Directory Integration and Identity Management

Revised: March 1, 2018

Identity management is a fundamental concept required in any application. Identity management involves the management of individual principals and the authentication and authorization of these principals. Traditionally each application handled identity management individually. This led to the situation that users had to authenticate against every individual application. Centralizing identity management, authentication, and authorization helps greatly to improve the user experience by providing services such as single sign-on (SSO).

The first step of centralizing identity management is to centralize storage of information about principals in an enterprise. These centralized enterprise-wide datastores are commonly known as directories.

Directories are specialized databases that are optimized for a high number of reads and searches, and occasional writes and updates. Directories typically store data that does not change often, such as employee information, user privileges, and group membership on the corporate network.

Directories are extensible, meaning that the type of information stored can be modified and extended. The term *directory schema* defines the type of information stored, its container (or attribute), and its relationship to users and resources.

The Lightweight Directory Access Protocol (LDAP) provides applications with a standard method for accessing and potentially modifying the information stored in the directory. This capability enables companies to centralize all user information in a single repository available to several applications, with a remarkable reduction in maintenance costs through the ease of adds, moves, and changes.

This chapter covers the main design principles for integrating a Cisco Unified Communications system based on Cisco Unified Communications Manager (Unified CM) with a corporate LDAP directory. The main topics include:

- What is Directory Integration?, page 16-3

  This section analyzes the various requirements for integration with a corporate LDAP directory in a typical enterprise IT organization.

- Directory Access for Unified Communications Endpoints, page 16-4

  This section describes the technical solution to enable directory access for Cisco Unified Communications endpoints and provides design best-practices around it.

- Directory Integration with Unified CM, page 16-7

  This section describes the technical solutions and provides design considerations for directory integration with Cisco Unified CM, including the LDAP synchronization and LDAP authentication functions.

- Directory Integration for VCS Registered Endpoints, page 16-33

  This section briefly introduces the technical solution to enable directory access for video endpoints registered to the Cisco TelePresence Video Communication Server (VCS).
- Identity Management Architecture Overview, page 16-33

  This section describes the identity management architecture.
- Single Sign-On (SSO), page 16-35

  This section provides an overview of SAML 2.0 single sign-on (SSO).
- Authorization Framework, page 16-45

  This section describes the OAuth authorization service available in Cisco Unified CM.

The considerations presented in this chapter apply to Cisco Unified CM as well as the following applications bundled with it: Cisco Extension Mobility, Cisco Unified Communications Manager Assistant, WebDialer, Bulk Administration Tool, and Real-Time Monitoring Tool.

For Cisco Unity, refer to the *Cisco Unity Design Guide* and to the following white papers: *Cisco Unity Data and the Directory*, *Active Directory Capacity Planning*, and *Cisco Unity Data Architecture and How Cisco Unity Works*, also available at

https://www.cisco.com

What's New in This Chapter

Table 16-1 lists the topics that are new in this chapter or that have changed significantly from previous releases of this document.

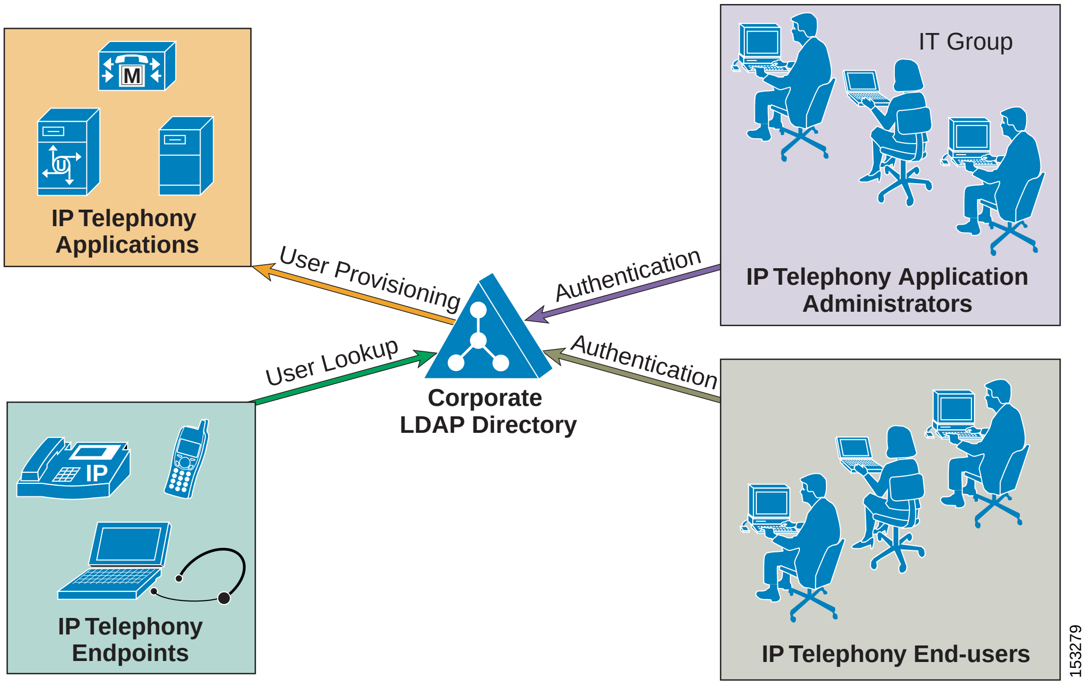
Table 16-1 New or Changed Information Since the Previous Release of This Document

New or Revised Topic	Described in	Revision Date
Directory access using User Data Service (UDS)	Directory Access for Unified Communications Endpoints Using Cisco User Data Service (UDS), page 16-6	March 1, 2018
Identity management	Identity Management Architecture Overview, page 16-33	March 1, 2018
Single Sign-On (SSO)	SSO for Cisco Jabber, page 16-43 Design Considerations for SSO, page 16-44	March 1, 2018
Authentication and OAuth 2.0	Authorization Framework, page 16-45	March 1, 2018

What is Directory Integration?

Integrating voice applications with a corporate LDAP directory is a common task for many enterprise IT organizations. However, the exact scope of the integration varies from company to company, and can translate into one or more specific and independent requirements, as shown in Figure 16-1.

Figure 16-1 Various Requirements for Directory Integration



One common requirement is to enable user lookups (sometimes called the "white pages" service) from IP phones or other voice and/or video endpoints, so that users can dial contacts quickly after looking up their numbers in the directory.

Another requirement is to provision users automatically from the corporate directory into the user database for applications. This method avoids having to add, remove, or modify core user information manually each time a change occurs in the corporate directory.

Authentication of end users and administrators of the voice and/or video applications using their corporate directory credentials is also a common requirement. Enabling directory authentication allows the IT department to deliver single log-on functionality while reducing the number of passwords each user needs to maintain across different corporate applications.

As shown in Table 16-2, within the context of a Cisco Unified Communications system, the term *directory access* refers to mechanisms and solutions that satisfy the requirement of user lookups for Cisco Unified Communications endpoints, while the term *directory integration* refers to mechanisms and solutions that satisfy the requirements of user provisioning and authentication (for both end users and administrators).

Table 16-2 Directory Requirements and Cisco Solutions

Requirement	Cisco Solution	Cisco Unified CM Feature
User lookup for endpoints	Directory access	Cisco Unified IP Phone Services SDK Cisco User Data Service (UDS)
User provisioning	Directory integration	LDAP Synchronization
Authentication for Unified Communications end users	Directory integration	LDAP Authentication
Authentication for Unified Communications application administrators	Directory integration	LDAP Authentication

The remainder of this chapter describes how to address these requirements in a Cisco Unified Communications system based on Cisco Unified CM.

Note Another interpretation of the term *directory integration* revolves around the ability to add application servers to a Microsoft Active Directory domain in order to centralize management and security policies. Cisco Unified CM is an appliance that runs on a customized embedded operating system, and it cannot be added to a Microsoft Active Directory domain. Server management for Unified CM is provided through the Cisco Real Time Monitoring Tool (RTMT). Strong security policies tailored to the application are already implemented within the embedded operating system.

Directory Access for Unified Communications Endpoints

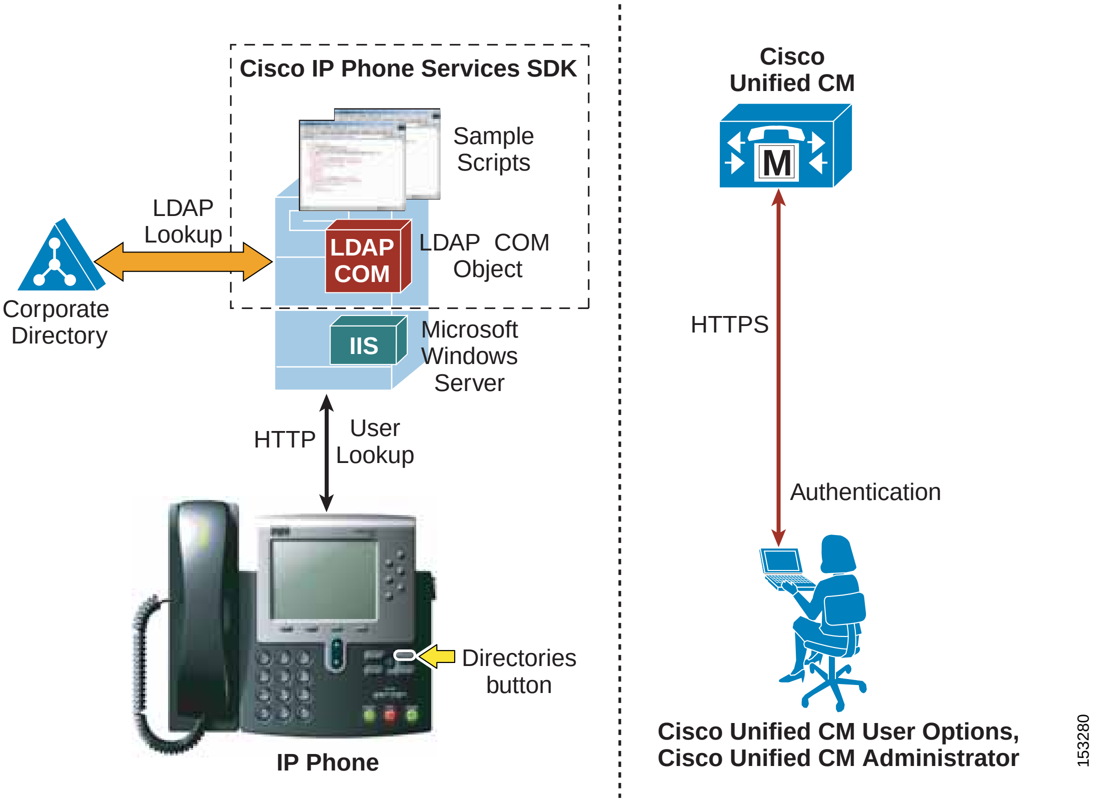
This section describes how to configure corporate directory access to any LDAP-compliant directory server to perform user lookups from Cisco Unified Communications endpoints (such as Cisco Unified IP Phones). The guidelines contained in this section apply regardless of whether Unified CM or other Unified Communications applications have been integrated with a corporate directory for user provisioning and authentication.

Cisco Unified IP Phones equipped with a display screen can search a user directory when a user presses the Directories button on the phone. The IP Phones use Hyper-Text Transfer Protocol (HTTP) to send requests to a web server. The responses from the web server contain specific Extensible Markup Language (XML) objects that the phone interprets and displays.

By default, Cisco Unified IP Phones are configured to perform user lookups against Unified CM's embedded database. However, it is possible to change this configuration so that the lookup is performed on a corporate LDAP directory. In this case, the phones send an HTTP request to an external web server that operates as a proxy by translating the request into an LDAP query which is then processed by the corporate directory. The web server encapsulates the LDAP response into an XML object that is sent back to the phone using HTTP, to be rendered to the end user.

Figure 16-2 illustrates this mechanism in a deployment where Unified CM has not been integrated with the corporate directory. Note that, in this scenario, Unified CM is not involved in the message exchange. The authentication mechanism to Unified CM web pages, shown on the right half of Figure 16-2, is independent of how directory lookup is configured.

Figure 16-2 Directory Access for Cisco Unified IP Phones Using the Cisco Unified IP Phone Services SDK



In the example shown in Figure 16-2, the web server proxy function is provided by the Cisco LDAP Search Component Object Model (COM) server, which is included in the Cisco Unified IP Phone Services Software Development Kit (SDK). You can download the latest Cisco Unified IP Phone Services SDK from Cisco DevNet, the Cisco developer community, at

https://developer.cisco.com/site/devnet/home/index.gsp

The IP Phone Services SDK can be installed on a Microsoft Windows web server running IIS 4.0 or later, but it cannot be installed on a Unified CM server. The SDK includes some sample scripts to provide simple directory lookup functionality.

To set up a corporate directory lookup service using the IP Phone Services SDK, perform the following steps:

Step 1 Modify one of the sample scripts to point to your corporate LDAP directory, or write your own script using the LDAP Search COM Programming Guide provided with the SDK.

Step 2 In Unified CM, configure the URL Directories parameter (under **System** > **Enterprise Parameters**) to point to the URL of the script on the external web server.

Step 3 Reset the phones to make the changes take effect.

Note If you want to offer the service only to a subset of users, configure the URL Directories parameter directly within the Phone Configuration page instead of the Enterprise Parameters page.

In conclusion, the following design considerations apply to directory access with the Cisco Unified IP Phone Services SDK:

- User lookups are supported against any LDAP-compliant corporate directory.
- When querying Microsoft Active Directory, you can perform lookups against the Global Catalog by pointing the script to a Global Catalog server and specifying port 3268 in the script configuration. This method typically results in faster lookups. Note that a Global Catalog does not contain a complete set of attributes for users. Refer to Microsoft Active Directory documentation for details.
- There is no impact on Unified CM when this functionality is enabled, and only minimal impact on the LDAP directory server.
- The sample scripts provided with the SDK allow only a minimal amount of customization (for example, you can prefix a digit string to all returned numbers). For a higher degree of manipulation, you will have to develop custom scripts, and a programming guide is included with the SDK to aid in writing the scripts.
- This functionality does not entail provisioning or authentication of Unified CM users with the corporate directory.

Directory Access for Unified Communications Endpoints Using Cisco User Data Service (UDS)

This section describes the mechanisms and best practices for directory access for endpoints using UDS to access user data instead of using the web service described in the previous section. The User Data Service (UDS) API is a REST-based set of operations that provide authenticated access to user resources and entities such as user devices, subscribed services, speed dials, and much more from the Unified Communications configuration database.

Current endpoints, including all endpoints running CE software, directly access the UDS REST-based directory search methods to obtain search results whenever a user invokes the search function on the endpoint. The results returned by the UDS search method are then displayed on the endpoint display. The UDS search function lists only those directory entries that exist in the Unified CM database, unless the UDS LDAP proxy functionality is used. When using the UDS LDAP proxy functionality, the endpoints still request directory information from Cisco Unified CM via UDS, but the results will then be requested by the UDS service from the configured external LDAP directory before being returned to the endpoint as a result of the UDS request.

Directory Integration with Unified CM

This section describes the mechanisms and best practices for directory integration with Cisco Unified CM to allow for user provisioning and authentication with a corporate LDAP directory. This section covers the following topics:

- Cisco Unified Communications Directory Architecture, page 16-7

  This section provides an overview of the user-related architecture in Unified CM.

- LDAP Synchronization, page 16-10

  This section describes the functionality of LDAP synchronization and provides design guidelines for its deployment, with additional considerations for Microsoft Active Directory.

- LDAP Authentication, page 16-22

  This section describes the functionality of LDAP authentication and provides design guidelines for its deployment, with additional considerations for Microsoft Active Directory.

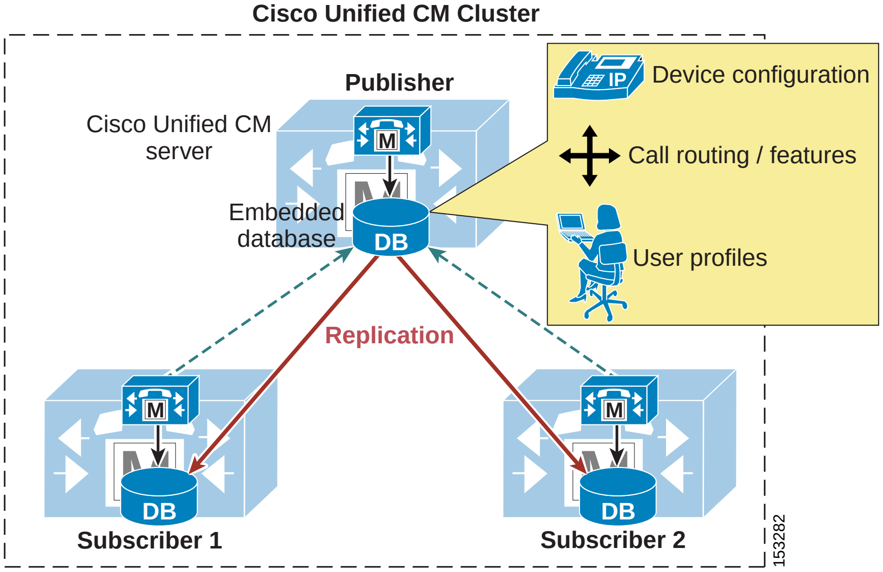
For a list of supported LDAP directories, refer to the latest version of the *System Configuration Guide for Cisco Unified Communications Manager*, available at

https://www.cisco.com/c/en/us/support/unified-communications/unified-communications-manager-callmanager/products-installation-and-configuration-guides-list.html

Cisco Unified Communications Directory Architecture

Figure 16-3 shows the basic architecture of a Unified CM cluster. The embedded database stores all configuration information, including device-related data, call routing, feature provisioning, and user profiles. The database is present on all servers within a Unified CM cluster and is replicated automatically from the publisher server to all subscriber servers.

Figure 16-3 Cisco Unified CM Architecture



By default, all users are provisioned manually in the publisher database through the Unified CM Administration web interface. Cisco Unified CM has two types of users:

- End users — All users associated with a physical person and an interactive login. This category includes all Unified Communications users as well as Unified CM administrators when using the User Groups and Roles configuration (equivalent to the Cisco Multilevel Administration feature in prior Unified CM versions).
- Application users — All users associated with other Cisco Unified Communications features or applications, such as Cisco Attendant Console, Cisco Unified Contact Center Express, or Cisco Unified Communications Manager Assistant. These applications need to authenticate with Unified CM, but these internal "users" do not have an interactive login and serve purely for internal communications between applications.

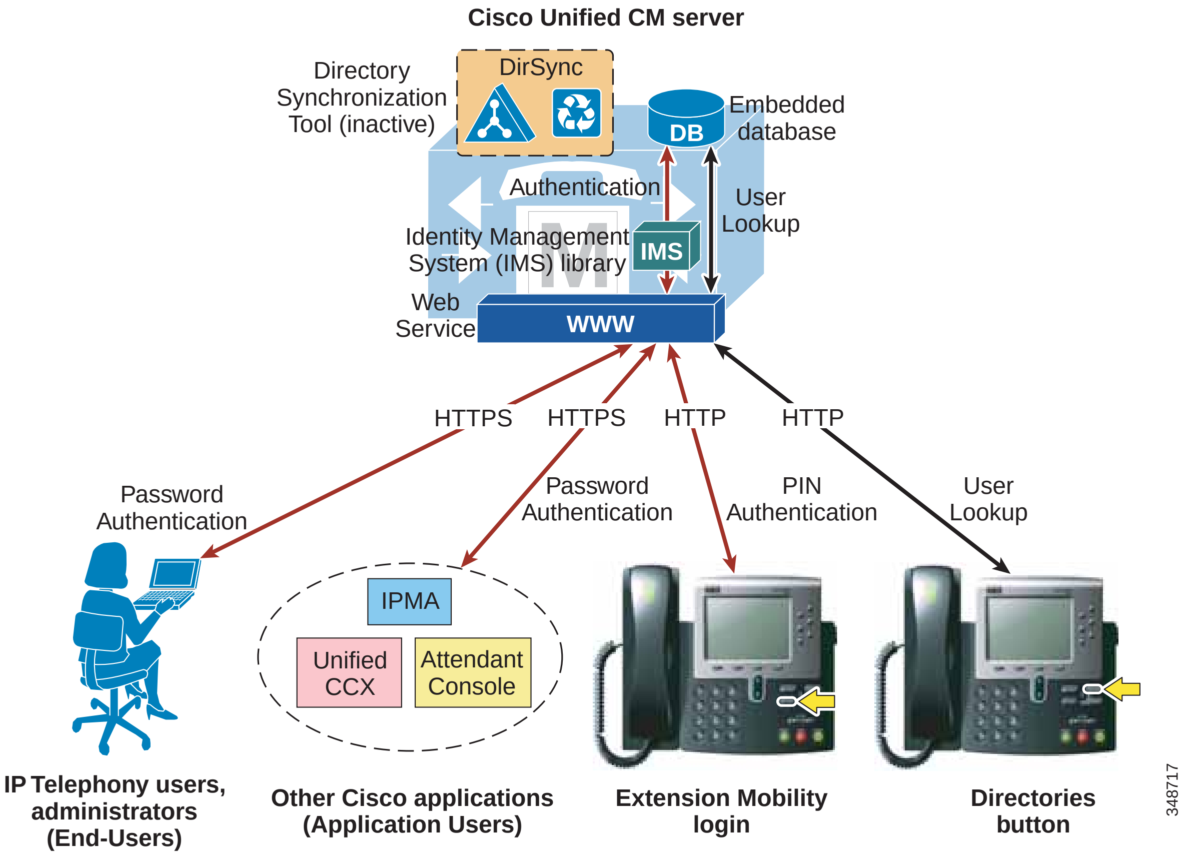
Table 16-3 lists the application users created by default in the Unified CM database, together with the feature or application that uses them. Additional application users can be created manually when integrating other Cisco Unified Communications applications (for example, the **ac** application user for Cisco Attendant Console, the **jtapi** application user for Cisco Unified Contact Center Express, and so forth).

