to the protocol client. The message includes the user's identity, domain, and usage policies.
8. The protocol client receives the response from the management server, creates the user's account and associates the account with the user's identity, domain, and usage policies.
9. The protocol client sends a **CreateAccount** request, as described in [MS-GRVSPCM], to the management server to register the user's account with the account GUID.
10. The management server receives the account registration request and registers the user's account in the database.
11. The management server returns a **CreateAccountResponse** success message to the protocol client.
12. The protocol client sends a **ManagedObjectInstall** message, as described in [MS-GRVSPCM], to the management server to indicate that the user's identity and usage policies were installed on the protocol client.
13. The management server returns a **ManagedObjectInstallResponse** success message to the protocol client.
14. The protocol client sends a **DomainEnrollment** request, as described in [MS-GRVSPCM], to the management server to activate the user in the domain.
15. The management server receives the request and activates the user in the domain. It also signs the data for the user's identity with the domain certificate.
16. The management server returns a **DomainEnrollmentResponse** success message to the protocol client. The message includes the signed data for the user's identity.
17. The protocol client receives the response and updates the user's identity with the signed data.

18. The protocol client sends a **Publish** message, as described in [MS-GRVWDPP], to the relay server that is associated with the account. This message changes the status of the account to online.

19. The protocol client notifies the user that the account was created successfully and is ready to use.

If an error occurs during the account creation process, the user’s account is not created and the protocol client notifies the user of the error. The user can then start the account creation process again.

If the protocol client cannot connect to the management server, the user’s account cannot be created and the protocol client displays an error message to the user. It is possible that the user entered an incorrect URL for the management server.

The following diagram illustrates the message sequence for this example.

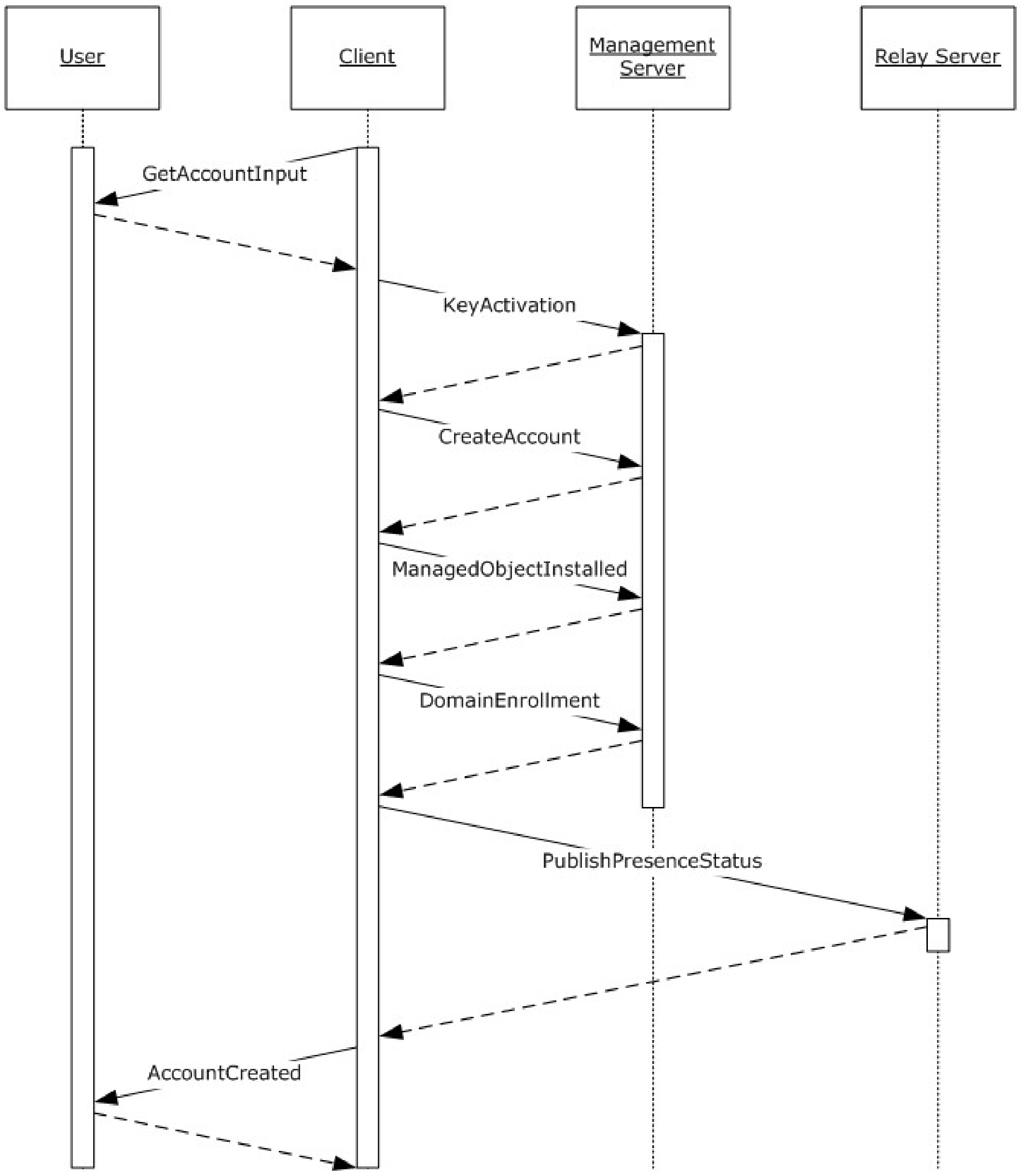

Figure 24: Message sequence for creating an account manually

The following diagram illustrates the state of system components during the account creation process.

