[MS-CASO]:
Certification Authority System Overview

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: http://www.microsoft.com/interop/osp) or the Community Promise (available here: http://www.microsoft.com/interop/cp/default.mspx). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **Trademarks.** The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications do not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments you are free to take advantage of them. Certain Open Specifications are intended for use in conjunction with publicly available standard specifications and network programming art, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

This document provides an overview of the Certification Authority System Overview Protocol Family. It is intended for use in conjunction with the Microsoft Protocol Technical Documents, publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts. It assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A Protocol Family System Document does not require the use of Microsoft programming tools or programming environments in order to implement the Protocols in the System. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

Abstract

The Microsoft Certification Authority (CA) System uses public key cryptography to issue certificates that can be used for a variety of purposes, including encryption and authentication. This document describes the intended functionality of the CA System and how the protocols in this system interact. It provides examples of some of the common usage scenarios. It does not restate the processing rules and other details that are specific for each protocol. These details are described in the protocol specifications for each of the protocols and data structures that make up this system.

Revision Summary

Date	Revision History	Revision Class	Comments
06/20/2008	0.1	Major	Updated and revised the technical content.
07/25/2008	0.1.1	Editorial	Revised and edited the technical content.
08/29/2008	0.1.2	Editorial	Revised and edited the technical content.
10/24/2008	0.1.3	Editorial	Revised and edited the technical content.
12/05/2008	0.1.4	Editorial	Revised and edited the technical content.
01/16/2009	0.1.5	Editorial	Revised and edited the technical content.
02/27/2009	0.1.6	Editorial	Revised and edited the technical content.
04/10/2009	0.1.7	Editorial	Revised and edited the technical content.
05/22/2009	1.0	Major	Conversion to new template.
07/02/2009	1.0.1	Editorial	Revised and edited the technical content.
08/14/2009	1.0.2	Editorial	Revised and edited the technical content.
09/25/2009	1.1	Minor	Updated the technical content.
11/06/2009	2.0	Major	Updated and revised the technical content.
12/18/2009	2.0.1	Editorial	Revised and edited the technical content.
01/29/2010	3.0	Major	Updated and revised the technical content.
03/12/2010	3.0.1	Editorial	Revised and edited the technical content.

Date	Revision History	Revision Class	Comments
04/23/2010	3.0.2	Editorial	Revised and edited the technical content.
06/04/2010	3.0.3	Editorial	Revised and edited the technical content.
07/16/2010	3.1	Minor	Clarified the meaning of the technical content.
08/27/2010	3.2	Minor	Clarified the meaning of the technical content.
10/08/2010	4.0	Major	Significantly changed the technical content.
11/19/2010	5.0	Major	Significantly changed the technical content.
01/07/2011	5.0	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	6.0	Major	Significantly changed the technical content.

Contents

1 Introduction

This Protocol Family System Document (PFSD) is primarily intended to cover the Protocol Family as a whole. In conjunction with Member Protocol Technical Documents (TDs), which are intended to cover Member Protocols, it presents the rules for information exchange relevant to those Member Protocols and the Protocol Family that are used to interoperate or communicate with a Windows operating system in its various environments.

A **certification authority (CA)** is an entity that issues **digital certificates**. The Certification Authority System Overview ([MS-CASO]) discusses the protocols used for submitting **certificate** requests to the CA, for CA server side processing of these requests and for remote administration of the CA. The certificates themselves do not generally contain sensitive information and are often publicly available. The certificates can be used for a variety of different purposes and are typically stored in a variety of methods and locations. For example, a smart card may contain certificates and **keys** that can be used for logging on to multiple systems. Other certificates may be stored within a **directory** and retrieved by those wishing to use it for encrypting content intended for the certificate owner.

There are other protocols which may interact with the **CA system** for the purpose of obtaining certificates. Some of these protocols may involve certificate storage and some may not. For example, the Health Certificate Enrollment Protocol Specification ([MS-HCEP]) discusses the process of **enrolling** certificates, by the health registration authority (HRA) server, on behalf of its **clients**. These certificates are not stored on the HRA server but are distributed to its clients using a protocol that is not part of the CA system. Other protocols, such as the Encrypting File System Remote (EFSRPC) Protocol, also interact with the CA system for the purpose of obtaining certificates. These certificates are also enrolled by the server. However, in this case, the certificates obtained are then stored locally on that server. In either case these protocols, and the applications or services that utilize them, act in the role of a consumer of the CA system. The CA system itself does not include certificate usage or storage. Any implementation of certificate usage and storage are up to the consumers.

This document is written for those interested in developing, implementing, or managing a CA system. It describes the functional architecture of the CA system and how the protocols in this system interact. This document provides an overview for the implementation of a CA system.

1.1 Glossary

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

Active Directory
attribute
certificate
certificate authority (CA) certification authority
certificate revocation lists (CRL)
certificate services
certificate template
client
digital certificate
digital signature
directory
Distributed Component Object Model (DCOM)
encryption
enhanced key usage (EKU)

enroll/enrollment
end entity
enterprise certificate authority
exchange certificate
interface
key
key archival
key exchange
key recovery agent (KRA)
keyholder
Lightweight Directory Access Protocol (LDAP)
object
object identifier (OID)
private key
public key
public key infrastructure (PKI)
registration authority (RA)
remote procedure call (RPC)
revocation
root CA
schema
standalone CA
trust

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

CA exit algorithm
CA policy algorithm
Cryptographic Message Syntax (CMS)
Enroll On Behalf Of
issuance
key length
key pair

The following terms are specific to this document:

CA Administrator: A human operator who is responsible for managing the CA System.

CA System: Certification authority system that implements the protocols and data structures specified in [MS-WCCE], [MS-CSRA], [MS-CRTD], and [MS-ICPR].

Component: The principal computational elements and data stores that execute in a system.

Entity: A unit that is part of the system such as a component or an element.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as described in [RFC2119]. Note that in [RFC2119] terms, most of these specifications should be imperative, to ensure interoperability. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

Any specification that does not explicitly use one of these terms is mandatory, exactly as if it used MUST.

1.2 References

This section lists normative and informative references relevant to the Certification Authority System.

1.2.1 Normative References

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

[MS-ADSO] Microsoft Corporation, "Active Directory System Overview", July 2009.

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification", June 2007.

[MS-AUTHSO] Microsoft Corporation, "Windows Authentication Services System Overview", August 2009.

[MS-CAESO] Microsoft Corporation, "Certificate Autoenrollment System Overview", August 2009.

[MS-CSRA] Microsoft Corporation, "Certificate Services Remote Administration Protocol Specification", March 2007.

[MS-CRTD] Microsoft Corporation, "Certificate Templates Structure", March 2007.

[MS-DCOM] Microsoft Corporation, "Distributed Component Object Model (DCOM) Remote Protocol Specification", March 2007.

[MS-DISO] Microsoft Corporation, "Domain Interactions System Overview", November 2009.

[MS-ICPR] Microsoft Corporation, "ICertPassage Remote Protocol Specification", August 2007.

[MS-RPCE] Microsoft Corporation, "Remote Procedure Call Protocol Extensions", January 2007.