Table 16-3 Default Application Users for Unified CM

Application User	Used by:
CCMAdministrator	Unified CM Administration (default "super user")
CCMQRTSecureSysUser	Cisco Quality Reporting Tool
CCMQRTSysUser	
CCMSysUser	Cisco Extension Mobility
IPMASecureSysUser	Cisco Unified Communications Manager Assistant
IPMASysUser	
WDSecureSysUser	Cisco WebDialer
WDSysUser	

Based on these considerations, Figure 16-4 illustrates the default behavior in Unified CM for user-related operations such as lookups, provisioning, and authentication.

Figure 16-4 Default Behavior for User-Related Operations for Unified CM



End users access the Unified CM User Options page via HTTPS and authenticate with a user name and password. If they have been configured as administrators by means of User Groups and Roles, they can also access the Unified CM Administration pages with the same credentials.

Similarly, other Cisco features and applications authenticate to Unified CM via HTTPS with the user name and password associated with their respective application users.

The authentication challenge carried by the HTTPS messages are relayed by the web service on Unified CM to an internal library called Identity Management System (IMS). In its default configuration, the IMS library authenticates both end users and application users against the embedded database. In this way, both "physical" users of the Unified Communications system and internal application accounts are authenticated using the credentials configured in Unified CM.

End users may also authenticate with their user name and a numeric password (or PIN) when logging into the Extension Mobility service from an IP phone. In this case, the authentication challenge is carried via HTTP to Unified CM but is still relayed by the web service to the IMS library, which authenticates the credentials against the embedded database.

In addition, user lookups performed by Unified Communications endpoints via the Directories button communicate with the web service on Unified CM via HTTP and access data on the embedded database.

The importance of the distinction between End Users and Application Users becomes apparent when integration with a corporate directory is required. As mentioned in the previous section, this integration is accomplished by means of the following two separate processes:

- LDAP synchronization

  This process uses an internal tool called Cisco Directory Synchronization (DirSync) on Unified CM to synchronize a number of user attributes (either manually or periodically) from a corporate LDAP directory. When this feature is enabled, users are automatically provisioned from the corporate directory in addition to local user provisioning through the Unified CM administration GUI. This feature applies only to End Users, while Application Users are kept separate and are still provisioned via the Unified CM Administration interface. In summary, End Users are defined in the corporate directory and synchronized into the Unified CM database, while Application Users are stored only in the Unified CM database and do not need to be defined in the corporate directory.

- LDAP authentication

  This process enables the IMS library to authenticate user credentials of LDAP synchronized End Users against a corporate LDAP directory using the LDAP standard Simple_Bind operation. When this feature is enabled, End User passwords of LDAP synchronized End Users are authenticated against the corporate directory, while Application User passwords and passwords of local End Users are still authenticated locally against the Unified CM database. Cisco Extension Mobility PINs are also still authenticated locally.

Maintaining and authenticating the Application Users internally to the Unified CM database provides resilience for all the applications and features that use these accounts to communicate with Unified CM, independently of the availability of the corporate LDAP directory.

Cisco Extension Mobility PINs are also kept within the Unified CM database because they are an integral part of a real-time application, which should not have dependencies on the responsiveness of the corporate directory.

The next two sections describe in more detail LDAP synchronization and LDAP authentication, and they provide design best-practices for both functions.

Note As illustrated in the section on Directory Access for Unified Communications Endpoints, page 16-4, user lookups from endpoints can also be performed against a corporate directory by configuring the Cisco Unified IP Phone Services SDK on an external web server.

LDAP Synchronization

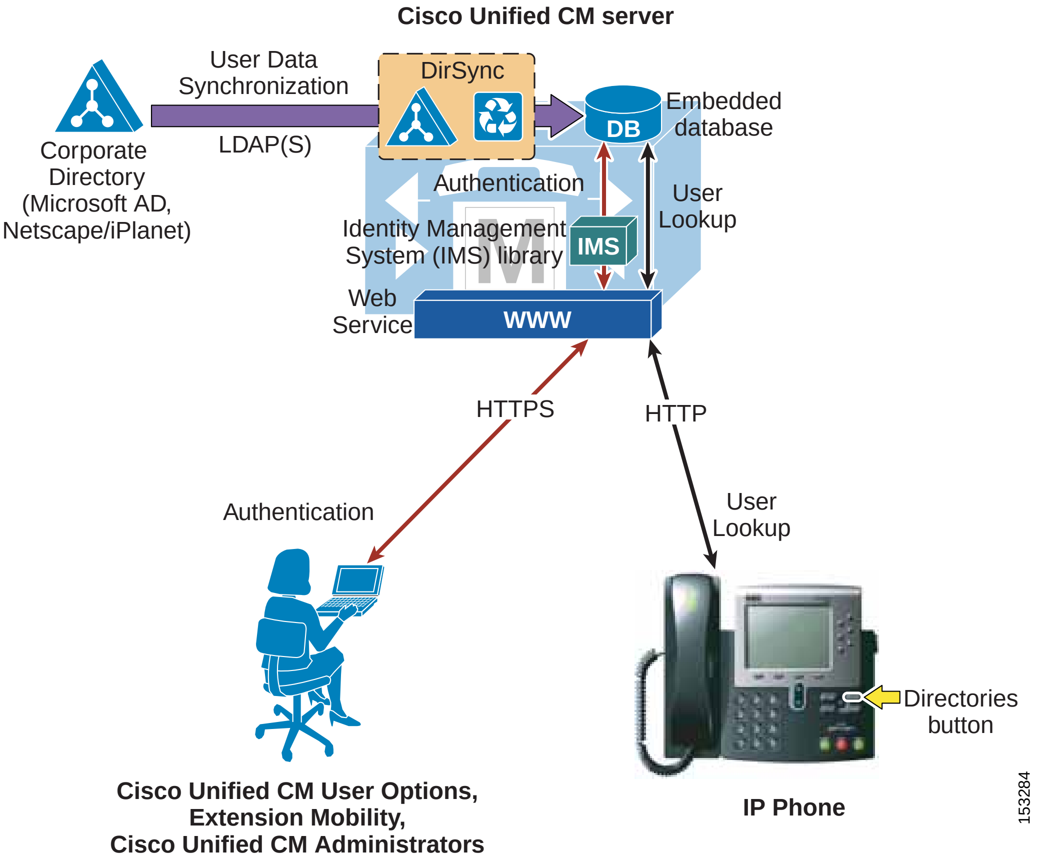
Synchronization of Unified CM with a corporate LDAP directory allows the administrator to provision users easily by mapping Unified CM data fields to directory attributes. Critical user data maintained in the LDAP store is copied into the appropriate corresponding fields in the Unified CM database on a scheduled or on-demand basis. The corporate LDAP directory retains its status as the central repository. Unified CM has an integrated database for storing user data and a web interface within Unified CM Administration for creating and managing user accounts and data. When LDAP synchronization is enabled, the local database is still used, and additional local end-user accounts can be created. Management of end-user accounts is then accomplished through the interface of the LDAP directory and the Unified CM administration GUI. (See Figure 16-5.). Accounts for application users can be created and managed only through the Unified CM Administration web interface.

The user account information is imported from the LDAP directory into the database located on the Unified CM publisher server. Information that is imported from the LDAP directory may not be changed by Unified CM. Additional user information specific to Cisco Unified Communications is managed by

Unified CM and stored only within its local database. For example, device-to-user associations, speed dials, call forward settings, and user PINs are all examples of data that is managed by Unified CM and does not exist in the corporate LDAP directory. The user data is then propagated from the Unified CM publisher server to the subscriber servers through the built-in database synchronization mechanism.

User information synchronized from the LDAP directory can be converted to local user information so that the user information then can be edited locally on Unified CM. Local end users can be added manually using the Unified CM administration GUI. During an LDAP sync, a local end user is converted to an active LDAP user, and if a user with the same user ID is found in LDAP, the locally configured data is replaced with data from the directory.

Figure 16-5 Enabling Synchronization of User Data


Cisco Unified CM server
User Data Synchronization
DirSync
Embedded database
DB
Corporate Directory (Microsoft AD, Netscape/iPlanet)
LDAP(S)
Authentication
User Lookup
Identity Management System (IMS) library
IMS
Web Service
WWW
HTTPS
HTTP
Authentication
User Lookup
Directories button
Cisco Unified CM User Options, Extension Mobility, Cisco Unified CM Administrators
IP Phone
153284

When LDAP synchronization is activated, only one type of LDAP directory may be chosen globally for the cluster at any one time. Also, one attribute of the LDAP directory user is chosen to map into the Unified CM User ID field. Unified CM uses standard LDAPv3 for accessing the data.

Cisco Unified CM imports data from standard attributes. Extending the directory schema is not required. Table 16-4 lists the attributes that are available for mapping to Unified CM fields. The data of the directory attribute that is mapped to the Unified CM User ID must be unique within all entries for that cluster. The attribute mapped to the Unified CM UserID field must be populated in the directory and the **sn** attribute must be populated with data, otherwise those records are skipped during this import action. If the primary attribute used during import of end-user accounts matches any application user in the Unified CM database, that user is not imported from the LDAP directory.

Table 16-4 lists the attributes that are imported from the LDAP directory into corresponding Unified CM user fields, and it describes the mapping between those fields. Some Unified CM user fields might be mapped from one of several LDAP attributes.

Table 16-4 Synchronized LDAP Attributes and Corresponding Unified CM Field Names

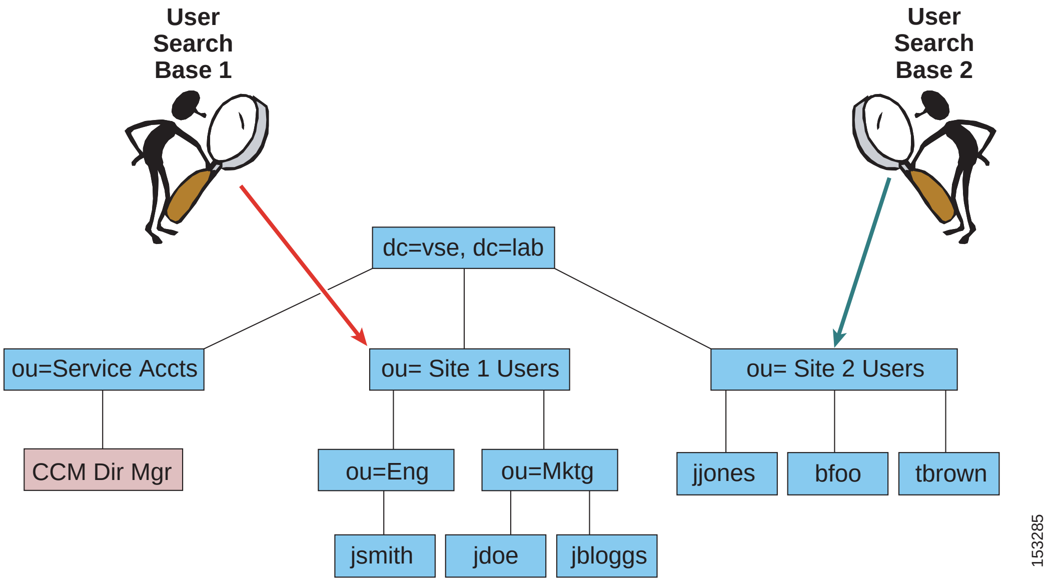
Unified CM User Field	Microsoft Active Directory	Microsoft Active Directory Application Mode (ADAM) or Active Directory Lightweight Directory Service (AD LDS)	Oracle DSEE and Sun	OpenLDAP and Other LDAPv3 Types
User ID	*One of:* sAMAccountName mail employeeNumber telephoneNumber userPrincipalName	*One of:* uid mail employeeNumber telephoneNumber userPrincipalName	*One of:* uid mail employeeNumber telephonePhone	*One of:* uid mail employeeNumber telephonePhone
First Name	givenName	givenName	givenName	givenName
Middle Name	*One of:* middleName initials	*One of:* middleName initials	initials	initials
Last Name	sn	sn	sn	sn
Manager ID	manager	manager	manager	manager
Department	department	department	departmentnumber	departmentnumber
Phone Number	*One of:* telephoneNumber ipPhone	*One of:* telephoneNumber ipPhone	telephoner	telephonenumber
Mail ID	*One of:* mail sAMAccountName	*One of:* mail uid	*One of:* mail uid	*One of:* mail uid
objectGUID	objectGUID	objectGUID	not applicable	not applicable
OCSPrimaryUser Address	msRTCSIP-PrimaryUser Address	not applicable	not applicable	not applicable
Title	title	title	Title	title
Home Phone Number	homePhone	homePhone	Homephone	hometelephonenumber
Mobile Phone Number	mobile	mobile	Mobile	Mobiletelephonenumber
Pager Number	pager	pager	Pager	Pagertelephonenumber
Directory URI	*One of:* msRTCSIP-PrimaryUser Address mail none	*One of:* mail none	*One of:* mail none	*One of:* mail none
Display Name	displayName	displayName	displayName	displayName

In addition to the direct mapping of directory attributes to local user attributes, other characteristics of the synchronized users are determined by settings on the LDAP directory synchronization agreement. Access control group membership of users created through LDAP synchronization is directly configured in the LDAP directory configuration setting. Further user capabilities are determined by the feature group template selected. The selection of a feature group template on an LDAP directory synchronization agreement is optional. The feature group templates allow administrators to define user characteristics, including home cluster selection, IM and Presence capabilities, mobility features, services profiles, and user profiles. The user profiles allow administrators to define a universal line template that is considered for automatic creation of directory numbers for LDAP synchronized users by Unified CM.

The synchronization is performed by a process called Cisco DirSync, which is enabled through the Serviceability web page. When enabled, it allows one to 20 synchronization agreements to be configured in the system. This number is reduced to 10 if more than 80,000 users are synchronized. An agreement specifies a search base that is a position in the LDAP tree where Unified CM will begin its search for user accounts to import. Unified CM can import only users that exist in the domain specified by the search base for a particular synchronization agreement.

In Figure 16-6, two synchronization agreements are represented. One synchronization agreement specifies User Search Base 1 and imports users jsmith, jdoe and jbloggs. The other synchronization agreement specifies User Search Base 2 and imports users jjones, bfoo, and tbrown. The CCMDirMgr account is not imported because it does not reside below the point specified by a user search base. When users are organized in a structure in the LDAP directory, you can use that structure to control which user groups are imported. In this example, a single synchronization agreement could have been used to specify the root of the domain, but that search base would also have imported the Service Accts. The search base does not have to specify the domain root; it may specify any point in the tree.

Figure 16-6 User Search Bases



To import the data into the Unified CM database, the system performs a bind to the LDAP directory using the account specified in the configuration as the LDAP Manager Distinguished Name, and reading of the database is done with this account. The account must be available in the LDAP directory for Unified CM to log in, and Cisco recommends that you create a specific account with permissions to allow it to read all user objects within the sub-tree that was specified by the user search base. The sync agreement specifies the full Distinguished Name of that account so that the account may reside anywhere within that domain. In the example in Figure 16-6, CCMDirMgr is the account used for the synchronization.

It is possible to control the import of accounts through use of permissions of the LDAP Manager Distinguished Name account. In this example, if that account is restricted to have read access to ou=Eng but not to ou=Mktg, then only the accounts located under Eng will be imported.

Synchronization agreements have the ability to specify multiple directory servers to provide redundancy. You can specify an ordered list of up to three directory servers in the configuration that will be used when attempting to synchronize. The servers are tried in order until the list is exhausted. If none of the directory servers responds, then the synchronization fails, but it will be attempted again according to the configured synchronization schedule.

Synchronization Mechanism

The synchronization agreement specifies a time for synchronizing to begin and a period for re-synchronizing that can be specified in hours, days, weeks, or months (with a minimum value of 6 hours). A synchronization agreement can also be set up to run only once at a specific time.

When synchronization is enabled for the first time on a Unified CM publisher server, user accounts that exist in the corporate directory are imported into the Unified CM database. Then either existing Unified CM end-user accounts are activated and data is updated, or a new end-user account is created according to the following process:

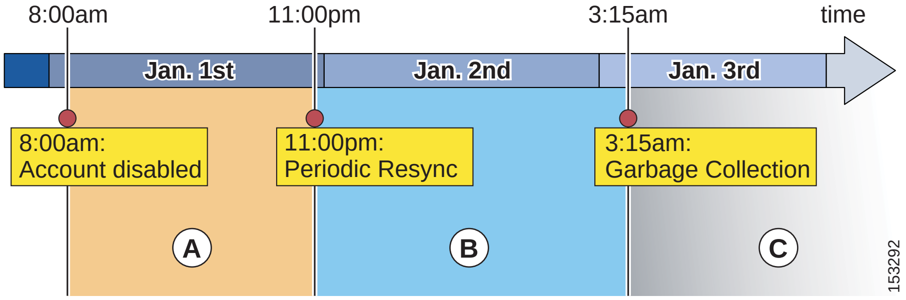
1. If end-user accounts already exist in the Unified CM database and a synchronization agreement is configured, all pre-existing accounts that have been synchronized from LDAP previously are marked inactive in Unified CM. The configuration of the synchronization agreement specifies a mapping of an LDAP database attribute to the Unified CM UserID. During the synchronization, accounts from the LDAP database that match an existing Unified CM account cause that Unified CM account to be marked active again.
2. After the synchronization is completed, any LDAP synchronized accounts that were not set to active are permanently deleted from Unified CM when the garbage collection process runs. Garbage collection is a process that runs automatically at the fixed time of 3:15 AM, and it is not configurable.
3. Subsequently when changes are made in the corporate directory, the synchronization from Microsoft Active Directory occurs as a full re-synchronization at the next scheduled synchronization period. On the other hand, the Sun ONE directory products perform an incremental synchronization triggered by a change in the directory. The following sections present examples of each of these two scenarios.

Note Once users are synchronized from LDAP into the Unified CM database, deletion of a synchronization configuration will cause users that were imported by that configuration to be marked inactive in the database. Garbage collection will subsequently remove those users.

Account Synchronization with Active Directory

Figure 16-7 shows an example timeline of events for a Unified CM deployment where LDAP Synchronization and LDAP Authentication have both been enabled. The re-synchronization is set for 11:00 PM daily.