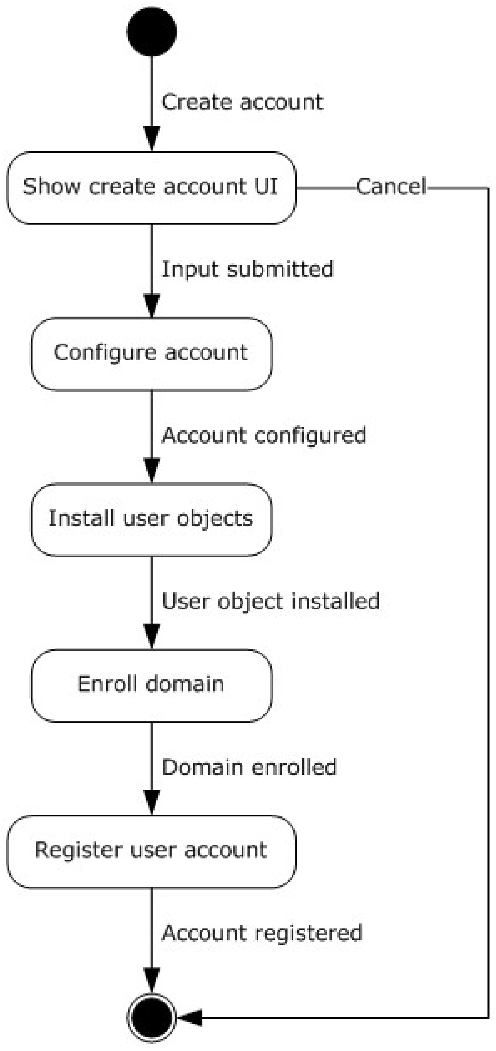

Figure 25: Component states during the account creation process

3.2 Example 2: Add Contacts

This example describes how a user adds other users to a contact list. It corresponds to the following use cases:

- "Publish Presence Status," as described in section 2.5.2
- "Display Presence Information," as described in section 2.5.4
- "Search for Users," as described in section 2.5.5,
- "Add Contacts," as described in section 2.5.6

In this example, a user searches for other users and then adds those users to a contact list in the protocol client. During the search for users to add, the protocol client communicates with the management server by using the Client to Management Server Groove SOAP Protocol, as described in [MS-GRVSPCM]. To get and display presence (1) information for users who are listed in the search results, the protocol client communicates with the relay server by using the Wide Area Network Device Presence Protocol (WAN DPP), as described in [MS-GRVWDPP].

The following steps describe the process for searching for and adding users to a contact list, and assume that no errors occur during the process:

1. In the protocol client UI, the user selects the option to add contacts.
2. The protocol client displays the add contacts UI and waits for user input.
3. The user selects the option to search for contacts.
4. The protocol client displays the search UI and waits for user input.
5. The user enters and submits search criteria.
6. The protocol client sends a **ContactSearch** request to the management server by using the Client to Management Server Groove SOAP Protocol, as described in [MS-GRVSPCM].
7. The management server receives the request and searches for users that meet the specified search criteria.
8. The management server returns a **ContactSearchResponse** success message to the protocol client.
9. The protocol client receives the search results.
10. The protocol client sends **Subscribe** requests, as described in [MS-GRVWDPP], to the relay servers for all of the users in the search results and subscribes to presence (1) information for all of those users.
11. The relay servers receive the requests and update the corresponding presence (1) subscription records.
12. The relay servers send **Notify** messages, as described in [MS-GRVWDPP], to notify the protocol client of the presence (1) information for the users.
13. The protocol client receives and displays presence (1) information for the users.
14. The user selects users from the search results and adds them to the contact list.
15. The user closes the add contacts UI.
16. The protocol client closes the add contacts UI.

The following diagram illustrates the message sequence for this example.