[MS-WCCE] Microsoft Corporation, "Windows Client Certificate Enrollment Protocol Specification", June 2007.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC5280] Cooper, D., Santesson, S., Farrell, S., et al., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, May 2008, http://www.ietf.org/rfc/rfc5280.txt

1.2.2 Informative References

[HOWARD] Howard, M., "Writing Secure Code", Microsoft Press, 2002, ISBN: 0735617228.

[MS-DRDM] Microsoft Corporation, "Directory Replication and Data Management (DRDM) Remote Protocol Specification", July 2007.

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MSDN-LDAPSync] Microsoft Corporation, "Example Code for Receiving Change Notifications", June 2007, http://msdn.microsoft.com/en-us/library/ms676877.aspx

[MSFT-ARCHIVE] Microsoft Corporation, "Key Archival and Management in Windows Server 2003", December 2004, http://technet2.microsoft.com/WindowsServer/en/Library/296f87df-06c3-4e27-89ff-5283cb76fb811033.mspx

2 Overview

Section 1, "Introduction", describes this Protocol Family System Document. This section introduces the system that is being documented.

2.1 System Summary

A certification authority (CA) issues certificates and confirms to other entities that the certificate is valid. People, computers, and applications, collectively described as end entities within this document, can all be issued certificates from the CA. Certificate holders can use the private key to digitally encrypt data, to digitally sign documents, and to identify themselves.

A public key infrastructure (PKI) supports public key cryptography within and between organizations. A PKI consists of digital certificates, key pairs, CAs, and other registration authorities. A certificate is a digital statement issued by a CA that vouches for the identity of the certificate holder; a certificate binds a public key and a collection of **attributes** to the certificate holder. The certificate can be freely shared with other entities.

An entity requests a new certificate or a renewal of an existing certificate from the CA. Policy will normally define whether a CA automatically issues the certificate or queues the request for manual review by a CA Administrator. The CA typically requires some sort of authentication before processing the request. The CA could support different policies for each kind of certificate. For example, it might automatically issue certificates to be used for signing and encrypting email messages but only allow smartcard authentication certificates to be issued by CA administrators who have visually verified the user's identity.

Policy-controlling certificate issuance can be restricted in two ways; the administrator decides to control certificate issuance either manually or automatically. Under the manual policy algorithm, the administrator would typically approve or deny each request in the queue. When certificate templates are used, the requestor is granted a certificate if the requestor has enroll permissions on the corresponding template. The template can also specify additional constraints around the issuance of certificates.

While certificates are not required for normal client or server functionality in an out-of-the-box installation, there are a variety of systems or components that might utilize or rely on digital certificates for their operation, depending upon their configuration. In situations where other systems or components use certificates, there is no requirement that these certificates be provided through the implementation of the system specified in this document. In some cases, there might be other systems or components that will attempt to interact directly with this system, if available, as described in section 4.3.2, in order to obtain certificates.

2.2 List of Member Protocols

This section describes the protocols and data structures used by the CA System. The CA System implements the following protocols:

Windows Client Certificate Enrollment Protocol, as defined in [MS-WCCE]. This protocol is responsible for certificate enrollment. It is based on DCOM, as specified in [MS-DCOM], and allows clients to request various services from a CA, such as certificate enrollment and property retrieval.

Certificate Services Remote Administration Protocol, as defined in [MS-CSRA]. This protocol is responsible for CA administration. It is based on DCOM and allows administrative tools to configure the state and policy of a CA on a server.

ICertPassage Remote Protocol, as defined in [MS-ICPR]. This protocol is a subset of the WCCE Protocol used for certificate enrollment over RPC by clients that do not support DCOM. The CA MAY implement the server role specified in [MS-ICPR] section 3.2 to support clients that do not use DCOM to communicate with the server<1>.

In addition to the protocols listed above, the CA system also uses the following data structure:

Certificate Templates Structure, as defined in [MS-CRTD]. Certificate templates are stored in the **Active Directory** and are used when the CA operates as an enterprise CA. They contain details about requesting and issuing certificates. Policy algorithms on the CA use certificate templates to determine how to respond to certificate requests. [MS-CRTD] defines attributes that are accessed by using the Lightweight Directory Access Protocol (LDAP).

2.3 Relevant Standards

Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, as specified in [RFC5280]. This specification is one part of a family of standards for the X.509 Public Key Infrastructure (PKI) for the Internet. This specification profiles the format and semantics of certificates and certificate revocation lists (CRLs) for the Internet PKI.

3 Foundation

This section describes the theoretical and practical information needed to understand this document and this system.

3.1 Background Knowledge and System-Specific Concepts

This section summarizes:

- Background knowledge required to understand this document.
- Concepts that are specific to this system.

The reader would benefit by being familiar with the concept of public key cryptography. Public key cryptography allows one entity to prove its identity to another and exchange encrypted information without having to exchange private encryption keys. In this form of cryptography, an entity has a key pair consisting of a private key and a public key. The public key is freely exchanged with other parties. The public key can be used to encrypt information to be sent to the owner of the key pair, and the key-pair owner can use the private key to decrypt the information. The owner of the key pair can also use the private key to digitally sign documents; anyone else can use the public key to verify that the signature is authentic.

End entities will normally evaluate certificates for validity when making trust decisions and will no longer trust the certificate if it is presented after the expiration date. In order to invalidate a previously issued certificate, prior to its expiration, an administrator can revoke it. This might be desired, for example, when an employee leaves the organization or when the private key has been compromised. The CA maintains a list of revoked certificates that it makes available publicly at a location specified in all of the certificates it issues. This list is known as the certificate revocation list (CRL). Entities that want to verify the validity of a certificate can download the CRL and determine if the certificate is on it.

3.2 System Purposes

In modern day computing, there is an ever increasing interest and focus on security. Cryptography plays an important role in providing that security. Some security mechanisms utilize PKI and digital certificates as part of their cryptographic security. The mission of the CA System is to issue these certificates to end entities for purposes such as digital signature, encryption, and authentication.

3.3 System Use Cases

3.3.1 Stakeholders and Interests Summary

The primary actors involved with the CA System include:

End Entity: An End Entity is a keyholder (person or computer) to whose key or name a particular certificate refers.

Enrollment Agent: An entity that submits requests on behalf of some other End Entity. An Enrollment Agent is typically authorized by the CA to enroll for certificates that End Entities themselves might not be able to. The policy enforcement for those certified End Entities is thus assumed to be done by the Enrollment Agent.

CA Administrators: A human operator who is responsible for management of the CA System, such as system configuration, and managing pending requests for certificates.

Additional stakeholders and their interests in this system are as follows:

Enterprise architects: The designers of the solutions who want to evaluate this system for use in their environments. They want to ensure the system is robust and reliable and that it will interoperate effectively with their existing information technology infrastructure.

Developers of a CA: These are the protocol implementers. They want to understand how the system interacts with client computers requesting services such as certificate enrollment or revocation.

Policy and Exit Algorithm developers: These are the people who customize CA behavior by creating policy and exit algorithms. They are interested in understanding how flexible and capable the system can be.

Developers of the enrollment clients: These developers create client applications that request certificates over the network. They need to understand how the system interacts with clients and what protocols are used for various operations.