Figure 16-7 Change Propagation with Active Directory



After the initial synchronization, the creation, deletion, or disablement of an account will propagate to Unified CM according to the timeline shown in Figure 16-7 and as described in the following steps:

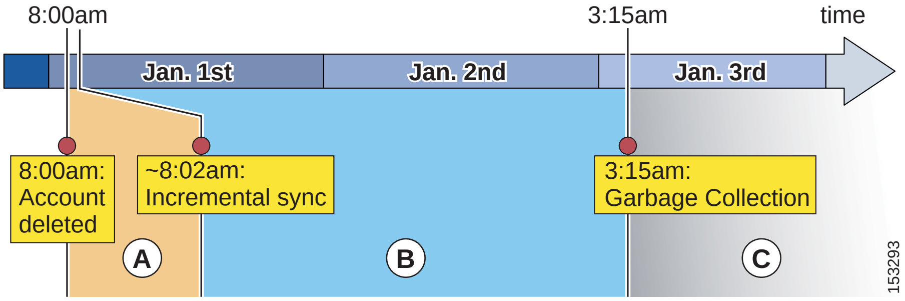
1. At 8:00 AM on January 1, an account is disabled or deleted in AD. From this time and during the whole period A, password authentication (for example, Unified CM User Options page) will fail for this user because Unified CM redirects authentication to AD. However, PIN authentication (for example, Extension Mobility login) will still succeed because the PIN is stored in the Unified CM database.
2. The periodic re-synchronization is scheduled for 11:00 PM on January 1. During that process, Unified CM will verify all accounts. Any accounts that have been disabled or deleted from AD will at that time be tagged in the Unified CM database as inactive. After 11:00 PM on January 1, when the account is marked inactive, both the PIN and password authentication by Unified CM will fail.
3. Garbage collection of accounts occurs daily at the fixed time of 3:15 AM. This process permanently deletes user information from the Unified CM database for any record that has been marked inactive for over 24 hours. In this example, the garbage collection that runs at 3:15 AM on January 2 does not delete the account because it has not been inactive for 24 hours yet, so the account is deleted at 3:15 AM on January 3. At that point, the user data is permanently deleted from Unified CM.

If an account has been created in AD at the beginning of period A, it will be imported to Unified CM at the periodic re-synchronization that occurs at the beginning of period B and will immediately be active on Unified CM.

Account Synchronization with Sun ONE

Sun ONE products support incremental synchronization agreements and use a different synchronization timeline than Microsoft Active Directory. The synchronization makes use of the Persistent Search mechanism supported by many LDAP implementations. Figure 16-8 shows an example of this synchronization timeline for a Unified CM deployment with LDAP Synchronization and LDAP Authentication both enabled.

Figure 16-8 Change Propagation with Sun ONE



The example in Figure 16-8 involves the following steps:

1. An account is deleted from the corporate directory at 8:00 AM on January 1, which causes an incremental update to be sent from the LDAP server to Unified CM. Unified CM sets its corresponding copy of the data to inactive. Because LDAP authentication is configured, the user will be unable to log in via password as soon as the LDAP server has deleted the record. Also, the PIN may not be used for login at the moment the Unified CM record is marked inactive.
2. During period B, the user's record is still present in Unified CM, albeit inactive.
3. When the garbage collection runs at 3:15 AM on January 2, the record has not yet been inactive for 24 hours. The data remains in the Unified CM database until the beginning of period C on January 3, when the garbage collection process runs again at 3:15 AM and determines that the record has been inactive for 24 hours or more. The record is then permanently deleted from the database.

Accounts that are newly created in the directory are synchronized to Unified CM via incremental updates as well, and they may be used as soon as the incremental update is received.

Automatic Line Creation

For users created during LDAP synchronization, Unified CM can automatically create directory numbers. These auto-generated directory numbers are either based on information found in the directory and defined based on a mask to be applied to the phone number found in the directory, or the numbers are taken from directory number pools defined on the LDAP synchronization agreement. If a mask is defined on the synchronization agreement, then to allow for variable length +E.164 directory numbers to be generated, the following rules apply:

- If the mask is left empty, then Unified CM takes all digits and also a leading "+" (if present) from the directory.
- X is used as a wildcard character in the mask.
- A wildcard matches on digits and "+".
- Wildcards in the mask are filled from the right.
- Unfilled wildcards in the mask are removed.

Table 16-5 shows some examples.

Table 16-5 Examples for Directory Number Creation from LDAP Phone Numbers Based on Masks

Number in LDAP	Mask	Result
14085551234		14085551234
14085551234	+XXXXXXXXXX	+14085551234
14085551234	+XXXXXXXXXXXXXXXXX	+14085551234
14085551234	XXXX	1234
+14085551234		+14085551234
+14085551234	+XXXXXXXXXXXXXXXXX	+14085551234
+496100123	+XXXXXXXXXXXXXXXXX	+496100123

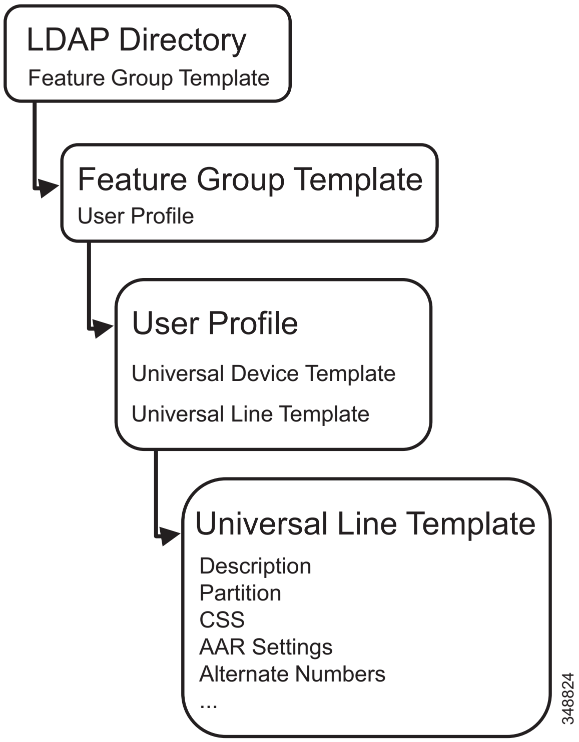
As an alternative to creating directory numbers based on information from LDAP, directory numbers for new users can also be taken from predefined number pools. Each pool is defined by a start and end number. Directory number pools support +E.164 numbers. Up to five pools can be defined. Numbers are assigned from the first pool until all numbers of that pools have been assigned. Number assignment then starts to take numbers from the next pool.

Automatic Line Creation is enabled only if *both* of the following conditions are met:

- A Feature Group Template is assigned in the directory synchronization agreement, **and**
- A Universal Line Template is selected in the User Profile selected in the Feature Group Template.

Figure 16-9 shows the hierarchy of configuration elements required to define line-level settings for automatic line creation.

Figure 16-9 Relation of LDAP Directory Configuration, Feature Group Template, User Profile, and Universal Line Template



Ultimately the Universal Line Template defines the characteristics for all directory numbers that are automatically created for users added through the corresponding LDAP synchronization definition.

Design Considerations

The calling search space defined in the Universal Line Template determines the class of service of devices using any of the auto-generated directory numbers. This implies that all directory numbers created through the same LDAP synchronization agreement share the same class of service, and thus if directory numbers for multiple sites and multiple classes of service need to be auto-generated, then multiple LDAP synchronization agreements (one per site and class of service) need to be configured. For each of these synchronization agreements, disjunct LDAP filters need to be defined, each exactly matching on only the users belonging to one of the site-specific and class-of-service-specific user groups. This mapping from LDAP attributes to site and class of service groups can be challenging unless the group membership based on site and class of service is explicitly encoded in few LDAP attributes (potentially even in a custom attribute). Also, the maximum number of supported LDAP agreements is limited, which limits the number of distinct user groups for which directory numbers can be created automatically.

Automatic creation of directory numbers applies only to users created during LDAP directory synchronization. Adding, changing, or updating the Universal Line Template for a given LDAP synchronization agreement will not create directory numbers for already existing users and will not change the settings of already existing directory numbers.

The Universal Line Template allows administrators to define call forward unregistered destinations and either to select voicemail as the forward destination or to define an explicit destination. To reach endpoints in remote sites from registered endpoints in case of WAN failure, the call forward unregistered destination for the remote site's phones must be set to the PSTN alias (+E.164 number) of the remote phone. This cannot be achieved with Universal Line Template settings because this would require defining the call forward unregistered destination to be set based on the assigned directory numbers (potentially with a mask applied).

Enterprise Group Support

To enable Jabber clients to search for groups in Microsoft Active Directory, you can configure Unified CM not only to synchronize end users from Active Directory but also to include distribution groups defined in Active Directory. Synchronization of enterprise groups is supported only with Microsoft Active Directory as the data source. It is not supported with Active Directory Lightweight Directory Services (AD LDS) or other corporate directories. Synchronization of enterprise groups is enabled in the Unified CM LDAP directory configuration. The maximum number of enterprise groups is 15,000 and the maximum number of members per group is 100. While groups and members cannot be added or modified in the Unified CM administration, the groups synchronized from Active Directory can be reviewed in the User Management/User Settings/User Group menu.

For each group member, the following information is available on Jabber clients:

- Display name
- User ID
- Title
- Phone number
- Mail ID

Security Considerations

During the import of accounts, no passwords or PINs are copied from the LDAP directory to the Unified CM database. If LDAP authentication is not enabled in Unified CM and single sign-on is not used, the password for the end user is managed by using Unified CM Administration. The password and PIN are stored in an encrypted format in the Unified CM database. The PIN is always managed on Unified CM. If you want to use the LDAP directory password to authenticate an end user, see the section on LDAP Authentication, page 16-22.

The connection between the Unified CM publisher server and the directory server can be secured by enabling Secure LDAP (SLDAP) on Unified CM and the LDAP server. Secure LDAP enables LDAP to be sent over a Secure Socket Layer (SSL) connection and can be enabled by adding the LDAP server into the Tomcat trust store within the Unified CM Platform Administration. For detailed procedure steps, refer to the Unified CM product documentation available at https://www.cisco.com. Refer to the documentation of the LDAP directory vendor to determine how to enable SLDAP.

Design Considerations for LDAP Synchronization

Observe the following design and implementation best practices when deploying LDAP synchronization with Cisco Unified CM:

- Use a specific account within the corporate directory to allow the Unified CM synchronization agreement to connect and authenticate to it. Cisco recommends that you use an account dedicated to Unified CM, with minimum permissions set to "read" all user objects within the desired search base and with a password set never to expire. The password for this account in the directory must be kept in synchronization with the password configuration of the account in Unified CM. If the service account password changes in the directory, be sure to update the account configuration in Unified CM.

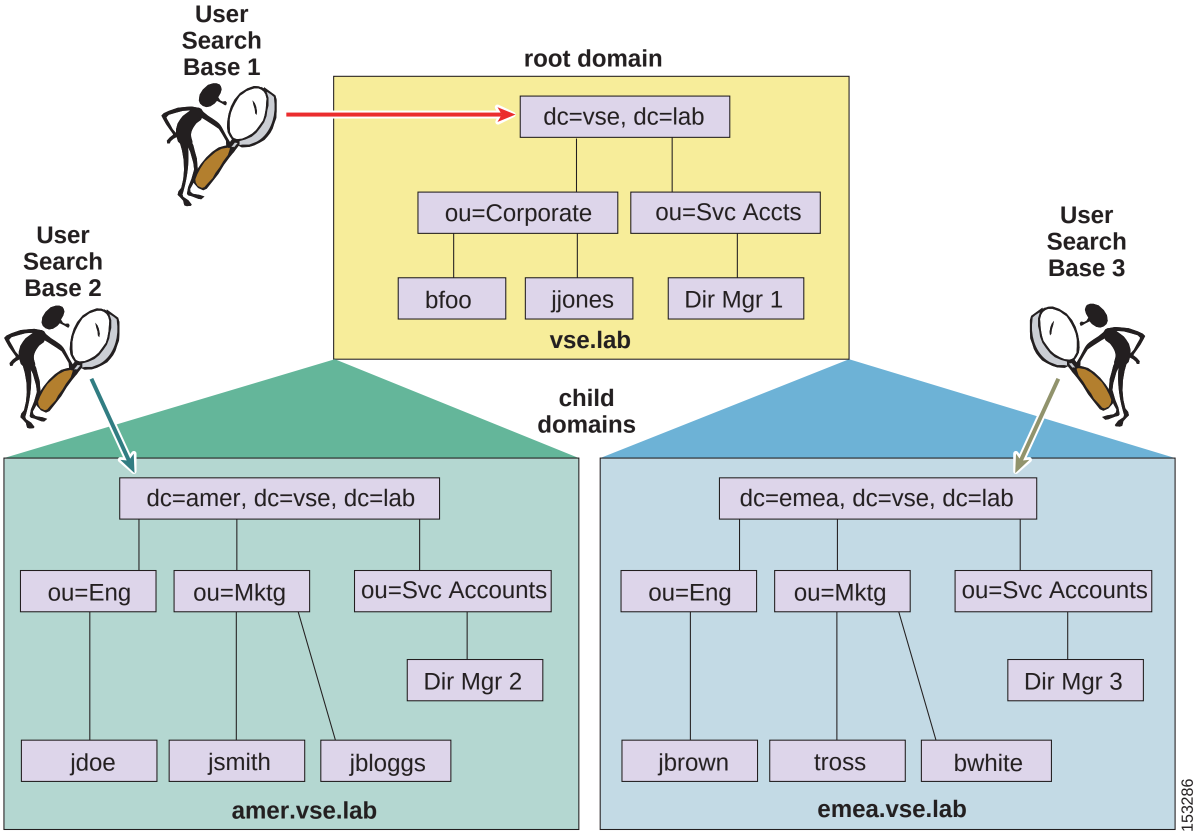
- All synchronization agreements on a given cluster must integrate with the same family of LDAP servers.
- Stagger the scheduling of synchronization agreements so that multiple agreements are not querying the same LDAP servers simultaneously. Choose synchronization times that occur during quiet periods (off-peak hours).
- If security of user data is required, enable Secure LDAP (SLDAP) by checking the **Use SSL** field on the LDAP Directory configuration page in Unified CM Administration.
- Ensure that the LDAP directory attribute chosen to map into the Unified CM UserID field is unique within all synchronization agreements for that cluster.
- The attribute chosen as UserID must not be the same as that for any of the Application Users defined in Unified CM.
- The LDAP attribute sn(lastname) is a mandatory attribute for LDAP Synchronization of users.
- An existing account in the Unified CM database before synchronization is maintained only if an account imported from the LDAP directory has a matching attribute. The attribute that is matched to the Unified CM UserID is determined by the synchronization agreement.
- Administer end-user accounts through the LDAP directory's management tools, and manage the Cisco-specific data for those accounts through the Unified CM Administration web page.
- For AD deployments, the ObjectGUID is used internally in Unified CM as the key attribute of a user. The attribute in AD that corresponds to the Unified CM User ID may be changed in AD. For example, if sAMAccountname is being used, a user may change their sAMAccountname in AD, and the corresponding user record in Unified CM would be updated.

  With all other LDAP platforms, the attribute that is mapped to User ID is the key for that account in Unified CM. Changing that attribute in LDAP will result in a new user being created in Unified CM, and the original user will be marked inactive.

Additional Considerations for Microsoft Active Directory

A synchronization agreement for a domain will not synchronize users outside of that domain nor within a child domain because Unified CM does not follow AD referrals during the synchronization process. The example in Figure 16-10 requires three synchronization agreements to import all of the users. Although Search Base 1 specifies the root of the tree, it will not import users that exist in either of the child domains. Its scope is only VSE.LAB, and separate agreements are configured for the other two domains to import those users.

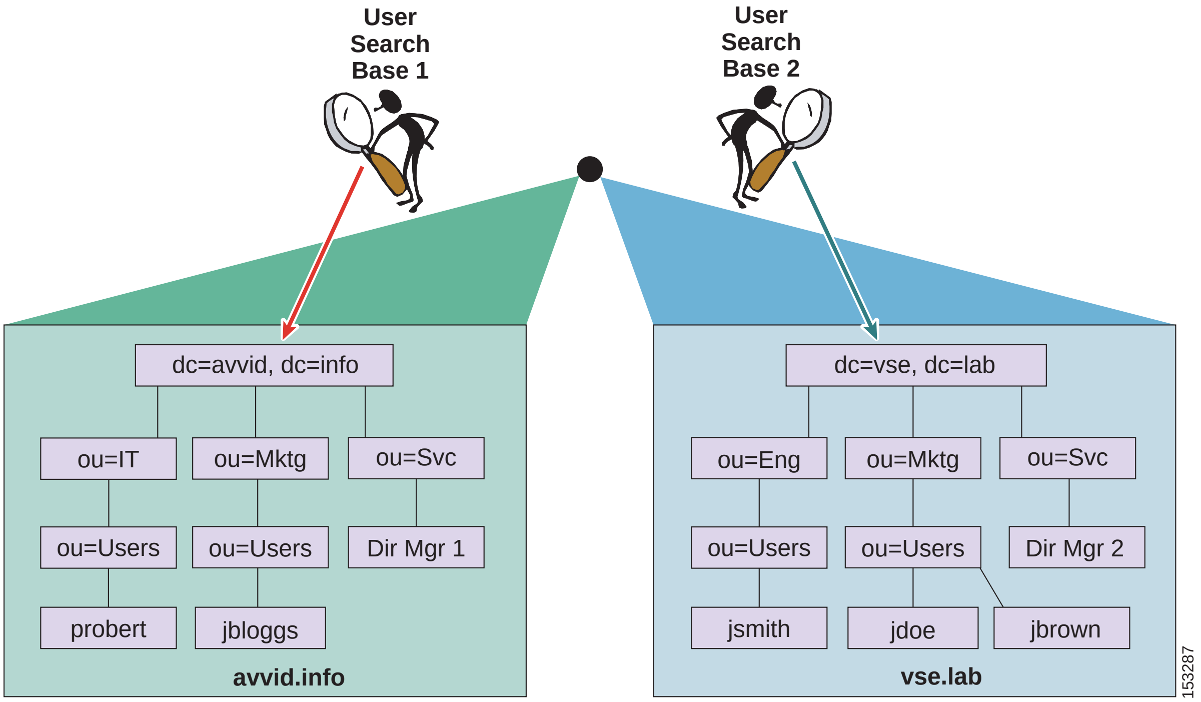
Figure 16-10 Synchronization with Multiple Active Directory Domains



In Figure 16-10, each of the domains and sub-domains contains at least one domain controller (DC) associated to them, and the three synchronization agreements each specify the appropriate domain controller. The DCs have information only on users within the domain where they reside, therefore three synchronization agreements are required to import all of the users.

When synchronization is enabled with an AD forest containing multiple trees, as shown in Figure 16-11, multiple synchronization agreements are still needed for the same reasons listed above. Additionally, the UserPrincipalName (UPN) attribute is guaranteed by Active Directory to be unique across the forest and must be chosen as the attribute that is mapped to the Unified CM UserID. For additional considerations on the use of the UPN attribute in a multi-tree AD scenario, see the section on Additional Considerations for Microsoft Active Directory, page 16-26.

Figure 16-11 Synchronization with Multiple AD Trees (Discontiguous Namespaces)



Unified CM sends a default LDAP search filter string to AD when performing the synchronization of accounts. One of the clauses is to not return accounts that have been marked as disabled in AD. An account marked disabled by AD, such as when failed login attempts are exceeded, will be marked inactive if synchronization runs while the account is disabled.