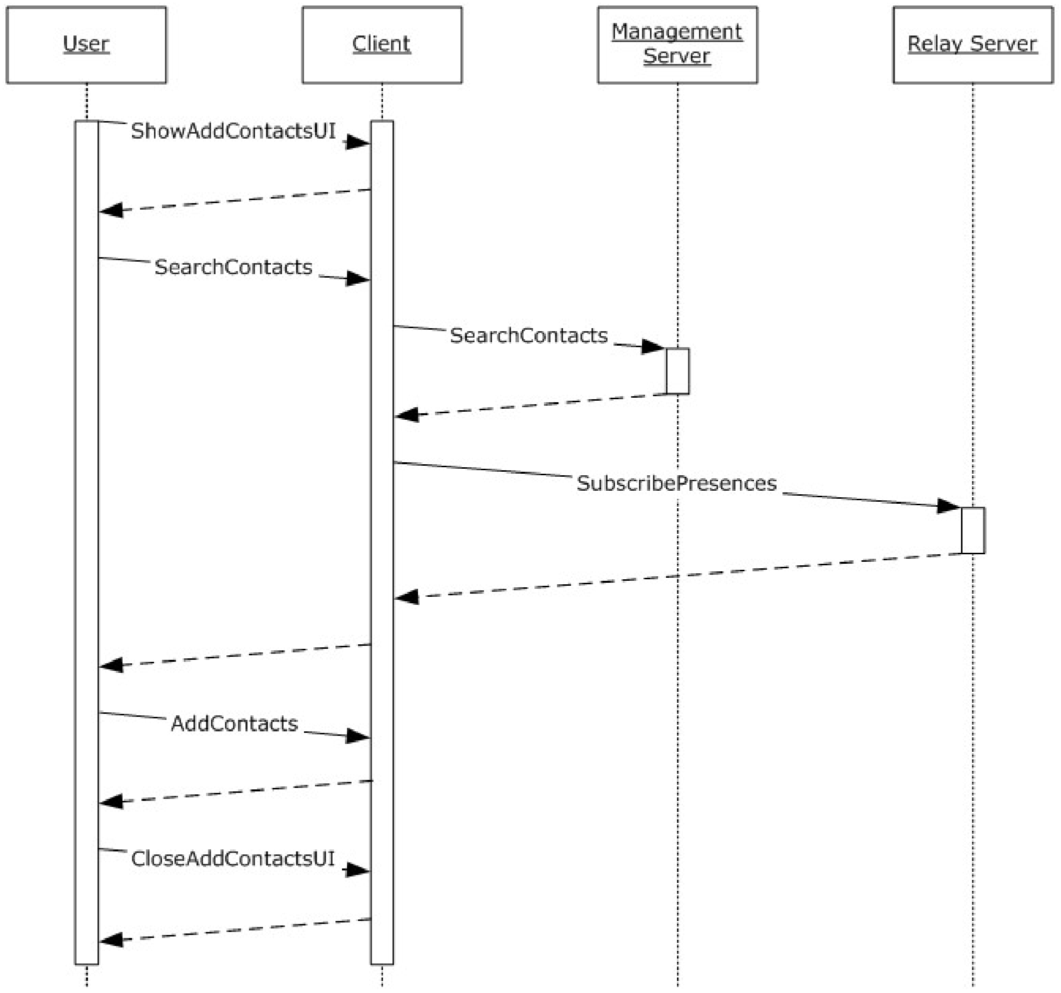

Figure 26: Message sequence for adding contacts

The following diagram illustrates the state of system components during the process of adding contacts.

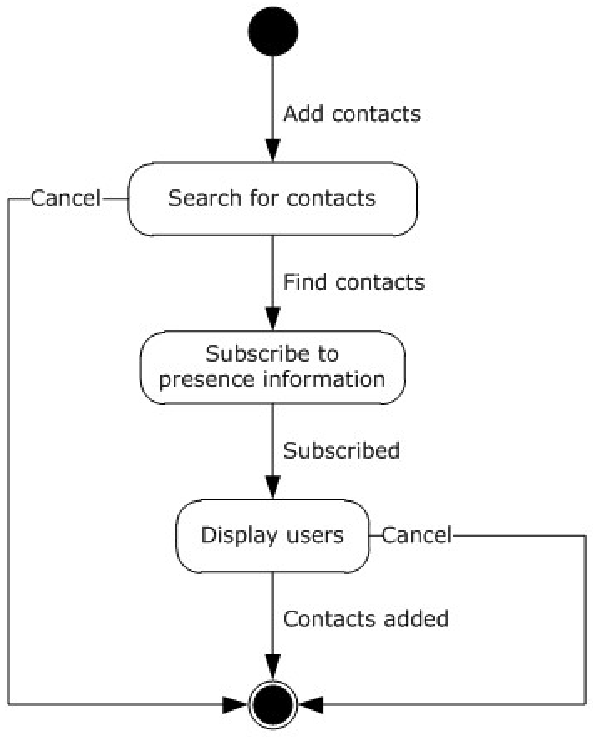

Figure 27: Component states during the add contacts process

3.3 Example 3: Update a Shared Space

This example describes the process of updating and synchronizing a shared space. It corresponds to the "Update a Shared Space" use case that is described in section 2.5.8.

In this example, a user updates a shared space by using a protocol client and the update is forwarded to a relay server. Members of the shared space then receive the update when their protocol clients contact the relay server.