Developers of the CA administration tools: These developers build remote administration tools for system management. They need to understand how the system communicates with management computers.

Testers: These are the people testing any of the solutions previously described. They want to understand the protocols and interfaces used, and the common use cases in order to develop effective test plans.

3.3.2 Supporting Actors and System Interests Summary

When the CA System is operating in enterprise mode, there are additional supporting actors involved with the system.

Active Directory Domain Services (AD DS): Active Directory Domain Services, as specified in [MS-ADTS], provides certificate template storage.

Domain Interactions System: The Domain Interactions System, as specified in [MS-DISO], provides a common infrastructure to provide identity, authentication, and authorization services to the CA System.

Authentication System: The Authentication System, as specified in [MS-AUTHSO], provides authentication services to secure communications in the CA System, and authentication services that support the client to server communication within and outside the system.

Domain Client: When running in enterprise mode, the CA System implements the role of the server as a domain client, as discussed in [MS-DISO].

3.3.3 Use Case Diagrams

There are two main use cases for the CA System:

- Enroll for a Certificate
- Administer the CA

The Enroll for a Certificate use case is the most important use case for this system. In its simplest form, it allows a caller (either an End Entity or Enrollment Agent) to request a certificate from a CA (see the examples in sections 6.1.1 and 6.1.2). Upon successful completion of the use case, the End Entity receives a certificate signed by the CA.

Common variations of the certificate enrollment use case are as follows:

- Certificate renewal is when an End Entity already possesses a valid certificate and uses the private key associated with that certificate to sign a renewal request for a new certificate of the same type (see the example in section 6.1.4).
- Enrollment on behalf of another user introduces an Enrollment Agent who acts as a cosigner for the certificate request to provide a higher level of control in the enrollment process (see the example in section 6.1.4).
- Certificate enrollment with CA Administrator approval interrupts the automatic flow of the certificate enrollment to allow the administrator to modify the request itself, modify the resulting certificate, or approve or deny the request (see the example in section 6.1.3).

The Administer the CA use cases include generic functions such as editing the CA configuration, as well as more specific functions such as revoking certificates or recovering escrowed private keys from a CA.

The primary specific CA administration use cases are:

- Edit CA configuration settings
- Revoke a certificate
- Recover an archived certificate and key

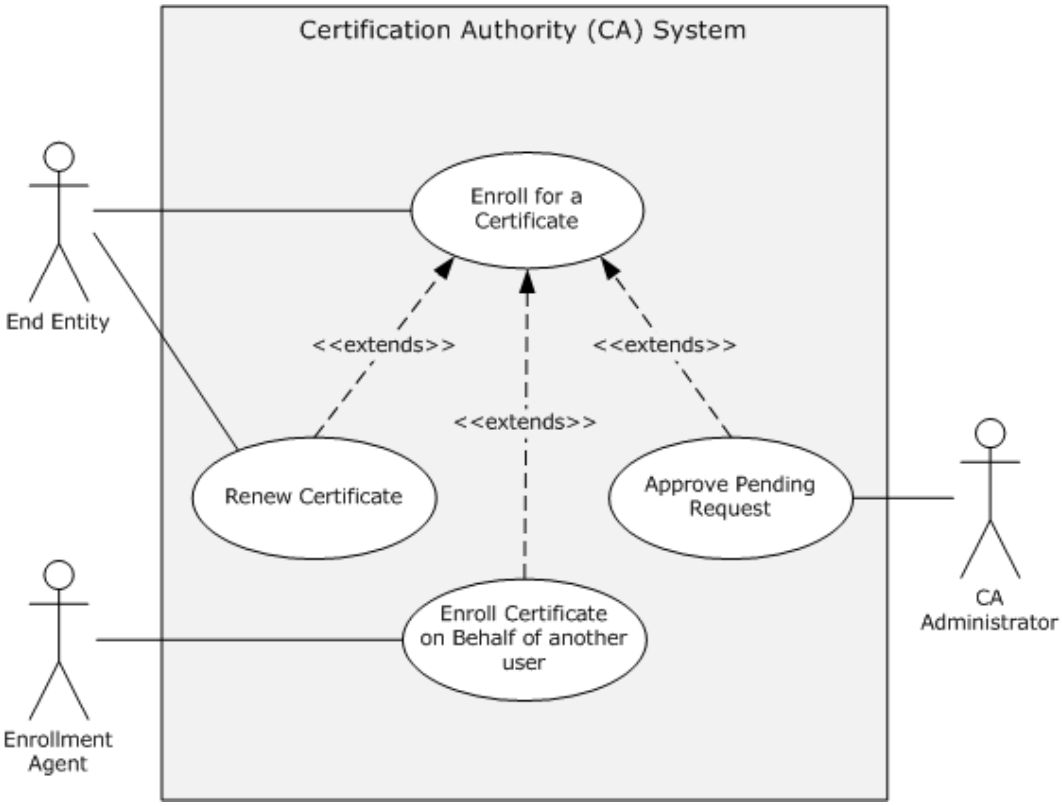

Figure 1: Enroll for a Certificate use case

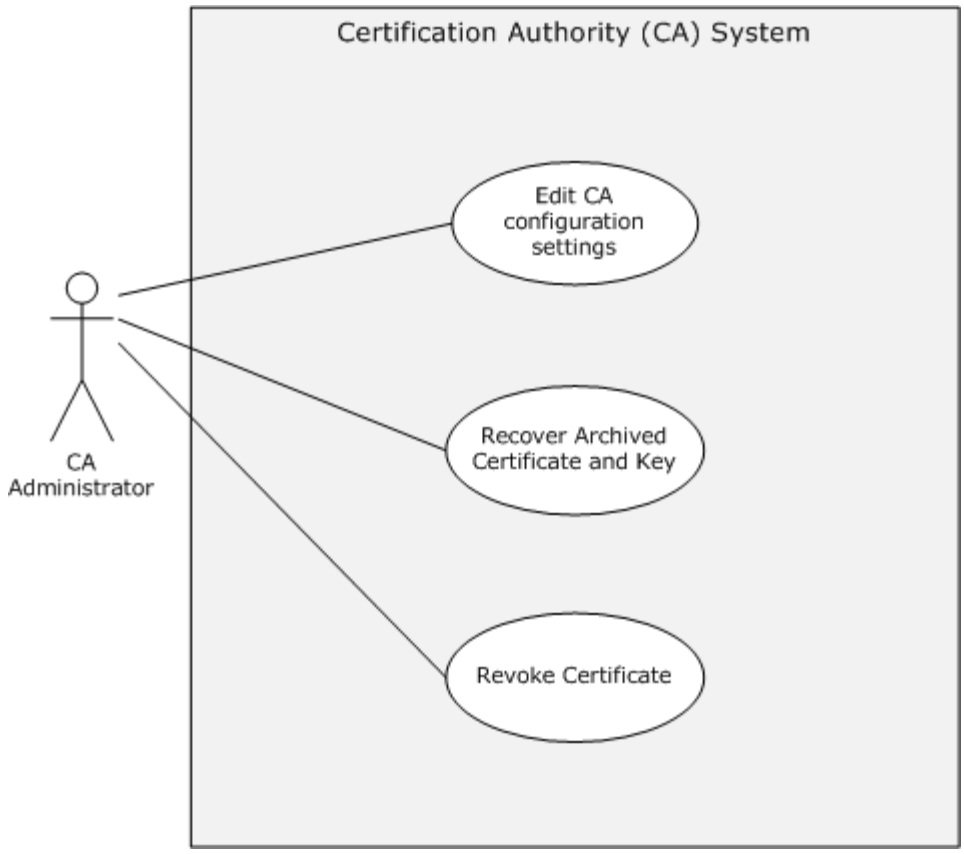

Figure 2: Administer the CA use cases

3.3.4 Use Case Descriptions

3.3.4.1 Enroll for a Certificate – End Entity

Goal: To enroll for a certificate. The goal of this use case is for the End Entity to be issued a certificate (see examples 6.1.1 through 6.1.4).