Unified CM Multi-Forest LDAP Synchronization

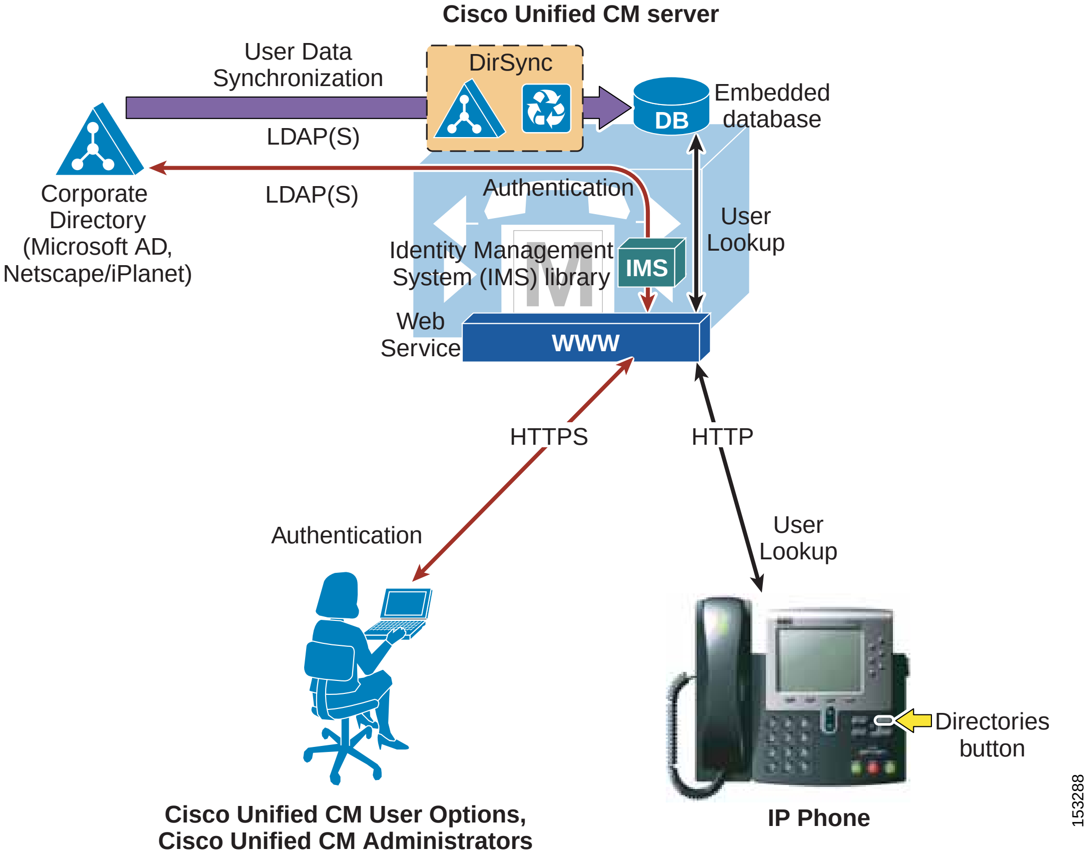
A Unified CM deployment using a multi-forest LDAP infrastructure can be supported by using Active Directory Lightweight Directory Services (AD LDS) as a single forest view integrating with the multiple disparate forests. The integration also requires the use of LDAP filtering (see User Filtering for Directory Synchronization and Authentication, page 16-28). For full details, refer to the document on *How to Configure Unified Communication Manager Directory Integration in a Multi-Forest Environment*, available at

https://www.cisco.com/en/US/products/sw/voicesw/ps556/products_configuration_example09186a0080b2b103.shtml

LDAP Authentication

The LDAP authentication feature enables Unified CM to authenticate LDAP synchronized users against the corporate LDAP directory. Application users and locally configured users are always authenticated against the local database. Also PINs of all end users are always checked against the local database only. This authentication is accomplished with an LDAPv3 connection established between the Identity Management System (IMS) module within Unified CM and a corporate directory server, as shown in Figure 16-12.

Figure 16-12 Enabling LDAP Authentication



To enable authentication, a single authentication agreement may be defined for the entire cluster. The authentication agreement supports configuration of up to three LDAP servers for redundancy and also supports secure connections LDAP over SSL (SLDAP) if desired. Authentication can be enabled only when LDAP synchronization is properly configured and used. LDAP authentication configuration is overridden by enabling SSO. With SSO enabled, end users are always authenticated using SSO, and LDAP authentication configuration is ignored.

The following statements describe Unified CM's behavior when authentication is enabled:

- End user passwords of users imported from LDAP are authenticated against the corporate directory by a simple bind operation.
- End user passwords for local users are authenticated against the Unified CM database.
- Application user passwords are authenticated against the Unified CM database.
- End user PINs are authenticated against the Unified CM database.

This behavior is in line with the guiding principle of providing single logon functionality for end users while making the operation of the real-time Unified Communications system independent of the availability of the corporate directory, and is shown graphically in Figure 16-13.

Figure 16-13 Authenticating End User Passwords, Application User Passwords, and End User PINs

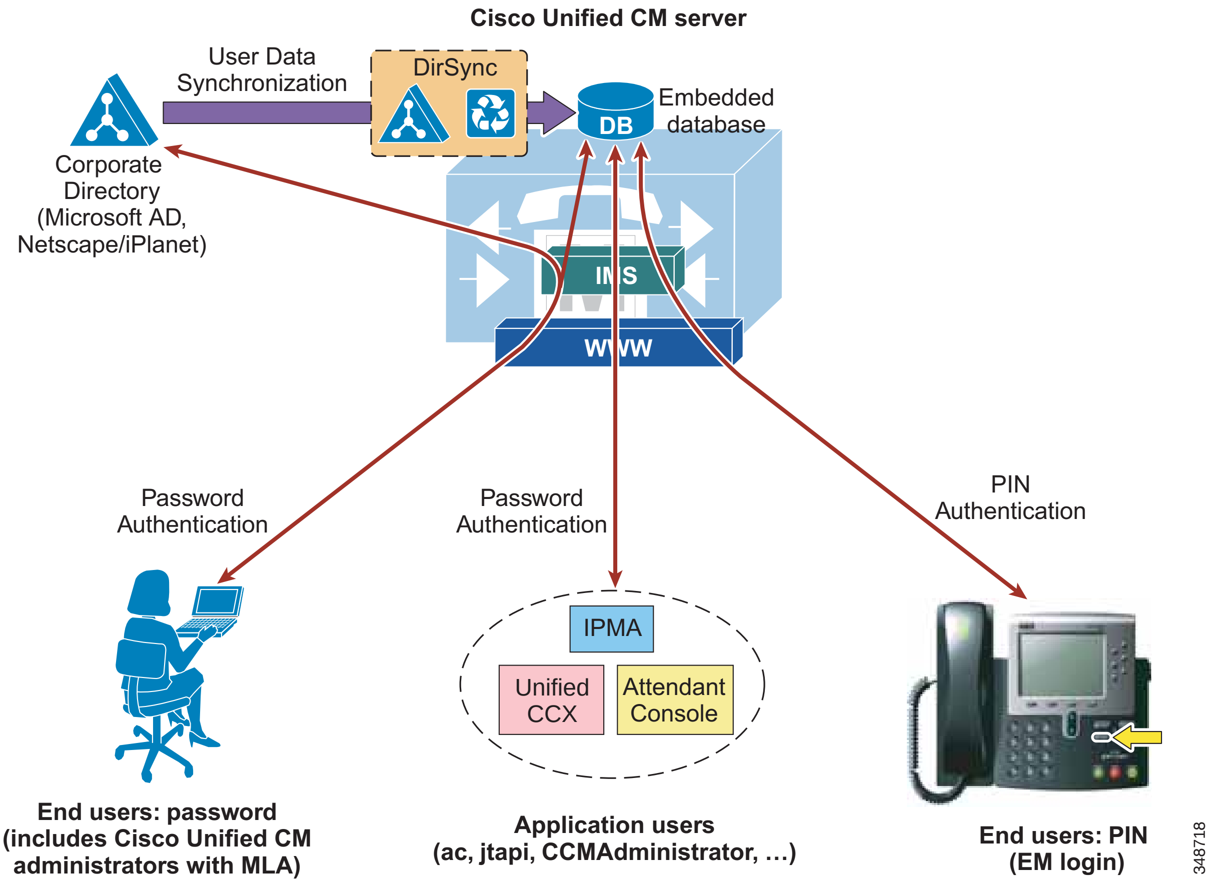

Cisco Unified CM server
User Data Synchronization
DirSync
Embedded database
DB
Corporate Directory (Microsoft AD, Netscape/iPlanet)
IMS
WWW
Password Authentication
Password Authentication
PIN Authentication
IPMA
Unified CCX
Attendant Console
End users: password (includes Cisco Unified CM administrators with MLA)
Application users (ac, jtapi, CCMAdministrator, …)
End users: PIN (EM login)
348718

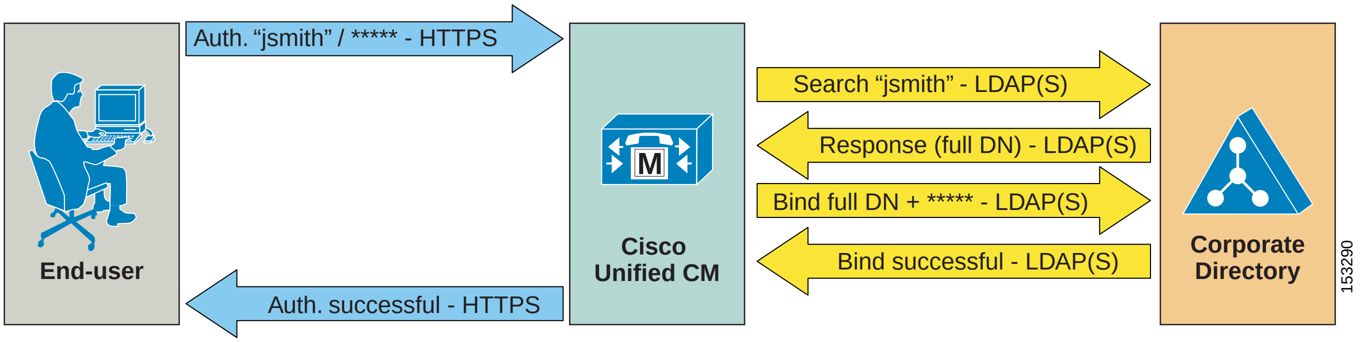
Figure 16-14 illustrates the following process, adopted by Unified CM to authenticate an end user synchronized from LDAP against a corporate LDAP directory:

1. A user connects to the Unified CM User Options page via HTTPS and attempts to authenticate with a user name and password. In this example, the user name is jsmith.
2. If the user is a local user, the password is checked against the local database.

The following steps apply only to LDAP synchronized users:

3. If the user is an LDAP synchronized user, Unified CM issues an LDAP query for the user name jsmith, using the value specified in the LDAP Search Base on the LDAP Authentication configuration page as the scope for this query. If SLDAP is enabled, this query travels over an SSL connection.
4. The corporate directory server replies via LDAP with the full Distinguished Name (DN) of user jsmith (for example, "cn=jsmith, ou=Users, dc=vse, dc=lab").
5. Unified CM then attempts to validate the user's credentials by using an LDAP bind operation to pass the full DN and password provided by the user.
6. If the LDAP bind is successful, Unified CM allows the user to proceed to the configuration page requested.

Figure 16-14 Authentication Process


End-user
Auth. "jsmith" / ***** - HTTPS
Auth. successful - HTTPS
Cisco Unified CM
Search "jsmith" - LDAP(S)
Response (full DN) - LDAP(S)
Bind full DN + ***** - LDAP(S)
Bind successful - LDAP(S)
Corporate Directory
153290

Design Considerations for LDAP Authentication

Observe the following design and implementation best-practices when deploying LDAP authentication with Cisco Unified CM:

- Create a specific account within the corporate directory to allow Unified CM to connect and authenticate to it. Cisco recommends that you use an account dedicated to Unified CM, with minimum permissions set to "read" all user objects within the desired search base and with a password set to never expire. The password for this account in the directory must be kept in synchronization with the password configuration of the account in Unified CM. If the account password changes in the directory, be sure to update the account configuration in Unified CM. If LDAP synchronization is also enabled, you can use the same account for both functions.
- Enable LDAP authentication on Unified CM by specifying the credentials of the aforementioned account under LDAP Manager Distinguished Name and LDAP Password, and by specifying the directory subtree where all the users reside under LDAP User Search Base.
- This method provides single logon functionality to all end users synchronized from LDAP. They can then use their corporate directory credentials to log in to the Unified CM User Options page.
- Manage end-user passwords for LDAP synchronized users from within the corporate directory interface. Note that the password field is no longer displayed for LDAP synchronized users in the Unified CM Administration pages when authentication is enabled.
- Manage end-user PINs from the Unified CM Administration web pages or from the Unified CM User Options page.
- Manage Application User passwords from the Unified CM Administration web pages. Remember that these application users facilitate communication and remote call control with other Cisco Unified Communications applications and are not associated with real people.
- Enable single logon for Unified CM administrators by adding their corresponding end user to the Unified CM Super Users user group from the Unified CM Administration web pages. Multiple levels of administrator rights can be defined by creating customized user groups and roles.

Additional Considerations for Microsoft Active Directory

In environments that employ a distributed AD topology with multiple domain controllers geographically distributed, authentication speed might be unacceptable. When the Domain Controller for the authentication agreement does not contain a user account, a search must occur for that user across other domain controllers. If this configuration applies, and login speed is unacceptable, it is possible to set the authentication configuration to use a Global Catalog Server.

An important restriction exists, however. A Global Catalog does not carry the employeeNumber attribute by default. In that case either use Domain Controllers for authentication (beware of the limitations listed above) or update the Global Catalog to include the employeeNumber attribute. Refer to Microsoft Active Directory documentation for details.

To enable queries against the Global Catalog, simply configure the LDAP Server Information in the LDAP Authentication page to point to the IP address or host name of a Domain Controller that has the Global Catalog role enabled, and configure the LDAP port as 3268.

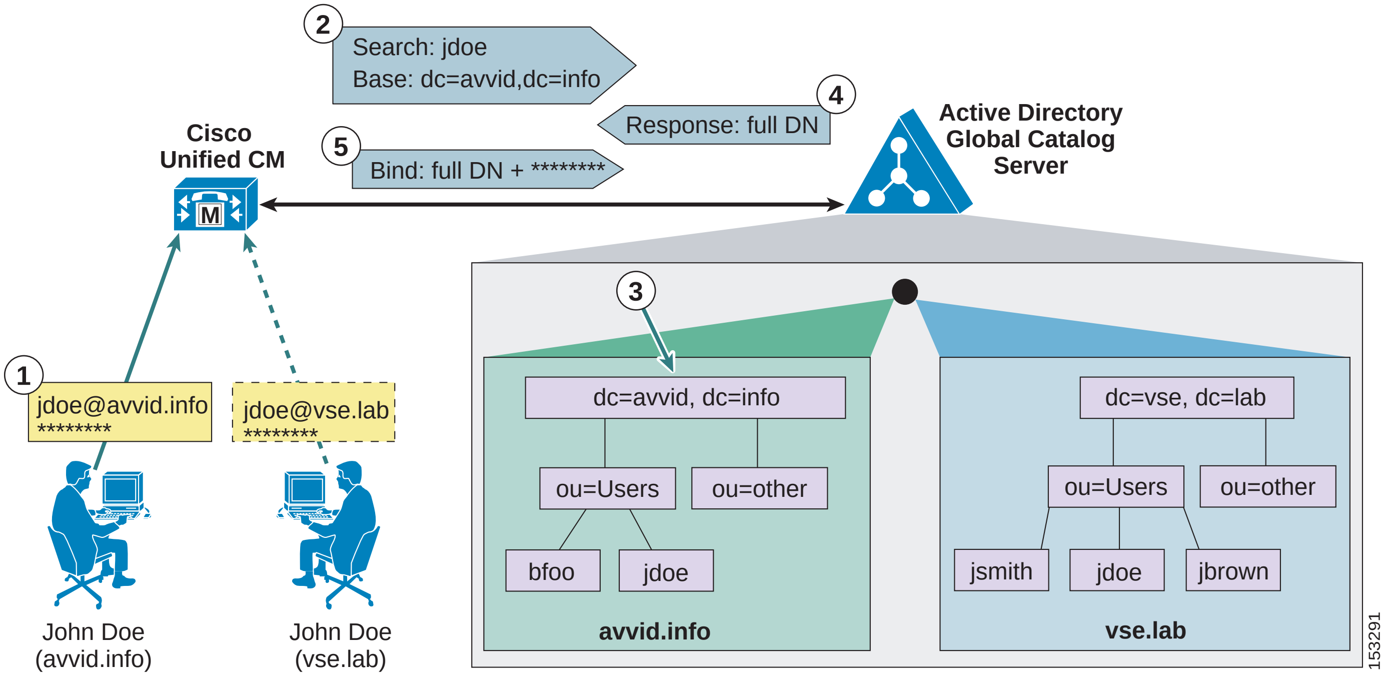
The use of Global Catalog for authentication becomes even more efficient if the users synchronized from Microsoft AD belong to multiple domains, because it allows Unified CM to authenticate users immediately without having to follow referrals. For these cases, point Unified CM to a Global Catalog server and set the LDAP User Search Base to the top of the root domain.

In the case of a Microsoft AD forest that encompasses multiple trees, some additional considerations apply. Because a single LDAP search base cannot cover multiple namespaces, Unified CM must use a different mechanism to authenticate users across these discontiguous namespaces.

As mentioned in the section on LDAP Synchronization, page 16-10, in order to support synchronization with an AD forest that has multiple trees, the UserPrincipalName (UPN) attribute should be used as the user ID within Unified CM. When the user ID is the UPN, the LDAP authentication configuration page within Unified CM Administration does not allow you to enter the LDAP Search Base field, but instead it displays the note, "LDAP user search base is formed using userid information."

In fact, the user search base is derived from the UPN suffix for each user, as shown in Figure 16-15. In this example, a Microsoft Active Directory forest consists of two trees, avvid.info and vse.lab. Because the same user name may appear in both trees, Unified CM has been configured to use the UPN to uniquely identify users in its database during the synchronization and authentication processes.

Figure 16-15 Authentication with Microsoft AD Forests with Multiple Trees



As shown in Figure 16-15, a user named John Doe exists in both the avvid.info tree and the vse.lab tree. The following steps illustrate the authentication process for the first user, whose UPN is jdoe@avvid.info:

1. The user authenticates to Unified CM via HTTPS with its user name (which corresponds to the UPN) and password.
2. Unified CM performs an LDAP query against a Microsoft Active Directory Global Catalog server, using the user name specified in the UPN (anything before the @ sign) and deriving the LDAP search base from the UPN suffix (anything after the @ sign). In this case, the user name is jdoe and the LDAP search base is "dc=avvid, dc=info".
3. Microsoft Active Directory identifies the correct Distinguished Name corresponding to the user name in the tree specified by the LDAP query. In this case, "cn=jdoe, ou=Users, dc=avvid, dc=info".
4. Microsoft Active Directory responds via LDAP to Unified CM with the full Distinguished Name for this user.
5. Unified CM attempts an LDAP bind with the Distinguished Name provided and the password initially entered by the user, and the authentication process then continues as in the standard case shown in Figure 16-14.

Note Support for LDAP authentication with Microsoft AD forests containing multiple trees relies exclusively on the approach described above. Therefore, support is limited to deployments where the UPN suffix of a user corresponds to the root domain of the tree where the user resides. AD allows the use of aliases, which allows a different UPN suffix. If the UPN suffix is disjointed from the actual namespace of the tree, it is not possible to authenticate Unified CM users against the entire Microsoft Active Directory forest. (It is, however, still possible to use a different attribute as user ID and limit the integration to a single tree within the forest.)

User Filtering for Directory Synchronization and Authentication

Unified CM provides an LDAP Query Filter to optimize directory synchronization performance. Cisco recommends importing those directory user accounts that will be assigned to Unified Communications resources. To allow for enterprise-wide UDS based service discovery, all users assigned to Unified Communications resources on any cluster in the enterprise need to be imported to all clusters in the enterprise. Differentiation between local and remote users is achieved by the **Home Cluster** setting on the Feature Group Template associated with the LDAP synchronization agreement that is used. When the number of directory user accounts exceeds the number supported for an individual cluster, filtering must be used to select the subset of users that will be associated on that cluster. The Unified CM synchronization feature is not meant to replace a large-scale corporate directory.

In many cases, a unique search base is all that is needed to control which accounts are synchronized. When a unique search base is not available, a custom LDAP filter might be required. The information in the following sections addresses both methods that can be used to optimize directory synchronization. When any mechanism is used to limit the accounts imported into Unified CM, the default directory lookup configuration will list only those directory entries that exist in the Unified CM database unless the UDS LDAP proxy functionality is used. For directory lookup to access the entire directory, you also can configure Unified CM to utilize an external web server. Details of this configuration are not discussed here but are discussed in the Unified CM product documentation available at

https://www.cisco.com/en/US/partner/products/sw/voicesw/ps556/tsd_products_support_series_home.html

Optimizing Unified CM Database Synchronization

The Unified CM Database Synchronization feature provides a mechanism for importing a subset of the user configuration data (attributes) from the LDAP directory store into the Unified CM publisher database. Once synchronization of a user account has occurred, the copy of each user's LDAP account information may then be associated to additional data required to enable specific Unified Communications features for that user. When authentication is also enabled, the user's credentials are used to bind to the LDAP store for password verification. The end user's password is never stored in the Unified CM database when enabled for synchronization and/or authentication.

User account information is cluster-specific. Each Unified CM publisher server maintains a unique list of those users receiving Unified Communications services from that cluster. Synchronization agreements are cluster-specific, and each publisher has its own unique copy of user account information. Only those users who will be assigned Unified Communications resources should be synchronized with Unified CM. The following is a partial list of common reasons why the entire set of users defined in the LDAP directory should not be imported into the Unified CM cluster:

- Importing users who will not be assigned Unified Communications resources can increase directory synchronization time.
- Importing users who will not be assigned Unified Communications resources can slow Unified CM searches and overall database performance.
- In many cases, the number of user accounts in the LDAP directory store far exceeds the total user capacity of the Unified CM database.