When a shared space is updated, the protocol client uses the Groove RDB Commands Protocol, as described in [MS-GRVRDB], to create update commands for the shared space. The protocol client uses the Groove Dynamics Protocol, as described in [MS-GRVDYNM] to define the sequence for executing those commands. The commands are then forwarded to a relay server by using the Simple Symmetric Transport (SSTP) Protocol, as described in [MS-GRVSSTP], and the SSTP Security Protocol, as described in [MS-GRVSSTPS]. If the protocol client cannot use SSTP to communicate with the relay server because of a firewall or proxy server, the HTTP Encapsulation of SSTP Protocol, as described in [MS-GRVHENC], is used. The Groove Dynamics Protocol messages are encrypted by the protocol client and are opaque to the relay server. When the relay server receives the messages, it queues them to be forwarded to the destination protocol clients when those clients contact the relay server.

The following steps describe the process for updating and synchronizing a shared space, and assume that no errors occur during the process:

1. The user updates content, such as documents or tools, in a shared space.

2. The protocol client uses the Groove RDB Commands Protocol, as described in [MS-GRVRDB], to create update commands for the shared space database records. The update commands are also sequenced by using the Groove Dynamics Protocol, as described in [MS-GRVDYNM].
3. The protocol client sends the sequenced set of update commands to the relay server by using SSTP, as described in [MS-GRVSSTP], and the SSTP Security Protocol, as described in [MS-GRVSSTPS]. If the protocol client cannot use SSTP to communicate with the relay server because of a firewall or proxy server, the HTTP Encapsulation of SSTP Protocol, as described in [MS-GRVHENC], is used.
4. The relay server receives the update commands and queues them to be forwarded to protocol clients of members of the shared space, when those clients contact the relay server.
5. A protocol client for a member of a shared space opens a connection to the relay server for relay services.
6. The relay server both services the protocol client and forwards the update commands for the shared space to the protocol client.
7. The protocol client receives the set of update commands and executes them in the specified order. If necessary, the protocol client also backs up previous changes to a point where the update commands can be rearranged in the correct order of execution to synchronize the shared space. The protocol client also acknowledges the sender.
8. The protocol client notifies the member that the shared space has been updated.

The following diagram illustrates the message sequence for this example.

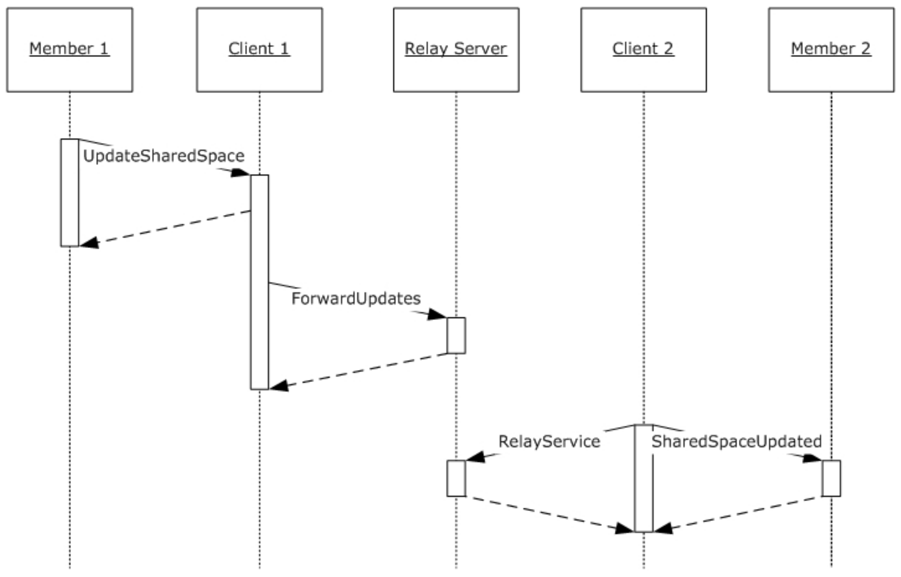

Figure 28: Message sequence for updating a shared space

The following diagram illustrates the state of system components during the process of updating a shared space.