Context of Use: An End Entity can use a certificate for any number of different reasons and scenarios. When a certificate is desired, a caller generates a certificate request and submits a certificate request to the CA as specified in [MS-WCCE]. The certificate enrollment can either be a new enrollment or a renewal. In the renewal case, an existing certificate is used to sign a request for a new certificate of the same type before being submitted to the CA. Depending upon the scenario, the caller might be an Enrollment Agent, rather than the End Entity. In the Enroll on Behalf Of case, a certificate request is signed by an Enrollment Agent before being submitted to the CA.

Direct Actor: The direct actor of this use case is the End Entity.

Primary Actors: The primary actors of this use case are the same as the direct actor with the possible inclusion of an Enrollment Agent.

Supporting Actors: The CA Administrator could be a supporting actor in this use case.

Stakeholders and Interests:

- End Entity for submitting a certificate request to the CA.
- Enrollment Agent for submitting a certificate request to the CA on behalf of the End Entity.
- CA Administrator for approving pending certificate requests for issuance by the CA.

Preconditions: The End Entity, and possibly the Enrollment Agent and CA Administrator, require access to the CA.

Minimal Guarantees: The CA System guarantees that it will process all certificate requests it receives, according to the policy algorithm.

Success Guarantees:

- The CA System guarantees that it will be able to issue certificates when permitted by its policy algorithm.
- The CA System guarantees that issuance decisions are determined by its policy algorithm

Trigger: The certificate enrollment process is triggered when the CA receives a certificate request.

Main Success Scenario:

1. When the trigger occurs, the CA makes a decision on whether the certificate can be issued based on its policy.
2. The CA constructs a certificate based on the certificate request and its policy.
3. The CA signs the certificate and returns it to the client.
4. Extensions:
    - 1a. Depending upon the configuration of the system, a CA Administrator might be involved in the certificate enrollment decision process. When the certificate request is held in a pending state by the CA, it requires CA Administrator approval before issuance, as specified in [MS-CSRA]. In the case of a request requiring administrator approval, the CA will hold the request in a pending state until a CA Administrator approves the request. Once approved, the certificate will be issued.

3.3.4.2 Edit CA Configuration Settings – CA Administrator

Goal: To edit configuration settings on the CA. The goal of this use case is for the CA Administrator to be able to define and edit various configuration settings on the CA that affect behavior and policy around the issuance of certificates.

Context of Use: When a CA server is put into service, there is a variety of configuration settings that need to be defined by the CA Administrator in order for the CA operation to be in line with the requirements and desires of the enterprise or organization that has deployed it. In order to be able to define and edit these configuration settings and CA properties, the CA Administrator often needs to be able to administer the CA remotely and is able to do so by using the interfaces defined within [MS-CSRA].

Direct Actor: The direct actor is the CA Administrator.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: There are no supporting actors in this use case.

Stakeholders and Interests:

- CA Administrator for ensuring CA is configured and operating as desired.
- End Entities for assurance that the CA will be able to issue certificates as expected.

Preconditions: The CA Administrator requires access to the CA.

Minimal Guarantees: The CA System guarantees that the CA Administrator is able to define configuration settings on the CA that will affect CA operational and certificate issuance behavior.

Success Guarantees:

- The CA System guarantees that configuration settings made by the CA Administrator will be maintained.
- The CA System guarantees that End Entities will be able to obtain certificates when requested in accordance with defined policy and configuration settings.

Trigger: The CA Administrator triggers all CA administration operations.

Main Success Scenario:

1. When the trigger occurs, the CA responds to connection attempts from the CA Administrator.
2. The CA Administrator then defines or edits configuration settings or CA properties as desired.

Extensions: None.

3.3.4.3 Revoke a Certificate – CA Administrator

Goal: To revoke a previously issued certificate and to publish a list of revoked certificates (see the example in section 6.1.6).

Context of Use: In the event it is desired to invalidate a previously issued certificate for any number of different reasons, such as a compromise, the CA Administrator can revoke the certificate and include this certificate within a CRL that can be referenced by any entity consuming the certificate and attempting to validate it. Many circumstances can warrant revocation, for example when a key pair is compromised.

Direct Actor: The direct actor is the CA Administrator.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: There are no supporting actors in this use case.

Stakeholders and Interests:

- The CA Administrator for ensuring the certificate is revoked and a new CRL is published.
- The End Entity for assurance that revoked certificates referencing them are no longer valid.
- Other application and system administrators that might rely upon or use the End Entity's certificate for a variety of purposes, for assurance that certificates are valid for their intended purpose.

Preconditions:

- The CA has previously issued a certificate.
- The CA Administrator can provide the serial number of the certificate to be revoked.

Minimal Guarantees: The CA System guarantees that a certificate can be revoked and that the revoked certificate can be added to a CRL.

Success Guarantees: The CA system guarantees that a certificate is revoked and added to a CRL.

Trigger: The CA Administrator requests a certificate revocation.

Main Success Scenario:

1. When the trigger occurs, the CA revokes the certificate.
2. The CA Administrator then invokes the CA to create and publish the CRL so the revoked status can be discovered by interested parties.

Extensions: None.

Upon successful completion of the use case, the certificate is revoked and a new CRL is published with the latest information on the status of the certificate.

3.3.4.4 Recover Archived Certificate and Key – CA Administrator

Goal: To recover a certificate and its private key that have been archived within the CA database (see the example in section 6.1.5).

Context of Use: Key archival and recovery is typically used in encryption scenarios. In the event an encryption certificate and/or its private key are unavailable for decryption, the ability to recover them will provide the ability to decrypt data that was encrypted using the certificate and its public key.

Key archival behavior is defined by a flag set within the certificate template as specified in [MS-CRTD] section 2.27. When an enterprise CA issues a certificate based on a template with the key archival flag, the issued certificate and its corresponding private key are archived within the CA database as detailed in [MS-WCCE] section 1.3.2.1.

Direct Actor: The direct actor is the CA Administrator.

Primary Actor: The primary actor is the same as the direct actor.

Supporting Actors: There are no supporting actors in this use case.

Stakeholders and Interests:

- The CA Administrator for ensuring access to the CA and archived key material.
- The End Entity is interested in order to maintain the ability to decrypt previously encrypted data.