Unified CM has no enforced limit on the number of accounts that may be added to the system. Cisco recommends limiting the number of users to twice the supported number of endpoints. There might be cases where accounts are needed for applications, and some designs might require additional accounts.

Cisco recommends using the control mechanisms described here to minimize the number of user accounts imported, regardless of the LDAP database size. This will improve the speed of the first and subsequent periodic synchronizations and will also improve manageability of the user accounts.

Using the LDAP Structure to Control Synchronization

Many deployments of LDAP directories use the Organizational Unit Name (OU) to group users into a logical order and sometimes hierarchical order. If the LDAP directory has a structure that organizes users into multiple OUs, then it often is possible to use that structure to control the groups of users imported. Each individual Unified CM synchronization agreement specifies a single OU. All active accounts under the specified OU, even within sub-OUs, are imported. Only those users in the OU are synchronized. When multiple OUs containing users are required in a cluster, multiple synchronization agreements are required. When an OU contains users that will not be assigned Unified Communications resources, Cisco recommends omitting those OUs from the directory synchronization.

The same technique may be used with AD, which defines containers. A synchronization agreement may specify a particular container in the directory tree and thereby limit the extent of the import.

Because there is only a limited number of synchronization agreements available, LDAP deployments with many OUs or containers can quickly exhaust this technique. One possible method to synchronize users in a multi-OU environment is to control the permissions assigned to the synchronization service account. Configure the synchronization agreement to a tree node that contains a mix of users, and then restrict the system account from read access to selected parts of the subtree. Refer to your LDAP vendor documentation on how to restrict this access.

LDAP Query

Additional control over filtering might be required for any of the following reasons:

- The LDAP directory has a flat structure that does not enable adequate control by configuration of the synchronization agreements. When the aggregate number of users that are imported by all the synchronization agreements is greater than the maximum number of users supported by the Unified CM cluster, then it is necessary to control the number of users imported through filters.
- You want to import a subset of user accounts into the Unified CM cluster, for administrative segmentation of users, to control a subset of users that have access and authentication to the cluster. Any account that is imported into a cluster has some level of access to the web pages and authentication mechanisms, which might not be desirable in some cases.
- The LDAP directory structure does not have an accurate representation of how users are going to be mapped into the Unified CM clusters. For instance, if OUs are set up according to an organizational hierarchy but users are mapped to Unified CM by geography, there might be little overlap between the two.

In these cases, the LDAP Query filter may be used to provide additional control over the synchronization agreements.

LDAP Query Filter Syntax and Server-Side Filtering

Unified CM uses standard LDAP mechanisms for synchronizing data from an LDAP directory store. It utilizes the Search mechanism, as defined by RFC 4510 et seq., to send a request to retrieve data from the LDAP server. Also defined by that mechanism is the ability to specify a filter string inside the Search message that is used by the LDAP server to select entries in the database for which to return data. The syntax of the filter string is defined by RFC 4515 String Representation of Search Filters. This RFC may be used as a reference for constructing more complex filter strings.

The filter string is embedded within a Search message that is sent by Unified CM to the LDAP server and is executed by the server to select which user accounts will be provided in the response.

Simple Filter Syntax

You can configure a filter by specifying standard attribute names and values that are desired for those attributes. The attributes may also be specified by DN element instead of name. The filter string that is used by Unified CM in LDAP queries is stored internally in the ldapfilter table and is the string inserted into the Search message.

A filter is a UTF-8 formatted string that has the following syntax:

(*attribute operator value***)**

or

(*operator***(***filter1***)(***filter2***))**

Where *filter1* and *filter2* have the syntax shown in first line, and the *operator* is one of those listed in Table 16-6. The *attribute* corresponds to an LDAP attribute that exists in the directory, *operator* is one of the operators listed in Table 16-6, and *value* corresponds to the actual data value that is requested for the attribute.

Table 16-6 Basic Filter String Operators

Operator	Meaning of Function
!	Logical NOT
&	Logical AND
|	Logical OR
*	Wildcard
=	Equal to
>=	Lexicographically greater than or equal to
<=	Lexicographically less than or equal to

An attribute specified in the filter can be any attribute that exists in the LDAP directory store, and it does not have to be one of the attributes that is understood and imported by Unified CM. The attribute is used only on the LDAP server to select data, and the corresponding entries will have a subset of their data imported into Unified CM.

Example 16-1 A Single Condition

```
(givenName=Jack)
```

The filter in Example 16-1 selects any user with a given name of Jack.

Example 16-2 Multiple Conditions May Be Joined with Logical Characters

```
(&(objectclass=user)(department=Engineering))
```

The filter in Example 16-2 selects all users in the engineering department.

Default Filter Strings

If no custom filter strings are defined, Unified CM uses a default LDAP filter string as follows:

- Default Active Directory (AD) filter string

  (&(objectclass=user)(!(objectclass=Computer))(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))

  This default filter selects entries for which the object class is a user but not a computer, and for which the account is not flagged as disabled.

- Default Active Directory Application Mode (ADAM) or Active Directory Lightweight Directory Services (AD LDS) filter string

  (&(objectclass=user)((objectclass=Computer))(!(msDS-UserAccountDisabled=TRUE)))

- Default filter string for all other directory types

  (objectclass=inetOrgPerson)

Extending the Default Filter

Cisco recommends that you use the default filter string and append additional conditions to it. For example:

(&(objectclass=user)(!(objectclass=Computer))(!(UserAccountControl:1.2.840.113556.1.4.803:=2))(telephonenumber=+1919*))

This filter selects only users that have a prefix of +1919 in their telephonenumber field. The synchronization agreement will import only users with an area code of 919 in the US. This example assumes all entries are in +E.164 format.

For the search filter, you may use any existing attribute or even a custom attribute that is defined in the LDAP directory store. The filter string controls which records are selected by the LDAP server to be returned to Unified CM, but the attributes that are imported are not affected by the filter string.

Custom LDAP filter strings can be up to 2048 characters long. Custom LDAP filters first need to be created, and then existing custom LDAP filters can be assigned to LDAP synchronization agreements. Different LDAP synchronization agreements can use different custom LDAP filters.

High Availability

Unified CM LDAP Synchronization allows for the configuration of up to three redundant LDAP servers for each directory synchronization agreement. Unified CM LDAP Authentication allows for the configuration of up to three redundant LDAP servers for a single authentication agreement. You should configure a minimum of two LDAP servers for redundancy. The LDAP servers can be configured with IP addresses instead of host names to eliminate dependencies on Domain Name System (DNS) availability.

Capacity Planning for Unified CM Database Synchronization

The Unified CM Database Synchronization feature provides a mechanism for importing a subset of the user configuration data (attributes) from the LDAP store into the Unified CM publisher database. Once synchronization of a user account has occurred, the copy of each user's LDAP account information may then be associated to additional data required to enable specific Unified Communications features for that user. When authentication is also enabled, the user's credentials are used to bind to the LDAP store for password verification. The end user's password is never stored in the Unified CM database when enabled for synchronization and/or authentication.

User account information is cluster-specific. Each Unified CM publisher server maintains a unique list of those users receiving Unified Communications services from that cluster. Synchronization agreements are cluster-specific, and each publisher has its own unique copy of user account information.

The maximum number of users that a Unified CM cluster can handle is limited by the maximum size of the internal configuration database that gets replicated between the cluster members. The maximum number of users that can be configured or synchronized is 160,000. With more than 80,000 users the maximum number of LDAP synchronization agreements is limited to 10, while with less than 80,000 users the total number of LDAP synchronization agreements is limited to 20. To optimize directory synchronization performance, Cisco recommends considering the following points:

- Directory lookup from phones and web pages may use the Unified CM database, the IP Phone Service SDK, or the UDS LDAP proxy functionality. When directory lookup functionality uses the Unified CM database, only users who were configured or synchronized from the LDAP store are shown in the directory. If a subset of users are synchronized, then only that subset of users are seen on directory lookup.
- When the IP Phone Services SDK is used for directory lookup, but authentication of Unified CM users to LDAP is needed, the synchronization can be limited to the subset of users who would log in to the Unified CM cluster.
- If only one cluster exists, and the LDAP store contains fewer than the maximum number of users supported by the Unified CM cluster, and directory lookup is implemented to the Unified CM database, then it is possible to import the entire LDAP directory.
- When multiple clusters exist and the number of users in LDAP is less than the maximum number of users supported by the Unified CM cluster, it is possible to import all users into every cluster to ensure directory lookup has all entries.
- If the number of user accounts in LDAP exceeds the maximum number of users supported by the Unified CM cluster and the entire user set should be visible to all users, it will be necessary to use the Unified IP Phone Services SDK to off-load the directory lookup from Unified CM.
- If both synchronization and authentication are enabled, user accounts that have either been configured or synchronized into the Unified CM database will be able to log in to that cluster. The decision about which users to synchronize will impact the decision on directory lookup support.

Note Cisco supports the synchronization of user accounts up to the limit mentioned above, but it does not enforce this limit. Synchronizing more user accounts can lead to starvation of disk space, slower database performance, and longer upgrade times.

UDS Proxy for LDAP

Clients that use User Data Service (UDS) for contact source access are limited to accessing only the users that exist in the Unified CM end-user database. Although this database can be populated from the corporate directory via LDAP synchronization, the number of users searched is still limited by the maximum number of users supported in Unified CM. (See the section on Capacity Planning for Unified CM Database Synchronization, page 16-31, for details.) To overcome this limitation, Cisco Unified CM 11.5 and later releases can be set up to act as a UDS-to-LDAP proxy for UDS-based user searches. In this mode, for every user search requested via UDS, Unified CM connects back to the corporate directory to execute the search and then relays the results back to the client via UDS. Instead of serving the UDS search requests directly, Unified CM in this mode relies on the information returned from the corporate LDAP directory.

Note The recommended contact source for on-premises Jabber deployments is Cisco Directory Integration (CDI). UDS proxy for LDAP should be used only for deployments where endpoints rely on UDS for contact searches and where the number of users in the corporate directory exceeds the maximum number of end users supported in Cisco Unified CM.

UDS proxy functionality is enabled globally in Unified CM. Up to three directory Unified Communications services can be selected to define the LDAP data sources for the proxy functionality. Unified CM uses an LDAP bind to execute the search operations, and the user and password to be used for this bind operation are configured specifically for this feature. The UDS proxy supports up to three LDAP user search bases.

Directory Integration for VCS Registered Endpoints

Cisco TelePresence Video Communication Server (VCS) endpoints are managed by VCS and as such they can receive directory information from the Cisco TelePresence Management Suite (TMS). Cisco TelePresence Management Suite offers many more services such as scheduling of Unified CM and VCS registered endpoints, and management of VCS registered endpoints.

Cisco TelePresence Management Suite can manage multiple phone books coming from multiple sources.

Cisco TMS 14.1 can also integrate with Cisco Unified Communications Manager and receive directory information from Unified CM. This is the recommended configuration in order to have a unified directory for Unified CM and VCS endpoints.

Multiple Unified CM clusters can be added as multiple directory sources to Cisco TMS and organized in a single directory. TMS can push directory information to endpoints connected to it and registered to a single VCS or to multiple VCSs.

For more information, refer to the latest versions of the *Cisco TelePresence Management Suite Administrator Guide* and the *Cisco TelePresence Management Suite Provisioning Extension Deployment Guide*, both available at

https://www.cisco.com/en/US/products/ps11338/tsd_products_support_series_home.html

Identity Management Architecture Overview

Figure 16-16 presents an overview of the identity management architecture. All Cisco Collaboration Applications (for example, Cisco Unified CM with IM and Presence, and Cisco Unity Connection) maintain their individual identity stores. Users in these identity stores can be synchronized from the enterprise directory by means of individual LDAP sync agreements, but they can also be configured locally. Synchronizing from LDAP is highly recommended to make sure that all relevant principals (users) exist both in the corporate directory and in the individual identity stores.

LDAP synchronization is a prerequisite to be able to use single sign-on (SSO) for collaboration clients and workstations accessing administration interfaces or the various Unified Communications services provided by the collaboration applications. SSO is implemented based on Security Assertion Markup Language (SAML) version 2.0 (SAML 2.0). SAML 2.0 authentication uses SAML authentication flows between the clients accessing the services, the collaboration applications providing these services, and an Identity Provider (IdP). The IdP is the component responsible for the actual authentication of users. The IdP can support various authentication mechanisms, including user/password based authentication against LDAP, Kerberos authentication, SmartCard based authentication, and others. The IdP can be any

IdP available on the market that complies with the SAML 2.0 specification. Cisco validates SSO with some of the IdPs such as OpenAM, Ping Federate, and Microsoft Active Directory Federated Services (ADFS).

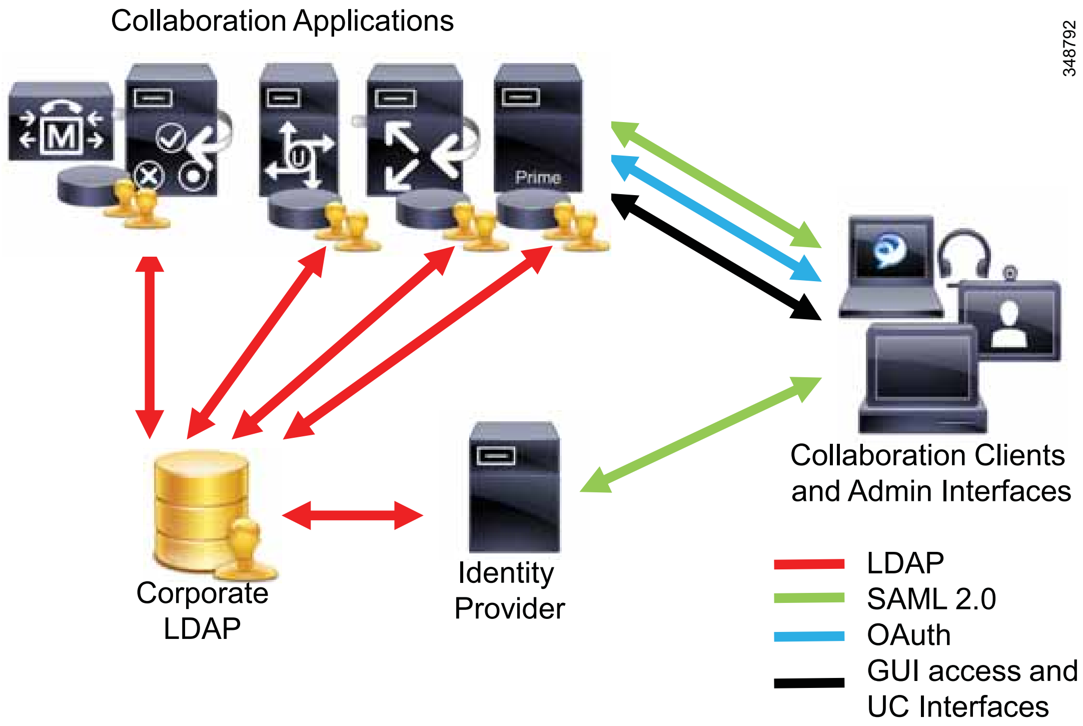
For single sign-on (SSO), authentication to Unified Communications services is delegated to the IdP through SAML 2.0. Using this mechanism, any user has to authenticate only once to any of the entities providing Unified Communications service and can then access all other Unified Communications service providers' GUIs without having to authenticate again.

SAML 2.0 only provides authentication of users and is browser based. SAML 2.0 does not address requirements for distributed authorization to use any of the UC interfaces including Unified CM UDS, Unified CM SIP, Unified CM CTI, Unified CM IM and Presence SOAP, Unified CM IM and Presence XMPP, and Unity Connection VMRest.

A centralized authorization service running on Unified CM provides authorization for UC services offered by Unified CM, Unified CM IM and Presence, and Unity Connection. This centralized authorization functionality is based on the OAuth 2.0 specification. OAuth 2.0 is an open framework for authorization providing delegated access to services on behalf of resource owners. The protocol enables authorization of clients to access resources without sharing the credentials used by the clients for authentication. OAuth essentially allows access tokens to be issued to clients by a central authorization instance. These tokens then are presented by the clients to the servers offering a service, as proof of authorization. An access token is a string value that represents the granted level and duration of access to specific resources. The access token either only represents an identifier that can be used to retrieve details of the authorization from the authorization service, or it may contain the actual details of the authorization. In the latter case, the access tokens are called *self-contained*, and the content of the access token must be signed so that the authenticity of the authorization details can be verified. Optionally, the content of self-contained tokens can also be encrypted. If the access tokens are not self-contained, then the service providers need to ask the authorization service for validation of the presented tokens to verify the authorization. In this process the content of the authorization token, as well as the mechanism used to authenticate the client and issue the token, is completely transparent for the service provider.

The authorization framework of the Cisco Collaboration solution uses self-contained access tokens. The keys used to sign and encrypt the access tokens are pushed from Unified CM to Unified CM IM and Presence, and they are pulled from Unified CM by Cisco Unity Connection and Cisco Expressway.

Figure 16-16 Identity Management Architecture


Collaboration Applications
348792
Prime
Collaboration Clients and Admin Interfaces
Corporate LDAP
Identity Provider
LDAP
SAML 2.0
OAuth
GUI access and UC Interfaces

Single Sign-On (SSO)

SSO via SAML 2.0 is an additional authentication option to the previously existing LDAP bind and local authentication. For SAML 2.0 SSO, all Unified Communications services, including the OAuth authorization service, integrate directly with the corporate identity management system using SAML 2.0.

The primary protocol used for SSO is SAML. Detailed information about SAML, such as protocol specification, use cases, and authentication flows, is openly available on the Internet. This section only introduces some key aspects of SAML.

All interactions with an Identity Provider (IdP) using SAML must be through a Web browser on the client side. If SAML authentication is to be used for clients that do not expose a Web GUI to the user, then these clients use internal WebView clients. Examples of this include Jabber softclients and collaboration endpoints supporting SSO.

Security Assertion Markup Language (SAML) is an XML data format specifically designed for the data exchange between service providers (SPs) and an IdP. SAML uses security tokens containing assertions to pass authentication related information between the IdP and the SP. The IdP in this exchange takes the role of a SAML authority, whereas the SP is a SAML consumer. Specifications of SAML can be found at

https://saml.xml.org/saml-specifications

Before SAML authentication can take place, a trust relationship between the service providers (SPs) and the Identity Provider (IdP) has to be established. This is done by exchanging metadata between the SP and IdP.

In general, a single SAML metadata instance describes either a single SAML entity or multiple entities. A SAML metadata instance describing multiply SAML instances contains a list of descriptions of single entities. Prior to Cisco Unified CM release 11.5, SAML metadata instances created by Cisco Collaboration solutions always describe only a single SAML instance.

For any SAML instance described by a SAML metadata instance, the metadata contains:

- A unique identifier
- Organization
- Expiration time for this information
- Caching period
- XML signature of this information
- Contact persons
- Unique identifier of the entity (entity ID)
- Description of SAML role of this SAML instance (identity provider, service provider, and so forth)

All pieces except the unique identifier are optional in the SAML specification and are not included in metadata created by Cisco collaboration SPs.

Each role description included in a SAML metadata instance defines the supported protocols and optionally also contains SSO key information. These keys are used later to sign SAML messages exchanged between SAML entities.

SAML metadata for a SAML service provider is required by SAML identity providers to understand the aspect of the service provider relevant for the SAML exchange between these two entities. The portion of SAML metadata specific to the service provider can indicate whether the service provider will sign SAML authentication requests and whether the service provider expects SAML assertions returned to the service provider to be signed. Also, the service provider SAML metadata defines where the authentication response should be posted. This authentication consumer service (ACS) definition basically is a URL. In addition, the service provider SAML metadata might define attributes to be exchanged between the SAML service provider and the identity provider as part of the SAML authentication process.

Similarly, identity provider metadata defines the IdP characteristics relevant for the SAML exchange between IdP and SP. IdP metadata also can define signing requirements for authentication requests and what attributes should be exchanged between IdP and SP as part of the SAML authentication process.