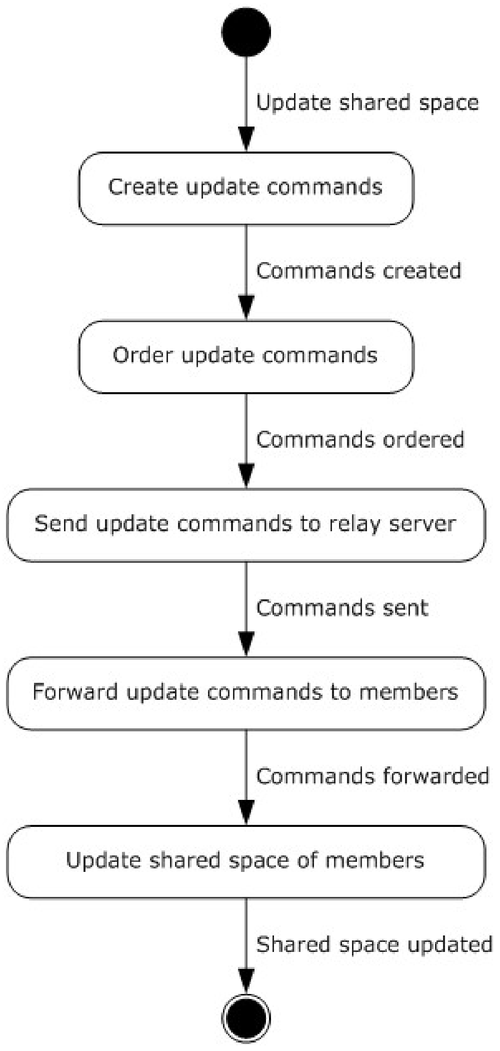

Figure 29: Component states during the update process for a shared space

3.4 Example 4: Add a Policy to a Domain

This example describes how a policy is added to a domain on a management server. It corresponds to the "Add a Policy to a Domain" use case that is described in section 2.5.11. This example also describes how a protocol client obtains the additional policy.

In this example, an administrator adds a new policy to a domain on a management server. A protocol client receives the additional policy when it checks for updates with the management server.

The protocol client communicates with the management server by using the Client to Management Server Groove SOAP Protocol, as described in [MS-GRVSPCM].

The following steps describe the process for adding a policy to a domain, and assume that no errors occur during the process:

1. The administrator starts the administration application on the management server.

2. The administration application displays the administration UI for the management server.

3. The administrator selects the domain to add the policy to.

4. The administration application displays the domain management UI.

5. The administrator selects the policy management option.

6. The administration application displays the policy management UI

7. The administrator selects the option to add a policy.

8. The administration application displays the UI for adding a policy.

9. The administrator enters and submits information about the policy.

10. The administration application validates the information. If any of the information is not valid, the application notifies the administrator and waits for the correct information.

11. The administration application adds the policy to the domain and displays it to the administrator.

12. A protocol client sends a **ManagedObjectStatus** request message, as described in [MS-GRVSPCM], to the management server to check for updates.

13. The management server receives the request and responds with the policy that was added.

14. The protocol client installs the additional policy.

The following diagram illustrates the message sequence for this example.

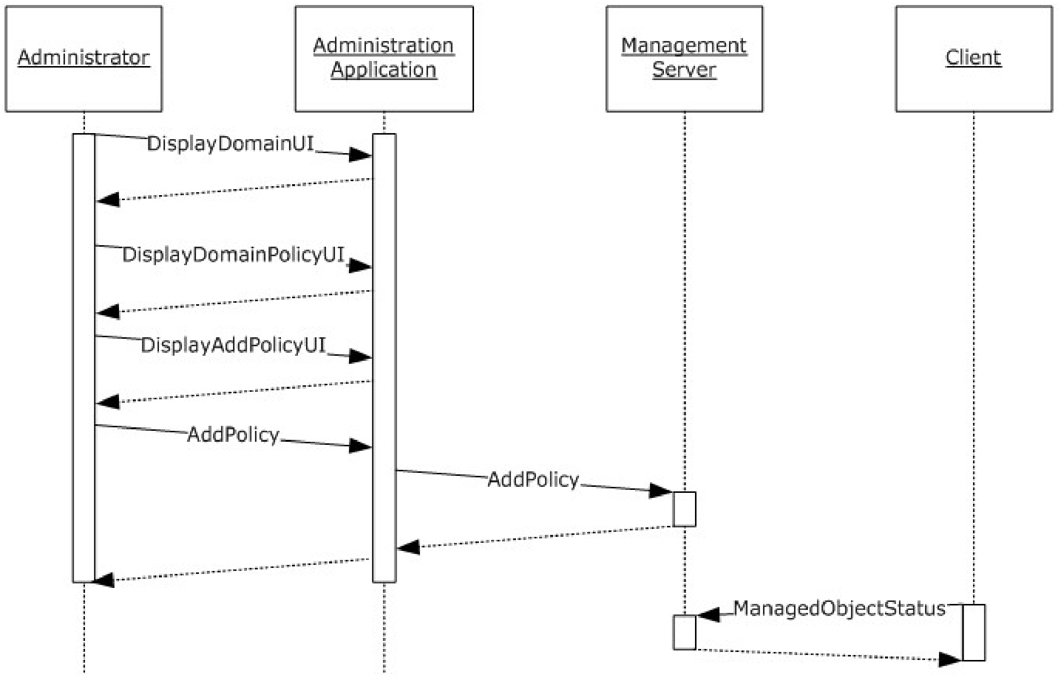

Figure 30: Message sequence for adding a policy

The following diagram illustrates the state of system components during the process of adding a policy to a domain.