Preconditions:

- The CA has been configured with one or more key recovery agent (KRA) certificates.
- An archived certificate has been issued by the CA and the archived material exists within the CA's database.
- The CA Administrator has access to any required KRA certificates and their private keys.

Minimal Guarantees: The CA System guarantees that archived certificates and private keys can be retrieved from the CA database.

Success Guarantees:

- The CA System guarantees that archived certificates and private keys can be retrieved from the CA database.
- The CA System guarantees that archived material retrieved from the CA database can be decrypted using the associated KRA certificates and private keys.

Trigger: The CA Administrator triggers the recovery operation.

Main Success Scenario:

1. When the trigger occurs, archived certificate and key information is retrieved from the CA database.
2. Once retrieved, this information is then decrypted using the associated KRA certificates and private keys.
3. The recovered certificate and private key are then available to be restored to the End Entity for use.

Extensions: None.

4 System Context

This section describes the relationship between this system and its environment.

4.1 System Environment

The system does not require any specific environment in which to run. It is capable of operating in a fully disconnected, independent state, or, in an integrated fashion with network communication to the Active Directory and enrolling systems. A PKI deployment could include CAs that are offline as well as those connected to a network. The environment in which the CA runs will typically be determined by its intended usage as well as the security requirements around its operation.

A CA running in Enterprise mode will require interaction with Active Directory as a repository for certificate templates and will require authentication of the end entities that interact with the system.

4.2 System Assumptions and Preconditions

The system has the following assumption, regardless of mode:

- The transport protocols, RPC and DCOM, MUST be available if the CA is to be accessed over a network.

Additional assumptions that apply when running in enterprise CA mode are:

- Active Directory MUST be available for the storage and retrieval of certificate templates.
- The system depends on DCOM to authenticate its clients.

The requirements and specifications for Active Directory communication are detailed within the Domain Interaction System Overview [MS-DISO].

4.3 System Relationships

The following sections list the systems that use the interfaces provided by this system of protocols and other systems that this system depends on.

4.3.1 Black Box Relationship Diagram

End entities use their client computers to connect to external interfaces provided by the system to request new certificates, to renew certificates, and to obtain information about the CA. Administrators use their client computers to connect to external interfaces to manage the CA remotely.

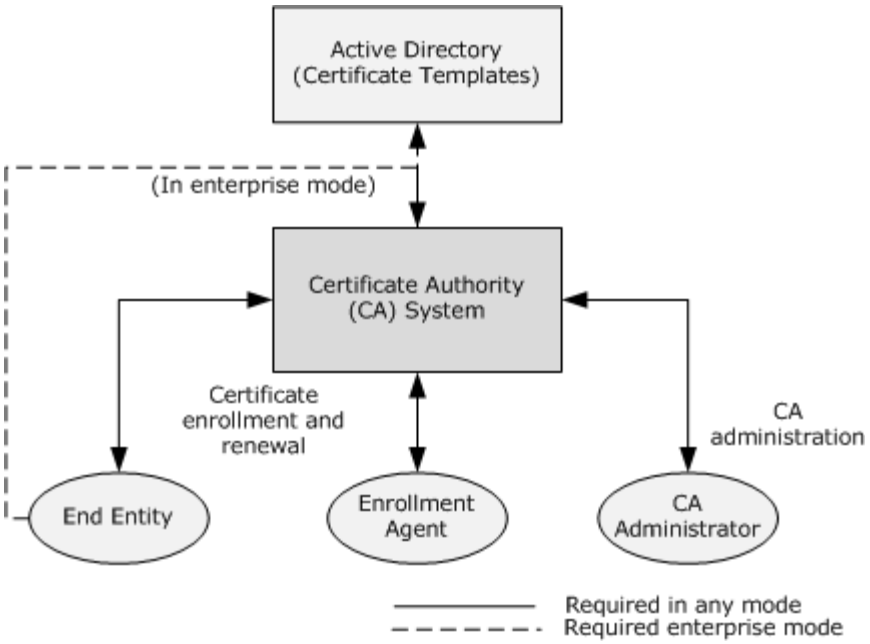

Figure 3: CA System external interfaces

4.3.2 System Dependencies

This system depends on the Distributed Component Object Model (DCOM) protocol specified in [MS-DCOM] and on the Remote Procedure Call (RPC) protocol specified in [MS-RPCE].

Any system implementing the enterprise CA server mode, as described in [MS-WCCE] section 3.2, has some additional dependencies. An enterprise CA MUST use certificate templates for its policy and it MUST use the LDAP profile, as specified in [MS-ADTS] section 3.1.1.3, for accessing the certificate templates, as specified in [MS-WCCE] section 3.2.2. The format of the structures stored in the directory and used by the CA is specified in [MS-CRTD].

No other systems or components directly depend on the system specified in this document. Other systems or components might use certificates. The use of the system described here is one way of many to obtain certificates, however, there is no requirement that these certificates be provided through the implementation of this system.

Most systems or components that use certificates will typically expect those certificates, and their corresponding keys, to be available for use and will not necessarily be actively involved in any attempts to obtain those certificates. Some systems or components MAY interact directly with the CA in order to obtain needed certificates. If the required certificates are not present, these systems or components MAY attempt to request a certificate from a CA, using the protocol specified in [MS-WCCE].

Domain Controllers, Network Access Protection and Encrypting File System are examples of systems or components that might attempt to interact directly with a Microsoft CA in order to enroll for needed certificates, depending upon system configuration. Default behavior for domain controllers is to attempt to enroll for a Domain Controller certificate from an enterprise CA that is configured to issue that specific certificate template. Encrypting File System will attempt to enroll for a Basic EFS certificate from an enterprise CA that is configured to issue that specific certificate template. This certificate template is the default for this behavior but it can be configured to enroll for other certificate templates if desired.<2>

Some systems or components MAY choose to generate and use a self signed certificate if they do not have a required certificate and are unable to obtain one through direct interaction with this system. Encrypting File System is an example of a system or component that can be configured to provide its own, self signed certificate if a required certificate is not available and it is unable to obtain one from an enterprise CA.<3>

4.3.3 System Influences

When the system operates as an enterprise CA, it interacts with Active Directory for the storage and retrieval of certificate templates and other objects, as specified in [MS-WCCE] section 3.2.2. The protocols and processes for accessing Active Directory are specified in the Active Directory System Overview [MS-ADSO] and the Domain Interaction System Overview [MS-DISO].

The system specified in this document does not have any specific restrictions or requirements on the protocols and processes that are used to interact with Active Directory.

4.4 System Applicability

All client systems use a WCCE component for certificate enrollment. There are a variety of WCCE client types and their behavior regarding the handling of certificate requests and the resulting issued certificate, could be different. The common WCCE client types include:

- Autoenrollment
- User Enrollment Tools
- Registration Authority Applications
- Direct Enrollment Applications

Autoenrollment is normally performed without user input as specified in [MS-CAESO]. In some cases, user input during the enrollment process might be required, such as in the case of smart card usage and PIN input. However, the enrollment process itself is not triggered by the user. Issued certificates are received from the CA and are then stored within a local certificate store on the client system.