Detailed information about the SAML metadata format can be found at

https://saml.xml.org/saml-specifications

SAML metadata created by Cisco Collaboration SPs contains only:

- ID, entityID: Both set to the FQDN of the node or, in case of cluster-wide SSO, the FQDN of the publisher node.
- AuthnRequestSigned: **false**. This indicates that authentication requests sent by this entity will not be signed unless requested otherwise by the IdP.
- WantAssertionsSigned: **false**. This indicates that SAML assertions do not need to be signed to be accepted by this entity, but signed assertions are also acceptable.
- Encryption key and signing key: The metadata contains the node's Tomcat certificate for both keys. In the case of cluster-wide SSO, the multi-server Tomcat certificate is used, which requires the cluster to be configured to use a multi-server Tomcat certificate.

- nameIDFormat: **transient**. This indicates that name identifiers used to identify subjects in SAML assertions will be transient, which means that these identifiers cannot be used to identify a subject because the IdP will issue a new unique opaque identified the next time the same subject authenticates successfully. Instead, the authenticated subject will be identified based on the **uid** attribute returned by the IdP.
- AssertionConsumerService: One or multiple assertion consumer service definitions are included. Prior to Cisco Unified CM release 11.5, only a singe assertion consumer service definition for the HTTP-POST binding is included. Starting with Unified CM release 11.5, one assertion consumer service definition for each node and binding (HTTP-POST and HTTP-Redirect) is included. Each assertion consumer service definition specifies the binding (HTTP-POST or HTTP-Redirect) and the URL of the assertion consumer service (for example: https://ucm.example.org:8443/ssosp/saml/SSO/alias/ucm.example.org).

SAML Authentication

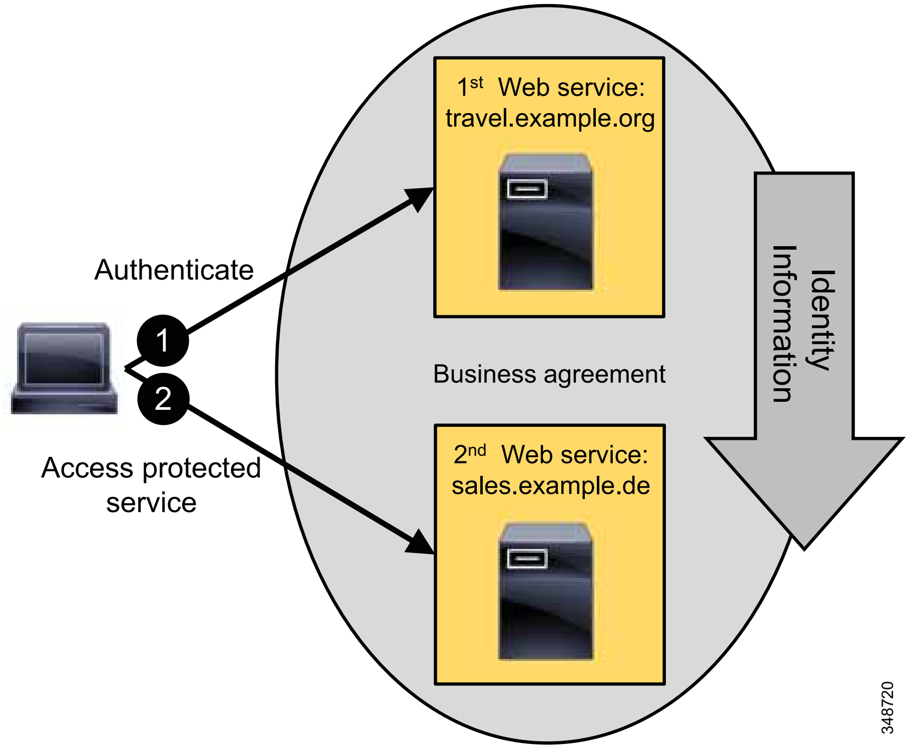
The actors in generic SAML authentication flow are:

- Client — Browser-based user client used to access the service
- SP — Application or service the user tries to access
- IdP — Entity performing the user authentication based on user credentials. The actual credentials and the actual authentication mechanism are hidden by the IdP. The IdP issues SAML assertions based on the authentication process result.

SAML defines a number of profiles to describe the use of SAML to solve typical use cases. The relevant profile used for SSO with Cisco Collaboration services is the web browser SSO profile of SAML V2.0.

The use case solved by this profile is the multi-domain web single sign-on, illustrated in Figure 16-17. In this use case, a user already has a login session with some web service (for example, travel.example.org) and is using this service. As part of the login process, a security context has been established for travel.example.org. If the same user now moves to another web service (for example, sales.example.de) and a business agreement exists between travel.example.org and sales.example.de that establishes a federated identity for the user between these services, then the user is able to access the web service sales.example.de without having to provide authentication credentials again. In this case the identity provider site (travel.example.org) asserts to the service provider site (sales.example.de) that the user is known, has been properly authenticated, and has certain identity attributes. The service provider site (sales.example.de) trusts this assertion based on the existing business agreement between the sites and grants access to the service.

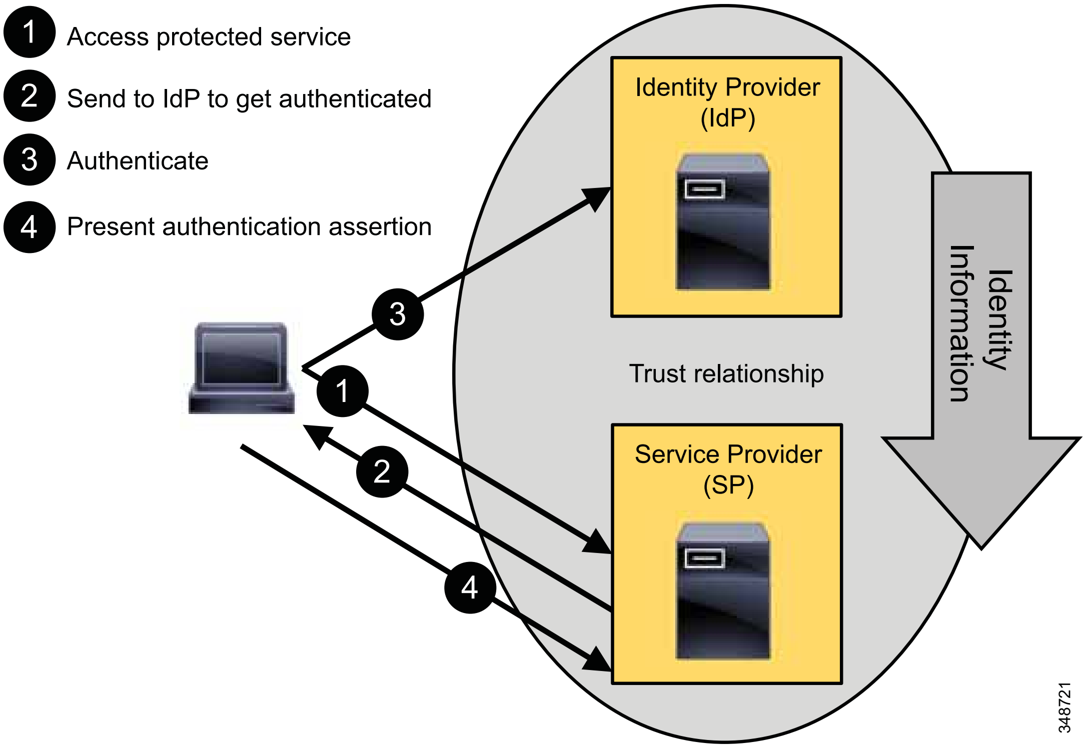
Figure 16-17 Multi-Domain Web Single Sign-On



This description implies that the user first is authenticated by a web service and that this first web service then provides an identity assertion to enable the user to access the second web service. The web service accessed first (travel.example.org) acts as the IdP for SP sales.example.de. This is known as IdP initiated web SSO.

The more typical web SSO flow used with Cisco Collaboration Services is SP initiated web SSO, illustrated in Figure 16-18. In this case the user directly (without visiting an IdP first) tries to access a protected resource on an SP. The SP sends the user to the IdP to get authenticated, and then the user presents the authentication assertion received from the IdP to the SP to get access.

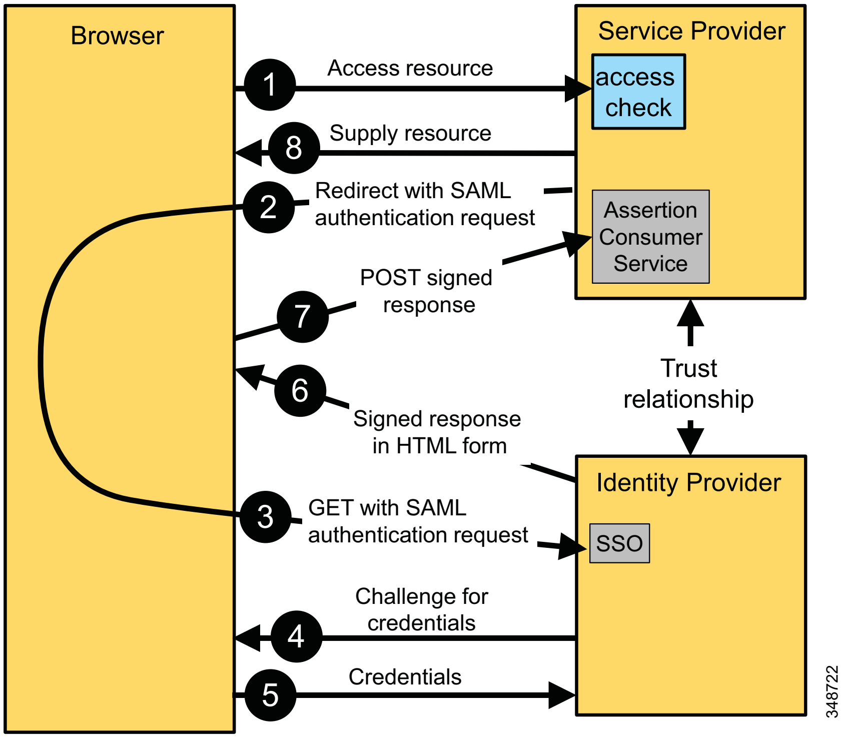
Figure 16-18 Web SSO Initiated by a Service Provider



The SAML web browser SSO profile provides a variety of options depending on whether the authentication is initiated by the IdP or SP and on how the messages are exchanged between IdP and SP. As mentioned above, Cisco Collaboration services use SP initiated SSO only where the SP sends a user to an IdP first to authenticate when the user is trying to access a protected resource and does not have an active session with the service provider. The IdP then builds an authentication assertion and sends the user back to the SP with that assertion.

The binding used for the messages exchange between IdP and SP for Cisco Collaboration services is the Redirect/POST binding, illustrated in Figure 16-19. Here an HTTP 302 redirect is used to send the SAML authentication request message from the SP to the IdP, and the authentication response from IdP to SP is sent using an HTTP POST message.

Figure 16-19 SP-Initiated SSO (Redirect/POST Binding)


Browser
Service Provider
access check
Assertion Consumer Service
1
Access resource
8
Supply resource
2
Redirect with SAML authentication request
7
POST signed response
6
Signed response in HTML form
Trust relationship
Identity Provider
SSO
3
GET with SAML authentication request
4
Challenge for credentials
5
Credentials
348722

The general steps of the SAML authentication flow are:

1. The user tries to access a service or resource by pointing the browser to the URL hosted on the application server. The browser at this moment does not have an active session with the service.
2. The SP realizes that the request originates from a client without an active session. Because HTTP is stateless, an active session can be detected by the SP only if the client sends a session cookie that has been issued by the SP earlier. Based on the SSO configuration, the SP now generates an SAML authentication request to be sent to the appropriate IdP defined as part of the SSO configuration. The SAML request contains information about the SP generating the request. This is required so that the IdP can identify the SPs sending SAML requests.

   The SP does not communicate directly with the IdP to authenticate the user. Instead the SP redirects the browser to the IdP. The URL used for this redirect is taken from the IdP metadata exchanged earlier. The SAML request to be sent to the IdP is included in the redirect as a URL query parameter using Base64 encoding.

   This redirecting HTTP 302 might look like this:

   ```
   HTTP/1.1 302 Found
   Location:
   https://pingsso.example.com:9031/idp/SSO.saml2?SAMLRequest=nZLNbtswEITveQqCd1m0pKo
   WY
   RlwYxQ1kDZK5OaQG02tYwISqXLJtH37kkra%2FBjwodflcPab3V2iGPqRr7076lv44QEdIb%2BGXiOfXm
   rqreZGoEKuxQDIneTt%2BusVz2aMj9Y4I01PL7abmmJWVCxnku07sYCqFAu2KGWVdaycV1AWRbnPPjJZl
   Dkld2BRGV3TYEPJFtHDVqMT2oUSm%2BcJq5Ks2LGK5x84K%2B8p2QQ0pYWbfh2dG5Gn6aj0A6KZHc0AM2
   MfeACYp6ob07a9nsUEGSWfjZUwJazpQfQIsWEjENUj%2FKs0z1E%2BKd0F0%2FO5908i5F92uyZprtsdJ
   WtEsJHu0mj0A9gW7KOS8P326oVXejkk4F94F0WRpyEBjmmkjdip6JXAEyldXSyjhE%2FDsq%2BWdJ5V%2
   FOWiq%2FeWy%2FSV4bP9yL8Fi%2B2mMb2Sv%2F%2FnFuK8B%2BHOq2NFdclhknJnhUYF2lHSNrH%2FjQ9
   DOCiwNT2ZA1n3vfl5aUG4sD5nPdDVU5K37CFQenrdqz8%3D&RelayState=s249030c0bda8e96a8086c
   92d0619e6446b270c463
   ```

The encoded SAML authentication request shown above can be decoded to:

```
<samlp:AuthnRequest xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    ID="s249030c0bda8e96a8086c92d0619e6446b270c463"
    Version="2.0"
    IssueInstant="2013-09-19T09:35:06Z"
    Destination="https://pingsso.example.com:9031/idp/SSO.saml2"
    ForceAuthn="false"
    IsPassive="false"
    ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    AssertionConsumerServiceURL="https://cucm-eu.example.com:
8443/ssosp/saml/SSO/alias/cucm-eu.example.com"
    >
<saml:Issuer xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion">
cucm-eu.example.com</saml:Issuer>
<samlp:NameIDPolicy xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient"
    SPNameQualifier="cucm-eu.example.com"
    AllowCreate="true"
     />
</samlp:AuthnRequest>
```

Among other details specifying authentication parameters and identifying the requesting SP, the above SAML authentication request also specifies the Assertion Consumer Service (ACS) URL. The ACS URL is the URL to which the SAML Authentication Response needs to be POSTed at the end of the authentication process.

3. The browser receives the redirect, follows the URL, and issues the corresponding GET to the IdP. The SAML request is maintained. The browser at this stage does not have an active session with the IdP.

4. After receiving the new request from a browser with no active session (browser is not sending a cookie issued by the IdP earlier), the IdP authenticates the user based on the pre-configured authentication mechanisms. Possible authentication mechanisms include user/password, PKI/CAC, or Kerberos. For user/password authentication, the IdP might push a form to the user to enter the credentials (for example, “200 OK” message with an IdP login form). For the actual authentication, the IdP might depend on back-end systems such as an LDAP server for user/password authentication.

   One key point here is that the exchange of credentials for the purpose of authentication takes place between the IdP and the browser. The SP is not involved and does not see the credentials.

5. The browser provides further information required for the authentication process. For the user/password case, this would be a POST with the information. For other authentication mechanisms, other details would need to be sent to the IdP by the browser.

6. The IdP now checks and validates the provided credentials. The check could involve interactions with respective back-end systems (LDAP bind for user/password authentication against LDAP, communication with Kerberos server to validate ticket, and so forth).

   Finally the IdP generates an SAML response for the SP. This response contains the SAML assertion documenting the result of the authentication process. The SAML assertion, in addition to the basic Yes/No information, also contains validity information and information about attributes describing the authenticated entity. At least the user ID of the authenticated entity has to be included in the well known attribute **uid** so that the SP can extract this information from the assertion to relate the authenticated entity to users existing in the local database.

   The SAML assertion is signed and potentially encrypted by the IdP according to the SSO key information published in the IdP metadata. This ensures that the SP can verify the authenticity of the SAML assertion.

The IdP returns the SAML assertion to the browser in a hidden form in a 200 OK message. The hidden form instructs the browser to POST the SAML assertion to the Assertion Consumer Service (ACS) URL of the SP.

The IdP has to establish a security context so that future authentication requests from the same browser can be answered without going through the exchange of credentials. The IdP will then realize that it already has a valid session with the browser and will assert the authentication of the previously authenticated user without prompting for credentials again. This context is established via a session cookie set on the browser by the IdP. This basically enables SSO for multiple SPs.

7. The browser follows the hidden POST received in the 200 OK message and POSTs the SAML assertion to the Assertion Consumer Service on the SP.
8. The SP extracts the SAML assertion from the POST and validates the signature of the assertion. This guarantees the authenticity of the SAML assertion and the IdP. The user identifier received in the SAML assertion in attribute **uid** as part of the attribute statement is then used to decide whether the user is authorized to access the requested service. This is based on local access control configuration on the SP. The **uid** value received in the SAML assertion has to match the Unified CM user ID of an end user authorized for the requested service. To make sure that user identifiers sent by the IdP in SAML assertions correlate to user IDs in Unified CM, SSO authentication is supported only for end users synchronized from LDAP. The assumption here is that the IdP is integrated with the same directory, so that **uid** values returned by the IdP are based on the same data source as Unified CM end user information.

   The SP grants access to the requested resource and sends back the content in a 200 OK message to the browser. The SP also sets a session cookie in the browser so that, for subsequent access requests from the same browser to the same SP, the SP does not have to initiate any more exchanges with the IdP. The IdP will be involved with additional requests from the same browser only after the SP session has expired.

Authentication Mechanisms for Web-Based Applications

When SSO is enabled for a collaboration service, any access to the respective service will be authenticated using SSO. As a fallback measure, a vanity or recovery URL also exists on the landing page. The vanity URL bypasses the SSO mechanism and provides access to all administrator GUIs. Access to the administrator GUI through the vanity URL is authenticated against the local user database. Access to the GUI through the vanity URL can be disabled on the CLI using the **utils sso recovery-url disable** command.

The vanity URL can be used as a recovery back door when there is an issue with the SAML infrastructure, such as when the IdP is unreachable or down, when there are metadata issues (for example, expired signing certificates), or when there are IdP configuration changes.

Collaboration services currently support the following user types:

- OS user

  This user is specified during installation and has access to the CLI, the Disaster Recovery System (DRS) GUI, and the OS Admin GUI. Credentials for OS users are maintained separately from credentials of other users. When enabling SSO, access to the CLI always is authenticated locally using the password stored in the local database, while access to the DRS and OS Admin GUI is authenticated via SSO and authorized against the platform database. The **set account name** command on the CLI can be used to create mappings from SSO UID values to platform users.

- Application user

  These are functional users created and managed locally. Passwords are stored in the local database. Application users are not enabled for SSO. With SSO enabled, application users can get access to only the Admin GUI through the vanity URL on the landing page.

- Local end user

  These users are created and managed locally. Passwords are stored locally. These users do not exist in the enterprise identity management system. If SSO is enabled, local end users cannot authenticate successfully. Local end users and LDAP synced users without SSO enabled are still supported.

- LDAP synced end user

  These users are managed in the corporate LDAP directory and are synchronized into the Unified Communications service through LDAP sync agreements. For every LDAP synced end user in the local database, there is a matching user in the corporate LDAP directory. If SSO is disabled, the passwords of LDAP synced end users are validated through an LDAP bind operation. With SSO enabled, the authentication of LDAP synced users is based on the authentication mechanism defined on the IdP, and authorization is based on local configuration. An LDAP synced end user has to have the proper rights assigned locally to be able to access the requested resource.

PIN-based authentication is always (even with SSO enabled) based on local configuration. Multiple collaboration services maintain individual PINs. Starting with Cisco Unified CM release 11.5, PINs can be synchronized between Unified CM and Cisco Unity Connection.