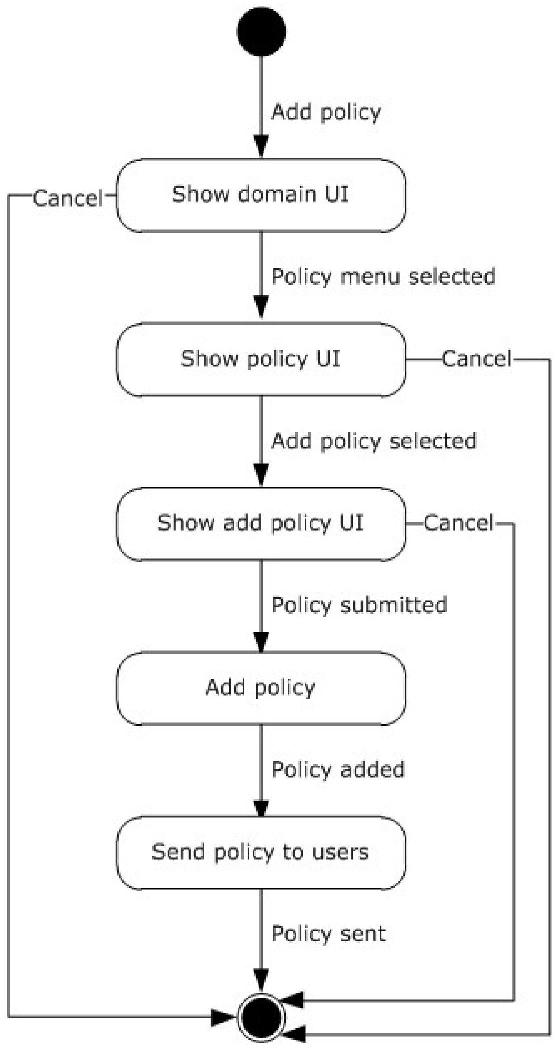

Figure 31: Component states during the process of adding a policy to a domain

3.5 Example 5: Add Users to a Relay Server

This example describes how users are added to a relay server. It corresponds to the "Add Users to a Relay Server" use case that is described in section 2.5.14.

In this example, an administrator adds users to a management server and the management server manages a relay server. The management server communicates with the relay server by using the Management Server to Relay Server Groove SOAP Protocol, as described in [MS-GRVSPMR].

The following steps describe the process for adding a user to a relay server, and assume that no errors occur during the process:

1. The administrator adds users to a domain on the management server, as described in section 2.5.10, by using the administration application.

2. The management server sends a **userAdd** request, as described in [MS-GRVSPMR], to the relay server. If all of the users cannot be added in one request, the management server sends additional **userAdd** requests to the relay server.

3. The relay server receives the request and processes it.

If the user database for the relay server exists and does not need to be rebuilt, the relay server adds the users to the existing user database and returns a response to the management server indicating that the users were added successfully.

If the user database for the relay server needs to be built or rebuilt:

1. The relay server responds to the management server indicating that the user database needs to be built.

2. The management server receives the response.

3. The management server sends a **RelayQuiescent** request with inactive status to the relay server by using the Management Server to Relay Server Groove SOAP Protocol, as described in [MS-GRVSPMR], to put the relay server in an inactive state before building the user database.

4. The relay server receives the request, enters the inactive state, and returns a successful response.

5. The management server receives the successful response and sends **userAdd** requests to the relay server by using the Management Server to Relay Server Groove SOAP Protocol, as described in [MS-GRVSPMR], to build the user database.

6. The relay server receives the requests, adds the users to the user database, builds or rebuilds the user database, and returns successful responses.

7. The management server sends a **RelayQuiescent** request with active status to the relay server by using the Management Server to Relay Server Groove SOAP Protocol, as described in [MS-GRVSPMR], to put the relay server in an active state.

8. The relay server receives the request, enters the active state, and returns a successful response.

The following diagram illustrates the message sequence for adding users to a relay server if the user database needs to be built or rebuilt.