User Enrollment Tools will typically work in a similar fashion, but are user initiated and could involve further user input during the enrollment process.

Registration Authority (RA) Applications are typically used in situations where a higher level of assurance is required for the certificate enrollment process. Often, their use involves an enrollment agent. Most RA applications are used in order to process or submit certificate requests for other users and therefore do not keep or install issued certificates for their own use. Instead, the issued certificates are transferred to the end entity in some fashion.

Direct Enrollment Applications are those that might be written by third parties to interact directly with the CA for certificate enrollment. How these applications handle the certificates after they have been issued is completely dependent upon the design and development of the application.

In all cases, the involvement of the CA system ends with the CA providing the issued certificate to the WCCE client. How the client handles the certificate after that point is independent of the CA.

4.5 System Versioning and Capability Negotiation

Three aspects of the system have multiple versions.

4.5.1 Interface Versions

There are multiple versions of the interfaces specified in [MS-WCCE] and [MS-CSRA]. The versioning rules for those interfaces are defined in section 1.7 of [MS-WCCE] and [MS-CSRA].

4.5.2 Client and Server Modes

The CA can operate in one of two modes: as a standalone CA or as an enterprise CA. The standalone CA is specified in [MS-WCCE] section 3.2.1 and the enterprise CA in [MS-WCCE] section 3.2.2. On client computers, these correspond to the basic enrollment mode as specified in [MS-WCCE] section 3.1.1 and the enrollment based on certificate templates mode as specified in [MS-WCCE] section 3.1.2.

4.5.3 Certificate Template Versions

Certificate templates have three different versions as specified in [MS-CRTD] section 2.16. The processing rules for the client and server for each version of the certificate templates are specified in [MS-WCCE] sections 3.1.2 and 3.2.2.

4.6 System Vendor-Extensible Fields

[MS-WCCE] section 1.8 lists all vendor extensible fields.

5 System Architecture

This section describes the basic structure of the system and the interrelationships among its parts, consumers, and dependencies.

The CA System implements the Windows Client Certificate Enrollment Protocol (WCCE), the ICertPassage Remote Protocol (ICPR), and the Certificate Services Remote Administration Protocol (CSRA) to enable certificate enrollment and CA administration. It also makes use of policy and exit algorithms to facilitate more complex enrollment processes.

This section describes the functional architecture of the CA System. The architecture is presented in an implementation-independent way, in order to describe what is essential for interoperability and leave a maximum of design freedom for the implementation of components of the architecture.

5.1 Abstract Data Model

The Abstract Data Model (ADM) is a union of the ADMs defined in [MS-WCCE] and [MS-CSRA]. There are several areas where coherency is important for the CA system:

- Access to the same type of information: When multiple clients attempt to perform operations that affect the same tables of the ADM (for example, submitting a new request that gets recorded in the Request table), the implementation MUST provide the record level coherency for that table.
- Access to the same information: When multiple clients (possibly clients of different protocols) access that same datum of the ADM, the implementation MUST provide the datum level coherency for that table.

5.1.1 Server ADM

Implementations of the WCCE, ICPR, and CSRA protocols within the server role share the ADM fields as described in the following sections.

5.1.1.1 Request Element Tables

- Request Table: Specified in [MS-WCCE] section 3.2.1.1.1 and [MS-CSRA] section 3.1.1.1.
- Attribute Table: Specified in [MS-CSRA] section 3.1.1.2.
- Extension Table: Specified in [MS-CSRA] section 3.1.1.3.

The preceding tables are all related to certificate request processing and their purpose is to hold the history and data associated with all requests processed by the CA. All the tables are commonly maintained by the CA and their data is persistent across all states of the CA, including system and server restarts, as well as backup and recoveries.

5.1.1.2 CRL Table

- Certificate Revocation List (CRL) Table: Specified in [MS-CSRA] section 3.1.1.4 and [MS-WCCE] section 3.2.1.1.3.

The purpose of this table is to store certificate revocation information. This data is persistent across all states of the CA. It is typically accessed and manipulated by CA Administrators via the CSRA protocol.

5.1.1.3 Configuration Data

- Configuration Data: ADM elements that are common are specified in [MS-CSRA] sections 3.1.1.6, 3.1.1.7, 3.1.1.8, 3.1.1.9, and 3.1.1.10 and in [MS-WCCE] section 3.2.1.1.4.

The purpose of the configuration data elements is to provide information pertaining to the operational behavior of the CA. This data is persistent across all states of the CA. It is most commonly accessed by CA Administrators using CSRA, but can also be read by enrollment clients using WCCE.

Server implementations of the Certificate Services Remote Administration Protocol, the Windows Client Certificate Enrollment Protocol, and the ICertPassage Remote Protocol in combination use the same list of configuration data elements across the implementations.

5.1.1.4 Certificate Template Table

- Certificate Template Replica Table: Specified in [MS-WCCE] section 3.2.2.3.1.

The CA maintains a replica of all certificate template data stored in Active Directory. This data is accessed by the policy algorithm and is used in processing enrollment requests. The data is persistent across all states of the CA.

5.1.2 Client ADM

Implementations of the WCCE, ICPR, and CSRA protocols by the client share the ADM fields described in [MS-WCCE] sections 3.1.1.1 and 3.1.2.1 and in [MS-CSRA] section 3.2.1.

5.2 White Box Relationships

The CA consists of two distinct groups. One group of components is responsible for the certificate enrollment and the other for CA system administration. The two groups communicate through shared Abstract Data Model (ADM) elements as specified in section 5.1 and the interaction between them is defined in [MS-WCCE] and [MS-CSRA].

The protocols and their interaction with the shared ADM elements are shown in the following diagrams.