The following web services are enabled for SSO based on SAML IdP redirects:

- Cisco Unified Communications Manager Admin GUI
- Cisco Unified CM Self Care Portal
- Cisco Unified Communications Manager Serviceability GUI
- Cisco Unified Communications Manager Reporting Tool GUI
- Cisco Unified Communications Manager Platform Admin GUI
- Cisco Unified Communications Manager Disaster Recovery GUI
- Cisco Unified Communications Manager IM and Presence Admin GUI
- Cisco Unified Communications Manager IM and Presence Platform Admin GUI
- Cisco Unity Connection Admin GUI
- Cisco Unity Connection Platform Admin GUI
- Cisco Unified Personal Communicator Assistant
- Cisco Unity Connection WebInBox

SSO for Cisco Jabber

All Jabber platforms use embedded browser controls for SSO. These controls rely on underlying operating system browser technologies (see Table 16-7). Even if the embedded browser controls used by Jabber are based on the same technology as the system browser, cookies are never shared between the system browser and the embedded controls used by Jabber. This implies that Jabber always requires dedicated authentication via SSO even if the user already authenticated against the same IdP using the system browser. The only way around this is to not set up the IdP to use persistent cookies instead of session cookies. This is not considered to be the best practice because by using persistent cookies the IdP authentication state gets exposed openly and can potentially be hijacked by other applications with access to the same cookie jar.

Table 16-7 Browser Technology and Cookie Sharing

OS	Windows	Mac OS	Apple iOS	Android
Underlying Browser	Internet Explorer	Safari	WebKit or Safari	WebKit
Control shares cookies with native OS browser	No	No	No	No

For Jabber on Apple iOS, the browser selection (WebKit or Safari) is determined by the **SSO Login Behavior for iOS** enterprise parameter. Choosing Safari as the browser for SSO on Apple iOS allows access to the iOS certificate store and other protected resources to which only Apple applications have access. This selection is required if certificate-based authentication schemes should be used via SSO.

Design Considerations for SSO

SAML SSO always has to be enabled or disabled for all nodes in a cluster. Either all nodes or no nodes in a cluster are enabled for SSO. Enabling SSO through the Admin GUI automatically enables SSO for all existing nodes at the same time. As part of this process, SP metadata is downloaded to be used to establish the circle of trust between the IdP and the cluster node(s).

Prior to Cisco Unified CM release 11.5, each cluster node had to be represented on the IdP as an individual SP. If a node was added later to a cluster that already was in SSO mode, the metadata of that added node had to be imported into the IdP to complete the list of defined SPs on the IdP.

Starting with Cisco Unified CM release 11.5, SSO can be enabled in cluster-wide mode. In this case only a single metadata file needs to be exchanged with the IdP. Cluster-wide SSO can be used only if a single multi-server Tomcat certificate is used on the cluster because the SAML metadata of the cluster can contain only a single encryption and signing key. Multi-server certificates need to be CA signed certificates. When new nodes are added to a cluster enabled for cluster-wide SSO, updated metadata has to be exchanged with the IdP to make sure that the IdP is aware of the new assertion consumer service URLs of the added nodes.

SP metadata of Cisco Collaboration SPs contains assertion consumer service definitions for HTTP-Post and HTTP-Redirect bindings. These bindings must be supported by and enabled on the IdP. For SSO in cluster-wide mode, the IdP must be able to support multiple assertion consumer service definitions for a single SP.

IdP metadata has to be imported into all SAML SPs. When SSO is enabled on the Admin GUI, the IdP metadata provided in the process is automatically imported on all nodes of the cluster. If assertion signing or encryption is used by the IdP, then the signing and encryption keys must be included in the IdP metadata exchanged with the Cisco Collaboration SPs.

The SP metadata does not include the optional ContactPerson information; therefore, the IdP will not be able to expose contact information for Cisco Collaboration SPs.

SAML SP can request signed assertions from the IdP by including WantAssertionsSigned in the SAML AuthnRequest. Currently Cisco Collaboration SPs do not send this information, and the same parameter is set to **False** in the SP metadata. This gives the IdP full control over assertion signing. Cisco recommends activating SAML Assertion signing on the IdP.

If not requested otherwise by the IdP, Cisco Collaboration SPs do not encrypt or sign SAML authentication requests. This must be supported by the IdP.

Cisco Collaboration SPs request namedid-format:transient in both the metadata and the SAML authentication requests. IdPs must support this format and must be configured accordingly.

As part of a SAML assertion, the IdP in the AttributeStatement must return an attribute **uid**, and the value of this attribute must match the user ID of the respective end user in Unified CM.

Availability of the IdP is a key requirement when using SSO. The IdP has to be deployed with full redundancy and fault tolerance. Essential for this kind of deployment is that the IdP is deployed with a single logical URL and that suitable load balancers and web server farms are deployed to make sure that the single IdP URL is highly available. The single IdP URL is included in the IdP metadata and is imported into all Cisco Collaboration SPs. Failure of a single element (for example, a single web server) should be invisible to the Collaboration service.

SAML requests and assertions are signed using SP and IdP certificates. The lifetime of these certificates has to be closely monitored to make sure that the SAML SSO mechanism continues to work.

SAML assertions contain validity information (NotBefore, NotOnOrAfter). To make sure that valid assertions are not rejected due to timing issues, it is essential to synchronize all services using appropriate mechanisms such as Network Time Protocol (NTP).

Authorization Framework

For users accessing web interfaces, authentication is either based on local configuration, based on LDAP or, in the case of SSO, based on the SAML exchange between the user's browser, the web server, and the IdP. After successful authentication, the web server (for accessing the Unified CM administration GUI this would be the administration application running on Unified CM) consults the local configuration to determine whether the authenticated user is authorized to access the given resource. For example, if user Bob when authenticating via SSO provided valid credentials to the IdP and thus authenticated successfully, then Unified CM could still deny access to the Unified CM administration interface if Bob is not a member of the "Standard CCM Super Users" group. In this case Bob would get only a prompt indicating that he does not have the required privileges to access the system, instead of getting access to Unified CM administration. Authorization for access to web services such as the Unified CM administration GUI or end-user pages always is based on access levels defined on the application.

Jabber clients and other endpoints require access to a number of collaboration interfaces (for example, Unified CM SIP, Unified CM CTI, Unified CM IM and Presence SOAP, and Unity Connection VMRest). To avoid multiple authentication mechanisms (per interface), the Cisco Collaboration system uses the OAuth authorization framework for centralized authorization based on a single authentication.

OAuth 2.0

The OAuth 2.0 authorization framework is an open standard defined by the IETF OAuth working group, and the current version of the standard has been released as RFC 6749.

Whenever applications need to access multiple services on behalf of users without OAuth, separate authentications and authorizations per service are required. This creates a suboptimal user experience for end users because they have to manage multiple sets of credentials (can be partially addressed by using SSO), and it also creates a trust problem in that an end user has to share the access credentials with multiple applications.

OAuth addresses these challenges by enabling applications to obtain access to a service on behalf of an end user by orchestrating an approval interaction between the end user and the service. As part of this approval interaction the end-user, after being authenticated, instructs the OAuth authorization service to grant access tokens to the application asking for authorization. The application then presents an access token as proof of authorization when accessing services.

Access tokens have a limited lifetime, thus limiting the time an access token can be used by an application. The OAuth specification allows the OAuth authorization service to not only grant access tokens but also refresh tokens. Refresh tokens typically have a longer lifetime than access tokens, and applications can use a refresh token to obtain a new access token from the OAuth authorization service as long as the refresh token is still valid. In contrast to obtaining access tokens using the full approval procedure, exchanging a refresh token for a new access token does not require any end-user interaction and especially no end-user authentication. The concept of a refresh token allows authorization of applications to access services on behalf of end users for longer periods of time (refresh token lifetime) while still limiting the exposure to the validity period of an access token by allowing refresh token revocation. If an application at the end of the validity of the currently used access token tries to obtain a new access token, then this requires interaction with the OAuth authorization service; and if the refresh token has been revoked, then the OAuth authorization service simply refuses to issue a new access token representing continued authorization of the application.

OAuth is commonly used with various services in the Internet. Instead of building their own authorization logic into their web applications, some services delegate that to the OAuth authorization services of sites such as Facebook, Google, Twitter, or others. On the main web site the user then only has to click on the icon representing the OAuth authorization link with, for example, Facebook. The main web site (client) will then initiate the OAuth authorization flow, which in turn redirects the end user's user agent (web browser) to the authorization server (for example, Facebook). The end user then authenticates against the authorization server using their credentials, and then the authorization prompts the end user for authorization of the level of access (scopes) the client requested via the flow (for example, access to the user's email address). As soon as the end user grants access, the authorization server grants access and an access token is issued to the client requesting access. The actual process for how the client obtains the access token depends on the type of OAuth authorization flow used by the client.

OAuth Roles

The following role definitions help to explain the operation of OAuth:

- Resource owner or end user

  The owner of the protected resource. In the OAuth framework the resource owner is the entity granting access to the protected resource. This can also be referred to as the end user when the resource owner is a person.

- Resource server

  The protected resource is hosted on this server. Requests to access resources use access tokens as proof of authorization, and the resource server grants or denies access based on the access token provided in the request. Protected resources in the context of the Cisco Collaboration solution are the interfaces used by Cisco Jabber clients and endpoints, including UDS, Unity Connection VMRest, and others.

- Client

  The application making requests to protected resources on behalf of the resource owner. The client can be anything such as an application running on a desktop machine or a mobile device, a server application, or a cloud service. The term "client" only denotes the role in the context of the OAuth framework.

  Depending on the particular use case, an OAuth client may be a Cisco Collaboration service (for example, the Collaboration Edge) or an end user client (for example, Jabber). If the Collaboration Edge requests an OAuth token on behalf of a user, then the Collaboration Edge acts as an OAuth client. In the case of a Jabber client login flow inside the enterprise, the Jabber client acts as the OAuth client.

  Every OAuth client has an unique identifier, the OAuth client_id . This OAuth client_id uniquely identifies a client type. For example, Jabber for Windows and Jabber for Android use different client_ids, but all releases of Jabber for Windows use the same client_id unless concrete reasons mandate a changed client_id to enable the authorization service to differentiate between different client releases (for example, support variation in the OAuth exchange with different client releases). A set of client_ids is predefined for Cisco products and also for a third-party client.

  When requesting authorization to a protected resource, an OAuth client might request a token with a particular scope. The scope indicates the range of services that an OAuth token can be used to access.

  An OAuth Access Token is granted by the authorization service and is used by bearers (clients) for access to a protected resource. Typically access tokens are issued to a specific user and have a specific expiration time. Whenever an access token expires, the client must get a new access token.

- Authorization server

  After authentication of the resource owner and authorization by the resource owner, the authorization server issues access tokens to be used by the client.

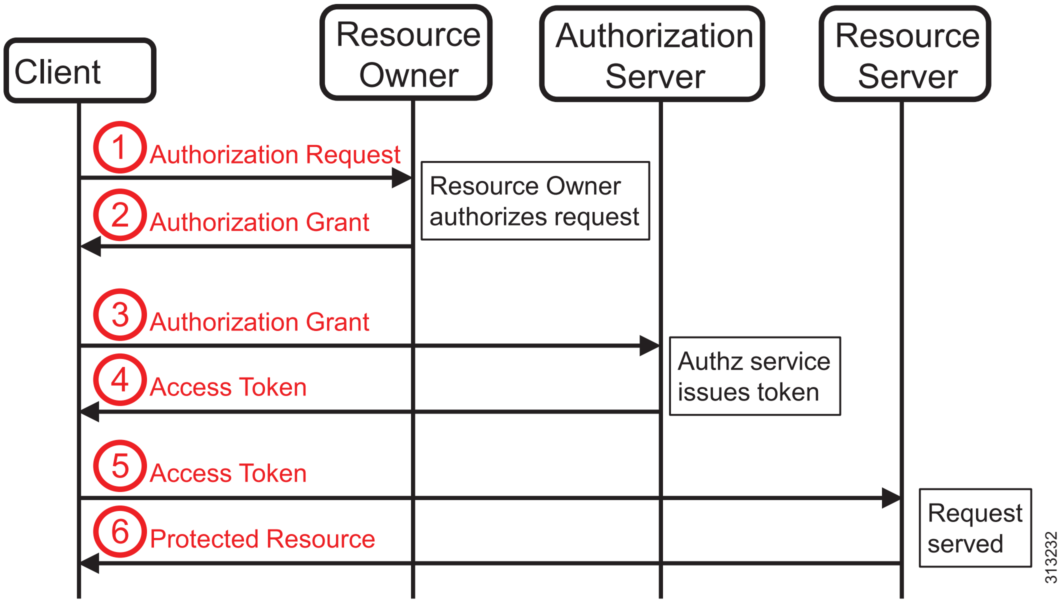
The resource server and the authorization server are not necessarily separate entities; these functions can exist on the same server. In addition, a single authorization server can issue access tokens for multiple resource servers.

In the Cisco Unified Communications architecture the authorization server is a function running on Unified CM, and the access tokens issued by the authorization server are used by Cisco Jabber clients and other endpoints to obtain access to a number of collaboration interfaces (for example, Unified CM SIP, Unified CM CTI, Unified CM IM and Presence SOAP, and Unity Connection VMRest).

General OAuth Flow

OAuth defines a number of different flows to obtain authorization, but all of them share some commonalities which are shown in Figure 16-20.

Figure 16-20 General OAuth Protocol Flow



The OAuth flow shown in Figure 16-20 illustrates the four roles and the following interactions between them:

1. The client requests authorization to access a protected resource from the resource owner (end user). Although Figure 16-20 shows this as a direct interaction between the client and the resource owner, in reality this request typically is made via the authorization server as intermediary. In this case the authorization server, after successful authentication of the resource owner, asks the resource owner to grant authorization to the client that initiated the authorization flow. The representation in Figure 16-20 is to clarify that the authorization of the client lies in the hands of the resource owner (end user).
2. The client receives the authorization grant, a representation of the resource owner's authorization. The grant can be expressed by one of four different grant types defined in the OAuth specification. The grant type depends on the method used by the client to request authorization and the types supported by the authorization server.
3. Using the authorization grant, the client can now request an access token from the authorization server. For this the client needs to authenticate and present the previously acquired authorization grant. Client authentication is required to avoid abuse of the authorization grant through untrusted intermediates.
4. The authorization server authenticates the client, validates the authorization grant, and if valid, issues an access token and (optional) a refresh token. Client authentication by the authorization server requires the client's authentication credentials to be registered with the authorization server beforehand.
5. The client can now request access to a protected resource and use the previously obtained authorization token as proof of authorization.

6. The resource server validates the access token and, if valid, serves the request. This validation can require a transaction between the resource server and the authorization server, especially if self-contained access tokens are not used.

As mentioned in the description of step 1, the preferred method to obtain an authorization grant from the authorization server is by using the authorization server as intermediary. The OAuth authorization code grant flow is an example of this procedure (see the section on Authorization Code Grant Flow, page 16-50, for more details).

Authorization Grants

As shown by the description of the general OAuth authorization flow in Figure 16-20, an authorization grant is a credential representing the authorization to access a protected resource granted by the resource owner. The authorization grant is used by the client to obtain an access token. The OAuth specification defines four grant types: authorization code, implicit, resource owner password credentials, and client credentials. For the purpose of this document only two flows are relevant:

Implicit Grant Flow

The implicit grant is a simplified authorization code grant flow. Instead of issuing an authorization code grant, the authorization server directly issues an access token to the client. This is called "implicit" because no intermediary credentials are issued.

This flow is optimized for clients implemented in a browser using scripting languages. The client is not authenticated in this flow, since the access token is issued directly. This exposes the access token to the resource owner and potentially other applications having access to the resource owner's browser.

While the implicit grant flow is more responsive (no additional transaction to exchange the authorization code grant for an access token), the implicit grant flow should be considered less secure.

Also in the context of Cisco Unified Communications solutions, where access tokens obtained via OAuth authorization flows are used by Jabber clients and other endpoints to access various system interfaces, it is important to note that to obtain a new access token when the reaching the lifetime of the previous access token with the implicit grant flow always a new authentication step is required while with the authorization code grant flow the client also obtains a refresh token which can be used to obtain a new access token directly without additional end user authentication.

SAML Bearer Assertion Grant Flow

An entity (typically a service) uses an assertion issued on behalf of an end user to get an OAuth token that is associated with the end user. A variation of this flow is used by the Collaboration Edge to get tokens on behalf of clients connecting from outside the edge to obtain authorization codes.

Authorization Code Grant Flow

This flow uses the authorization server as intermediary between the client and the resource owner, as shown in Figure 16-21.

Figure 16-21 Authorization Code Grant Flow

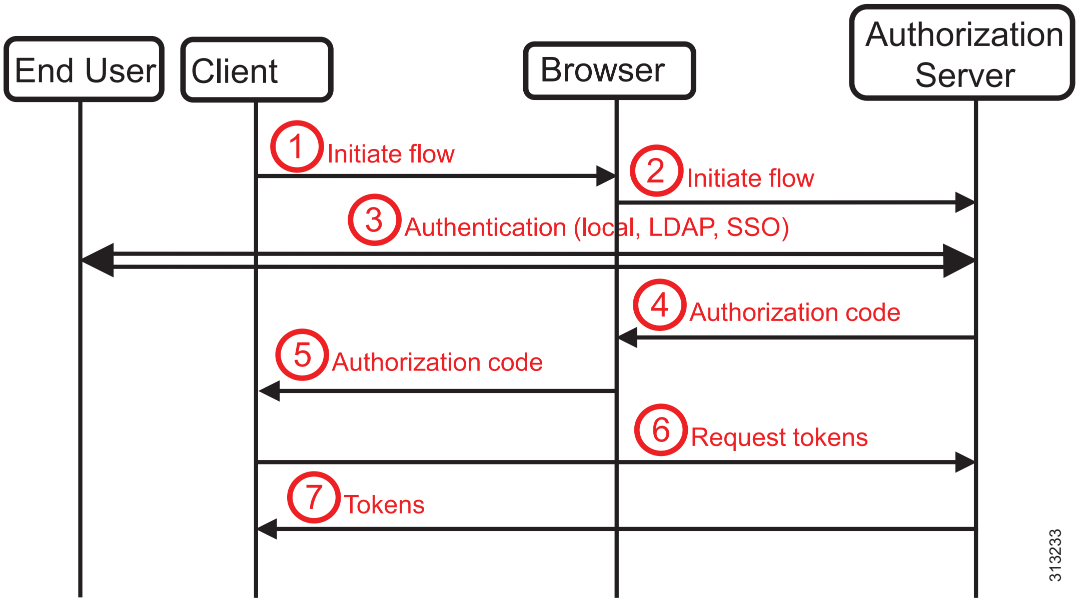


Figure 16-21 shows the details of the authorization code grant flow. The client does not directly request the authorization grant (authorization code) from the end user. Instead the client directs the end user to an authorization server which in turn later in the flow directs the end user back to the client with the authorization code. For this redirection the end user's web browser is used, and the authorization code grant flow is browser based. For the redirection of the end user’s browser in this flow, HTTP 302 redirects can be used as well as JavaScript code-based redirection in web content returned to the browser.

1. The client (for example Cisco Jabber) initiates the authorization flow by redirecting the end user’s browser to the authorization endpoint. For the Cisco Unified Communications solution this is /ssosp/oauth/authorize on Unified CM.
2. The browser accesses (GET) the endpoint on the authorization server. With this request a number of parameters are passed: the client identifier, the requested level of access (scopes), a unique request identifier, and the redirection URI to which the resulting authorization code should be posted at the end of the flow.
3. The authorization server now authenticates the end user. For authentication against the local end user table on Unified CM or using LDAP bind, this involves asking the end user to enter their username and password. For this authentication based on username and password, the authorization server returns a web form as the result to the GET on the authorization endpoint, the end user enters the credentials, and the authorization server validates the credentials either against the local end user table or against the configured LDAP server using an LDAP bind.

   If SSO is configured, the authorization server initiates a SAML 2.0 Redirect/Post flow. Posting the SAML assertion to the authorization service at the end of the Redirect/Post flow finishes the authentication step in this case.

After successful authentication, the Cisco authorization service immediately proceeds to issues the authorization code grant; the end user's authorization of the client (for example, Cisco Jabber) to use the requested resources is assumed as given.

4. To grant the authorization code to the client, the authorization server redirects the end user's browser to the redirect URI provided in steps 1 and 2. The redirection URI includes the authorization code and the request identified from step 2.
5. The browser accesses the URL and thus reveals the authorization code and request identified to the client. The request identified allows the client to correlate this event with the outstanding authorization event that triggered the flow.
6. The client now requests an access token from the authorization service using the authorization code obtained in the previous step. The token request to the authorization service is authenticated using the client's credentials (client ID and secret), and the client also again passes the redirection URI.
7. The authorization server authenticates the client, checks the redirection URL, and if valid responds back with an access token and a refresh token.

It is important to note that, because end-user authentication is obtained by the authorization server using local authentication, LDAP bind, or SSO, resource owner authentication details (for example, type of authentication and resource owner credentials) are never shared with the client. The client obtains only the authorization code grant.