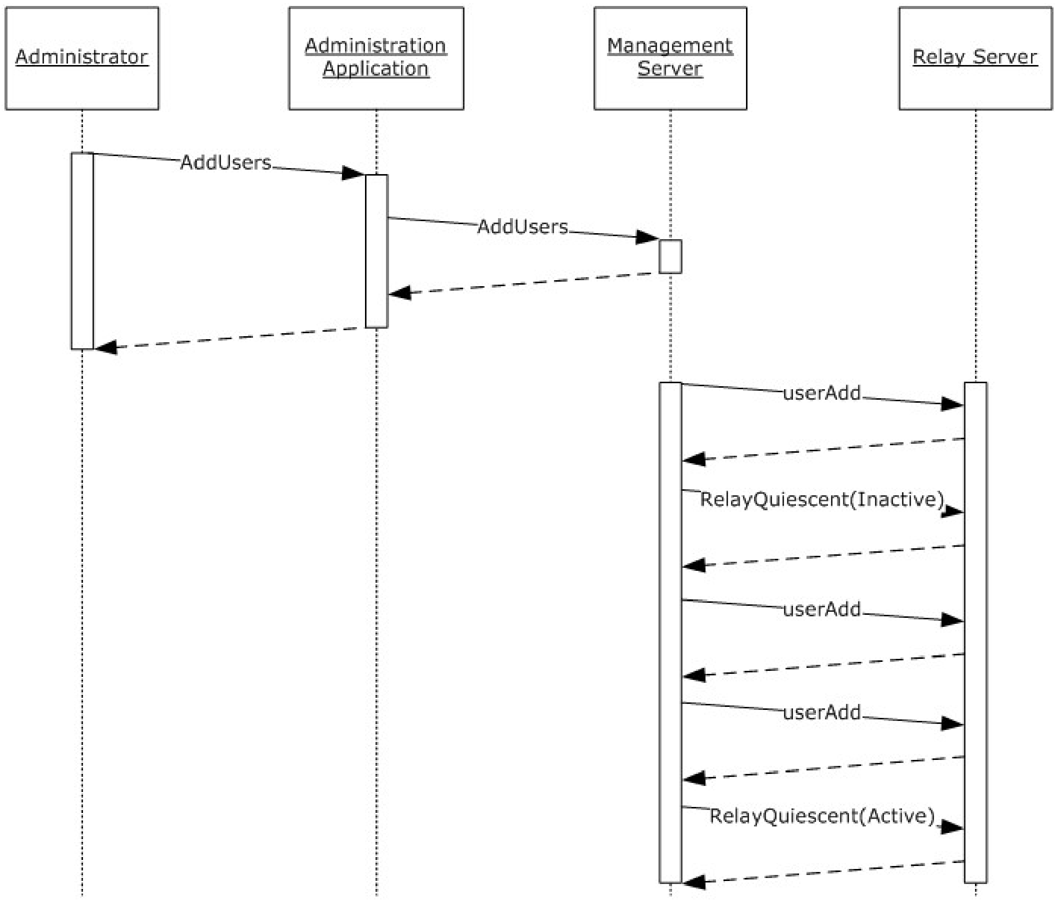

Figure 32: Message sequence for building a user database on and adding users to a relay server

The following diagram illustrates the message sequence for adding users to a relay server if the user database does not need to be built or rebuilt.

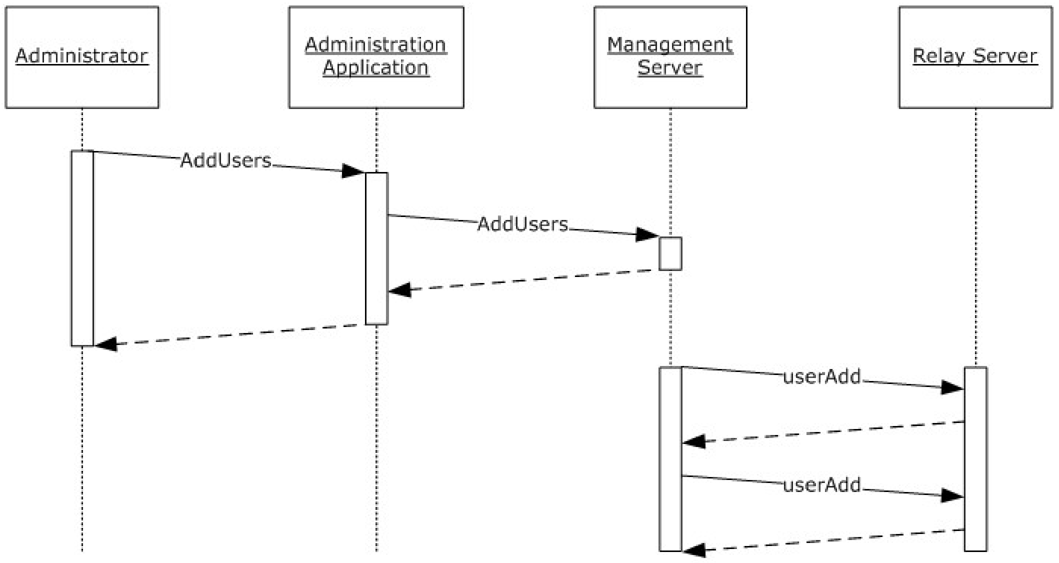

Figure 33: Message sequence for adding users to a relay server

The following diagram illustrates the state of system components during the process of adding users to a relay server if the user database does not need to be built or rebuilt.