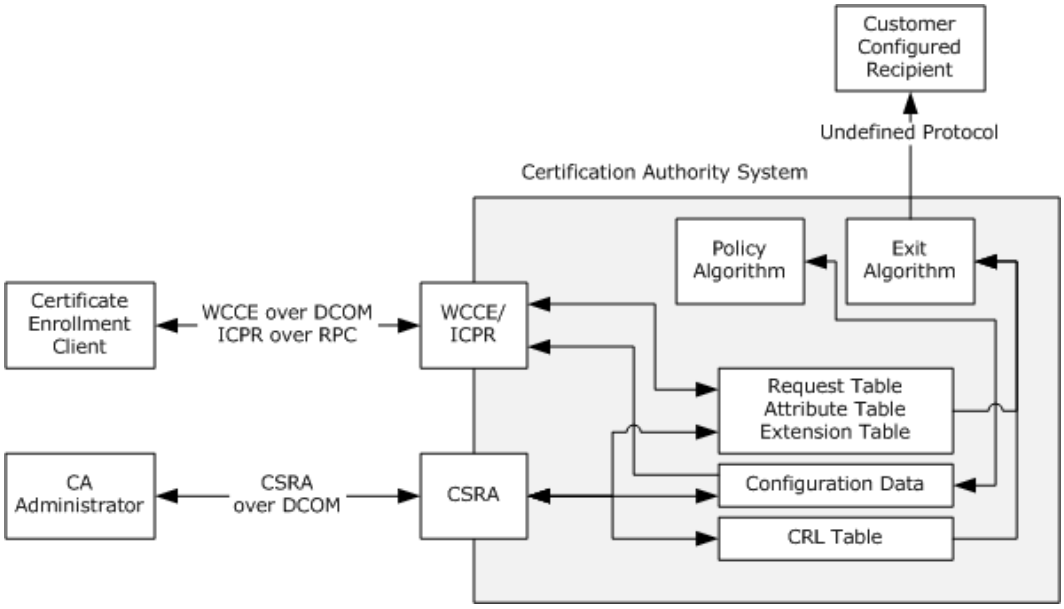

Figure 4: Standalone CA System

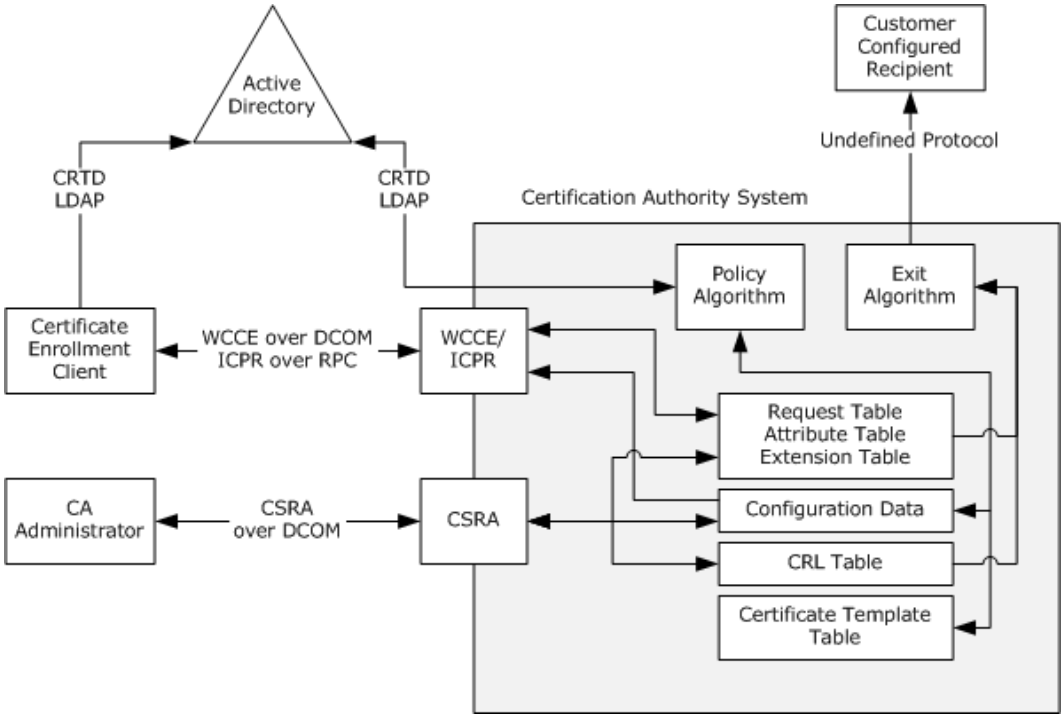

Figure 5: Enterprise CA System

5.3 Member Protocol Functional Relationships

5.3.1 Member Protocol Roles

The protocols implemented by the CA System are as follows:

- Windows Client Certificate Enrollment Protocol (WCCE) is the protocol that is responsible for certificate enrollment. It is based on DCOM and allows a client to interact with a CA to request a certificate and to query for information about the CA. The CA MUST follow instructions in [MS-WCCE] section 3.2 to implement the server role of the protocol. Details of the protocol are documented in [MS-WCCE].
- ICertPassage Remote Protocol (ICPR) is the protocol for the certificate enrollment over RPC for clients that do not support DCOM. It allows a client to interact with a CA to request a certificate. Details of the protocol are documented in [MS-ICPR]. The CA MAY implement the server role specified in [MS-ICPR] section 3.2 to support clients that do not use DCOM to communicate with the server.
- Certificate Services Remote Administration Protocol (CSRA) is the CA administration protocol. It is based on DCOM and allows remote management of the CA when compatible tools are used. Details of the protocol are documented in [MS-CSRA]. The CA SHOULD implement the server role of the CSRA (see [MS-CSRA] section 3.1) to enable scenarios where CSRA is involved. The Certificate Templates Data Structure (CRTD), while not a protocol, is the schema for certificate templates that are stored in Active Directory. When operating in enterprise mode, the CA requires that all certificates issued are based upon certificate templates. Certain settings on the certificate template participate in policy decisions made by the CA when evaluating a certificate request. Other settings simply define details to be contained within issued certificates or act as suggestions for clients to follow when creating certificate requests. Details of the schema are documented in [MS-CRTD].

5.3.2 Member Protocol Groups

There are scenarios that involve both the Windows Client Certificate Enrollment Protocol and Certificate Services Remote Administration Protocol. Processing rules for both protocols are documented in [MS-WCCE] and [MS-CSRA], respectively. However, the state transition of a single request and its corresponding certificate within the system can be of interest because both protocols are utilized.

The following figure illustrates the major logical states of a single request and its corresponding certificate in the Request table of the ADM defined in [MS-WCCE] section 3.2.1.1.1.