Other benefits of this flow include:

- The access token is obtained by the client directly and thus is not exposed to the resource owner’s browser.
- The transaction to exchange the authorization code grant for an access token is authenticated using client credentials. The authorization code grant alone cannot be used to obtain access to the protected resource without knowledge of the client credentials.

The authorization code grant flow was introduced with Cisco Unified CM release 11.5(1) SU3. The authorization code grant flow with refresh tokens needs to be via the enterprise parameter **OAuth with Refresh Login Flow**. The default setting for this parameter is "disabled," but Cisco recommends enabling this flow. The implicit grant flow is always supported for backward compatibility even if the authorization code grant flow with refresh tokens is enabled.

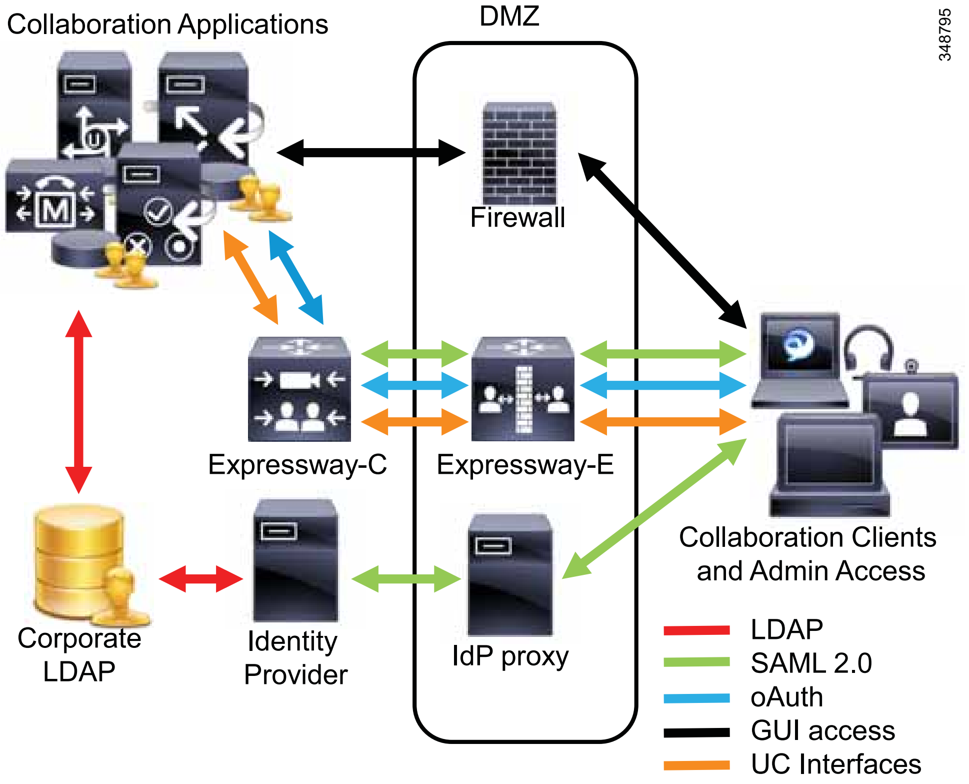
Tokens

Access tokens used in the Cisco Collaboration solution have a default lifetime of one hour (3,600 seconds). If an expired access token is used, then the resource server rejects the service and returns an error indicating that the token has expired. The client then has to obtain a new access token. To avoid this forced access token refresh, Cisco Collaboration clients initiate a token refresh after 75% of the token expiration time has elapsed. Note that access token expiration does not affect existing sessions with protocols such as XMPP and SIP, where authorization based on access tokens happens only during session establishment (for example, during SIP registration).

Mobile and Remote Access (MRA) Authentication and Authorization

To provide OAuth authorized access for clients outside the enterprise that are connected using mobile and remote access (MRA) through the Collaboration Edge, deploy Cisco Expressway-E and Expressway-C as illustrated in Figure 16-22.

Figure 16-22 OAuth with Collaboration Edge



Endpoints and collaboration clients registering through the Collaboration Edge can use OAuth to obtain access tokens by using the authorization API on Expressway-C (proxied by Expressway-E). When calling this authorization endpoint, the client needs to specify that the authorization code grant flow should be used along with the client_id, a request-specific state identifier, and the user identification. The user can be identified by username, email address, or user identifier. The authorization API on Expressway-C then will either initiate a SAML Redirect/Post authentication flow by redirecting the browser to an IdP or present a local authentication page, depending on the authentication method of the user. The authentication method for the user is determined by the MRA access control settings on Expressway-C and the home cluster settings of the user:

- The **Authentication path** setting reflects whether authentication via Unified CM LDAP, SAML SSO, or both should be allowed.
- **Authorize by OAuth token with refresh** enables the authorization code grant flow with refresh tokens.
- **Authorize by OAuth token** enables the legacy implicit grant authorization flow. This is not needed with Unified CM 12.*x* and current Jabber clients.

- **Authorize by user credential** needs to be enabled to allow MRA functionality for all IP phones and Cisco TelePresence endpoints.
- **Check for internal authentication availability** needs to be enabled only until all Unified CM clusters are migrated to a release of Unified CM that supports OAuth with refresh token and all clusters are using a common authentication (either SSO or Unified CM LDAP basic authentication). When this parameter is enabled, Expressway-C will first determine a user's home Unified CM cluster and then determine the authentication settings on the home cluster.

Depending on the required authentication method, the respective authorization flow is initiated.

MRA Sign-On with Local Authentication

Figure 16-23 shows the flow that is used to obtain an access token with authentication through Cisco Expressway based on username and password.

Figure 16-23 OAuth Authorization through Expressway with Local Authentication

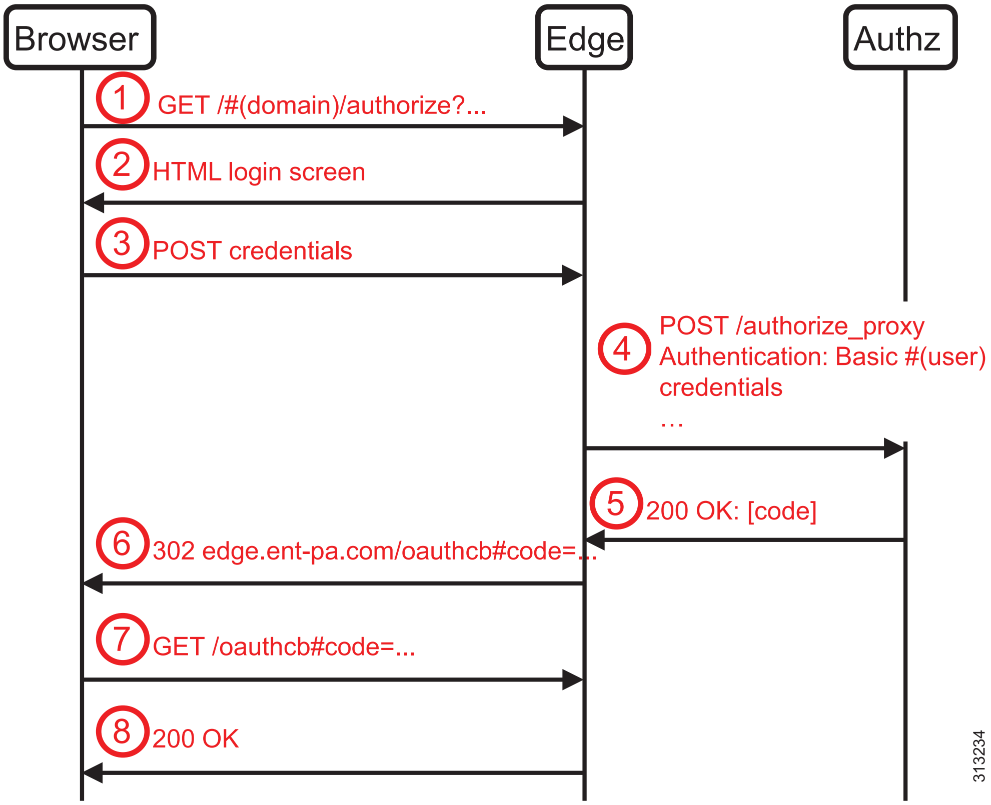


Figure 16-23 illustrates the following flow steps:

1. The Jabber client directs the web browser to access the authorization API on Expressway. The request contains state information uniquely identifying the request, the client_id, and the user identification.
2. Based on the user identification, Expressway determines that username and password authentication is required and then returns a web page with a web form to enter the user credentials. All parameters obtained in step 1 are passed through hidden input fields in the web form.

3. After the user credentials are entered, the web form is posted back to Expressway.
4. Expressway uses the /authorize_proxy endpoint on the authorization service on Unified CM to obtain the authorization code. This is a variation of a SAML bearer assertion grant flow. This request is authenticated using the username and password of an application user. The referenced application user has to have rights to access the AXL API on the authorization service on Unified CM. The /authorize_proxy request contains all authorization parameters cached earlier.
5. The authorization service validates the credentials by checking them against the local end-user table or via an LDAP bind. The authorization service then returns the authorization code to (the Collaboration) Edge.
6. Edge can then cache the code but also needs to return the code to the client. This is achieved by returning a 302 message to the client, redirecting the browser instance on the client to the OAuth callback on Edge. The target URL of the redirect contains the required information about the authorization code.
7. The browser on the client follows the redirection and accesses the OAuth callback resource on Edge.
8. The 200 OK message finishes the SSO flow through Edge. The client can now extract the authorization code from the final URL.

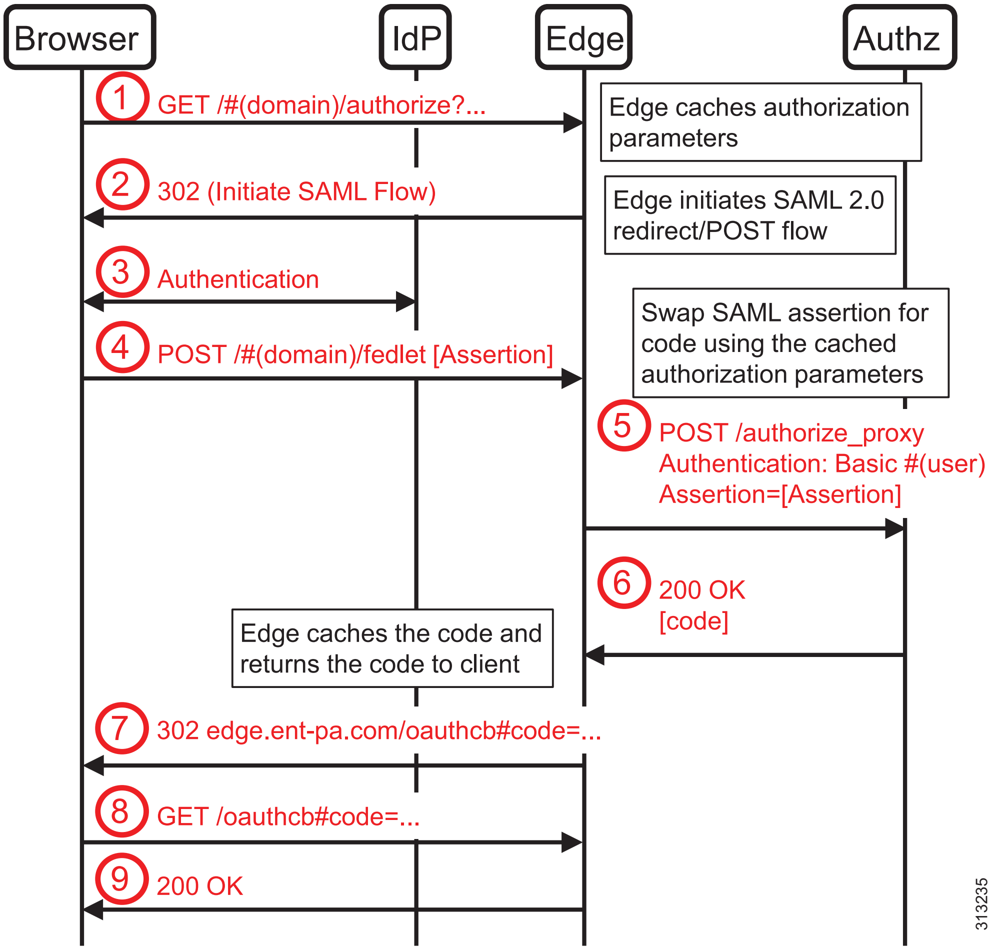
As a result, the authorization parameters are now cached on Expressway, and the Jabber client owns an authorization code.

Jabber can exchange the authorization code for an access token by again accessing the authorization API on Expressway. Expressway in this case, similar to steps 4 and 5 in the above flow, proxies the request to the access_token endpoint of the authorization service.

MRA Sign-On with SSO Authentication

This flow is very similar to the flow for authentication based on username and password. In this case Cisco Expressway uses a SAML 2.0 Redirect/Post flow to obtain authentication, and the browser on the client is redirected to a publicly accessible identity provider. This typically is an IdP proxy in the customer's DMZ, acting as proxy for the IdP deployed inside the enterprise. The IdP proxy in the DMZ essentially is only a generic HTTPS reverse proxy for the enterprise IdP. Some IdP vendors offer an option to install an IdP instance in the DMZ as an IdP proxy role. Expressway-E and Expressway-C proxy only proxy collaboration client requests for services on the collaboration applications. While the SAML authentication flow is redirected to an IdP proxy in the DMZ by making sure that the public DNS resolved the DNS name of the enterprise IdP to the public IP address of the IdP proxy in the DMZ, the OAuth exchange to achieve an OAuth token passes through Expressway-C and Expressway-E, and Expressway-E requests the OAuth token as a proxy for the actual client. This is a variant of an OAuth SAML bearer grant flow as shown in Figure 16-24.

Figure 16-24 OAuth Authorization through Expressway with SSO Authentication


Browser
IdP
Edge
Authz
1 GET /#(domain)/authorize?...
Edge caches authorization parameters
2 302 (Initiate SAML Flow)
Edge initiates SAML 2.0 redirect/POST flow
3 Authentication
4 POST /#(domain)/fedlet [Assertion]
Swap SAML assertion for code using the cached authorization parameters
5 POST /authorize_proxy
Authentication: Basic #(user)
Assertion=[Assertion]
6 200 OK
[code]
Edge caches the code and returns the code to client
7 302 edge.ent-pa.com/oauthcb#code=...
8 GET /oauthcb#code=...
9 200 OK
313235

1. To acquire an OAuth token, the browser sends an HTTP GET request to the /authorize endpoint on Edge. The /authorize endpoint on Edge is accessed using prefix encoding to refer to the customer domain. Edge in this description refers to a Cisco Expressway-C and Expressway-E pair implementing the Collaboration Edge.
2. Expressway initiates an SP-initiated SAML 2.0 redirect/POST authentication flow by returning a 302 response redirecting the browser to the IdP. Edge also caches the authorization parameters from the client request because they are needed later in the actual OAuth proxy request.
3. Browser and IdP then exchange the messages required to authenticate the user. The message exchange depends on the authentication method configured on the IdP.
4. If the SAML authentication succeeds, then as the last step of the SAML exchange the browser POSTs the SAML assertion to the assertion consumer service on Edge. Edge still needs to exchange this SAML assertion for an authorization code.
5. To achieve this, Edge uses the /authorize_proxy endpoint on the authorization service. This request is authenticated using the username and password of an application user. The referenced application user has to have rights to access the AXL API on the authorization service on Unified CM. The /authorize_proxy request contains all authorization parameters cached earlier.

6. The authorization service then can check whether the authenticated end user has the required privileges. If the authenticated user is authorized to access the requested service, then the authorization service issues an authorization code and returns that code in a 200 OK message.
7. Edge can then cache the code and still needs to return the code to the client. This is achieved by returning a 302 message to the client, redirecting the browser instance on the client to the OAuth callback on Edge. The target URL of the redirect contains the required information about the authorization code.
8. The browser on the client follows the redirection and accesses the OAuth callback resource on Edge.
9. The 200 OK message finishes the SSO flow through Edge. The client can then extract the authorization code from the final URL.

As a result, the authorization parameters are cached on Expressway, and the Jabber client owns an authorization code.

Jabber can exchange the authorization code for an access token by again accessing the authorization API on Expressway. Expressway in this case, similar to the flow above, proxies the request to the access_token endpoint of the authorization service.

Understanding OAuth Tokens

This sections covers access and refresh tokens, token expiration, and token management.

Access Tokens

Access tokens are issued to clients by the authentication service. The content of the access token is opaque to the client. Clients do not need to understand the semantics of access tokens; they are treated as an arbitrary string value by the client. Access tokens are used only to authorize requests sent to services by the clients.

Access tokens issued by the authorization service on Cisco Unified CM are self-contained tokens. These self-contained tokens can be validated by Unified Communications services without contacting the authorization service. The access tokens are JSON Web Tokens as described in RFC 7519, and they contain information about the authorized user, the expiration, and the authorized scopes. The access tokens are encrypted and digitally signed by the authorization service.

Because Unified Communications services check authorization based on self-contained access tokens without contacting the authorization service, there is no way to centrally revoke previously authorized access within the validity period of an access-token. To limit the exposure, self-contained access tokens typically have a short lifetime. The default lifetime of access tokens issued by the authorization service on Cisco Unified CM is 60 minutes, and it can be set to values between 1 minute and 1440 minutes (one day).

Jabber typically tries to obtain a new access token as soon as the current access token has 25% lifetime left; thus, a token with a lifetime of 60 minutes will be refreshed after 45 minutes.

Refresh Tokens

Refresh tokens are issued to clients by the authorization service on Cisco Unified CM. Clients can present a refresh token to the authorization service on Unified CM to obtain an access token within the lifetime of the refresh token. This does not require end-user authentication, so that obtaining a new access token is a very fast transaction that can be executed without impacting the user experience. Again, the content of a refresh token is opaque to the client. The request to obtain a new access token is authenticated using the client credentials (client ID, client secret, and redirect URI).

The default lifetime of refresh tokens issued by the authorization service on Unified CM is 60 days, and can be set to values between 1 day and 1825 days (five years).

Note Changing the refresh token lifetime invalidates all refresh tokens currently issued by the authorization service, thereby requiring all users to re-authenticate to obtain new refresh tokens.

A new full authorization flow is required to obtain a new refresh token. This requires end-user authentication. Jabber clients will advise end users when refresh tokens are about to expire, and a new authorization flow can then be initiated by the end user.

Administrators can revoke all refresh tokens for a user by means of the **https://<*unifed_cm*>:8443/ssosp/token/revoke?user_id=<*uid*>** endpoint on Unified CM. Here *unified_cm* needs to be replaced with the IP address of hostname of the Unified CM publisher and *uid* needs to be replaced with the user id of the user whose refresh tokens are to be revoked. The request needs to be authenticated using the credentials of an administrator user.

Token Signing and Encryption Keys

The self-contained access tokens are encrypted and digitally signed by the authorization service on Cisco Unified CM. The required keys are created on the Unified CM publisher node and are distributed across all the nodes of the cluster. While Unified CM IM and Presence obtains the keys via intra-cluster replication, Cisco Expressway and Unity Connection need to pull the keys from Unified CM to enable access token validation. Access to the keys is obtained via a token key API on Unified CM. Access to this API requires authentication using credentials of an application user with AXL access. On Cisco Unity Connection, authorization servers are defined in the **Authz Server** section in the **System Setting** tab in Cisco Unity Connection Administration.

Signing and encryption keys can be regenerated if the administrator believes that the keys have been compromised. Regenerating either of these keys invalidates all access tokens issued by the authorization service, so that all clients need to obtain new tokens leading to re-authentication of all end users.

Signing keys can be regenerated using the **set key regen authz signing** CLI command. Encryption keys can be regenerated using the **set key regen authz encryption** CLI command. Information about the current signing and encryption keys can be displayed using the **show key authz signing** and **show key authz encryption** CLI commands.

Scopes

Access tokens contain a scope element. The scope defines the Unified Communications services that the holder of the access token is authorized to use. The scopes for access tokens issued for any given user are defined by setting the **Jabber Desktop Client Policy** and **Jabber Mobile Client Policy** under **Mobile and Remote Access Policy** in the user profile configuration on Cisco Unified CM. This allows the administrator to define different scopes for Jabber desktop and Jabber mobile clients. Possible values are **No Service**, **IM&Presence only**, and **IM&Presence, Voice and Video calls**.

The scope is checked by Expressway whenever a client establishes a connection and Expressway will only establish connections to authorized services.