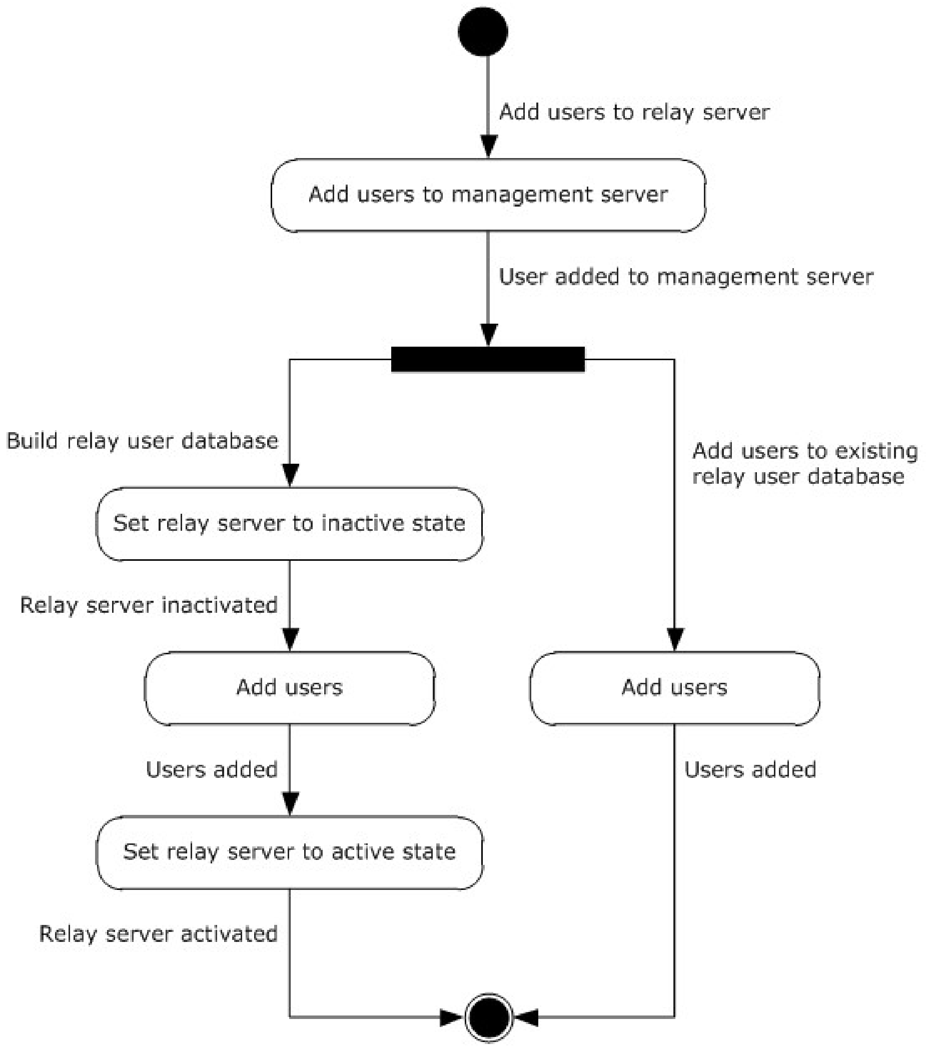
Add users to relay server
Add users to management server
User added to management server
Build relay user database
Add users to existing relay user database
Set relay server to inactive state
Relay server inactivated
Add users
Add users
Users added
Users added
Set relay server to active state
Relay server activated

Figure 34: Component states during the process of adding users to a relay server

4 Microsoft Implementations

There are no variations in the behavior of the Microsoft® Groove® and Microsoft SharePoint® Workspace system in different versions of Groove and SharePoint Workspace beyond those described in the specifications of the protocols supported by the system, as listed in section 2.2.

The information in this document is applicable to the following versions of Groove Server, Groove, Microsoft Office, and SharePoint Workspace:

- Microsoft® Office Groove® 2007
- Microsoft® Office Groove® Server 2007
- Microsoft® Groove® Server 2010
- Microsoft® Office 2010 suites
- Microsoft® SharePoint® Workspace 2010

The following table identifies the relationships between the abridged component names that are used in this document and the specific product and component names within the Microsoft implementation of the system.

Abridged name	Microsoft product or component name
management server	Groove Server 2010 Office Groove Server 2007
relay server	Groove Server 2010 Office Groove Server 2007
data bridge server	Office Groove Server 2007
protocol client	SharePoint Workspace 2010 Office Groove 2007

The Microsoft implementation of the Groove and SharePoint Workspace system has the following dependencies:

- The management server requires the Windows Server® 2003 operating system or the Windows Server® 2008 operating system and Microsoft® SQL Server® 2005 or Microsoft® SQL Server® 2008. In addition, the management server supports, but does not require, integration with Active Directory® Domain Services (AD DS) for user management.
- The protocol client requires Windows Server 2003, Windows Server 2008, the Windows® XP operating system, the Windows Vista® operating system, or the Windows® 7 operating system.
- Microsoft® SharePoint® Workspace 2010 integrates with Microsoft® SharePoint® Server 2010. It acts as a Microsoft Office client that communicates with Microsoft® SharePoint® Server. For more information, see [MS-OCPROTO].

Exceptions, if any, are noted in the following section.

4.1 Product Behavior

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

5 Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

6 Index