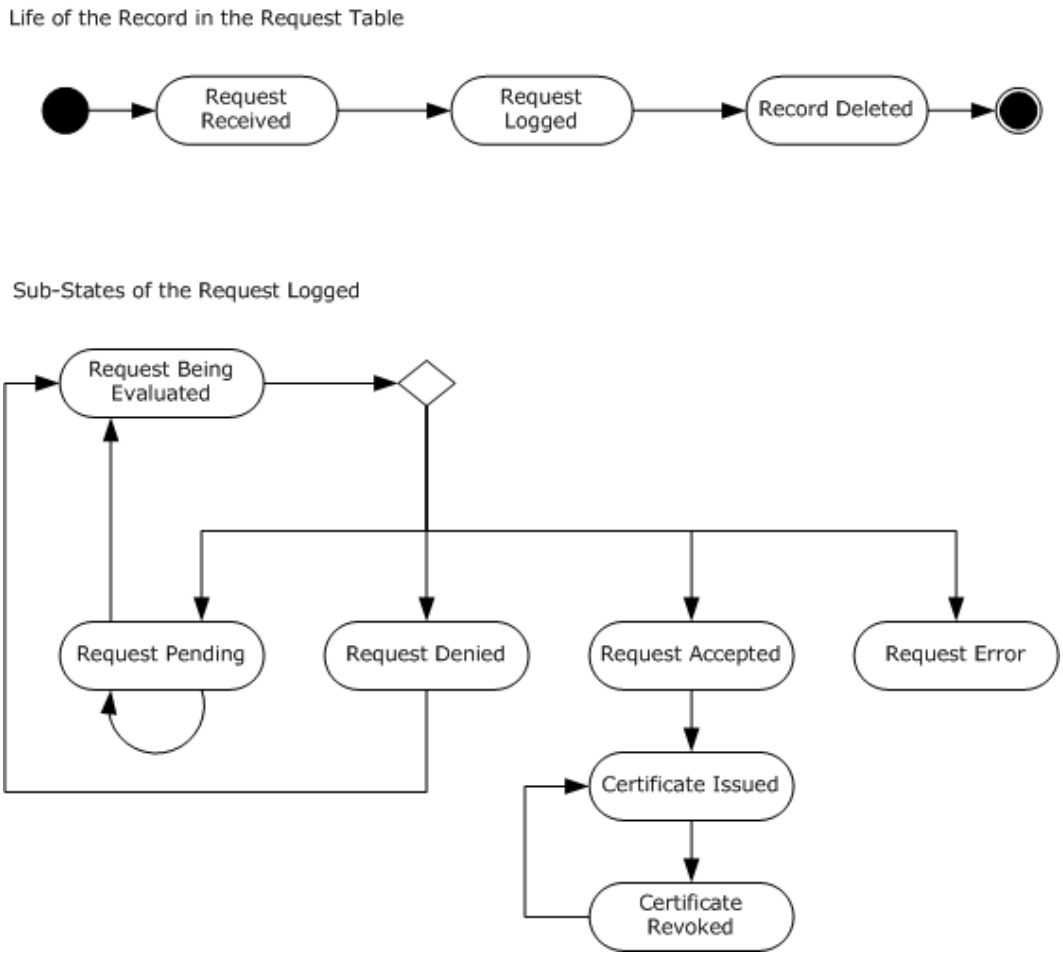

Figure 6: Logical states of a certificate request

The preceding figure does not represent all of the possible permutations of values in a single row. In the context of the state transition diagram the flow is as follows:

- Request Received

  CA receives a request from a caller.

- Request Logged

  The CA SHOULD record all requests it receives unless some error is encountered during processing (see the Request method sections of [MS-WCCE]). It SHOULD store those requests in the Request table of the ADM defined in [MS-WCCE] section 3.2.1.1.1. <4>

  This state has multiple sub-states that can all transition to the Record Deleted state directly.

  - Request Being Evaluated

    Once the request is received, the CA MUST evaluate it by invoking its policy algorithm (see [MS-WCCE] sections 3.2.1.4.2.1 and 3.2.2.6.2.1), which will decide if the certificate is to be set to pending (Request Pending state), rejected (Request Denied state), or issued (Request

Accepted state). In the case where there is an error during request processing, and the request was already logged, the request goes into Request Error state.

- Request Denied

  In the case where a certificate request is denied by policy algorithm, the request remains in the database. The CA Administrator can resubmit the request on the behalf of the caller that originally requested the certificate. When the request is resubmitted, it MUST be evaluated just like a new request (Request Being Evaluated state).

- Request Pending

  When the policy algorithm sets the status of the request to pending, the CA allows the CA Administrator to modify the request by adding extensions and/or attributes (see the SetAttributes and SetExtension methods of [MS-CSRA]). However, the CA MUST NOT automatically change the status of the request or consider it resubmitted if those methods are called. Resubmission has to be done as a separate step by the CA Administrator.

  When resubmission is done as specified in section ResubmitRequest of [MS-CSRA], the request is evaluated as a new request (Request Being Evaluated state). In this case, if current policy required CA Administrator approval for that specific type of certificate, that requirement would be satisfied since the CA Administrator is the one who has resubmitted the request.

- Request Accepted

  Once the request is accepted, the CA MUST issue the certificate (transition to the Certificate Issued state).

- Certificate Issued

  Once the certificate has been issued, the CA MUST allow the CA Administrator to revoke a certificate for different reasons, such as key compromise or cessation of operation, as described in [MS-CSRA] section 3.1.4.1.8 (transition to the Certificate Revoked state).

- Certificate Revoked

  When the certificate has been revoked, the CA MUST allow the administrator to return the certificate into issued state in certain cases, as specified in [MS-CSRA] section 3.1.4.1.8.

- Request Error

  The CA MUST NOT allow resubmission or modification of the request. That request will remain in the database for auditing or diagnostic reasons until it gets deleted by the CA administrator.

- Record Deleted

  As specified in [MS-CSRA] section 3.1.4.2.18, the CA SHOULD allow deleting individual records from the CA database. <5>

5.4 System Internal Architecture

There are several internal components that the CA implements. The interaction between those components is defined in [MS-WCCE] and [MS-CSRA]. The following figure illustrates the system implemented as a standalone CA, with the clients and administrator interacting with the system. It also shows the optional implementation of an exit algorithm.

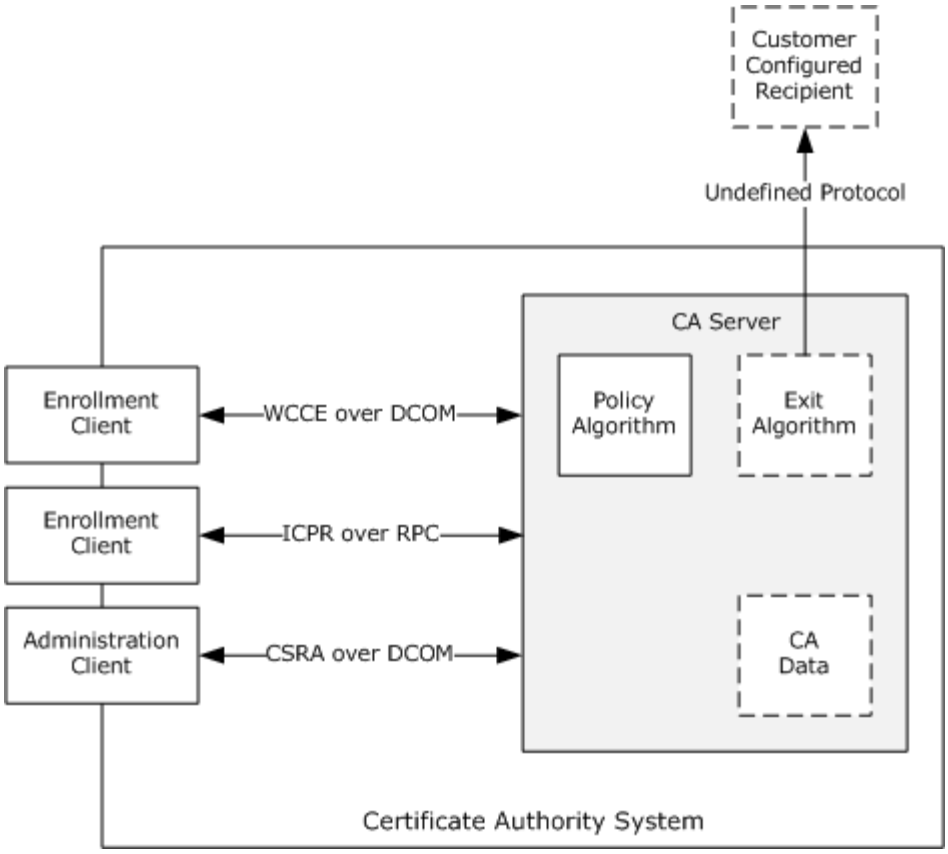

Figure 7: The CA System as a standalone CA

The following figure shows the system implemented as an enterprise CA. Note the addition of Active Directory, where the system stores and retrieves certificate templates.

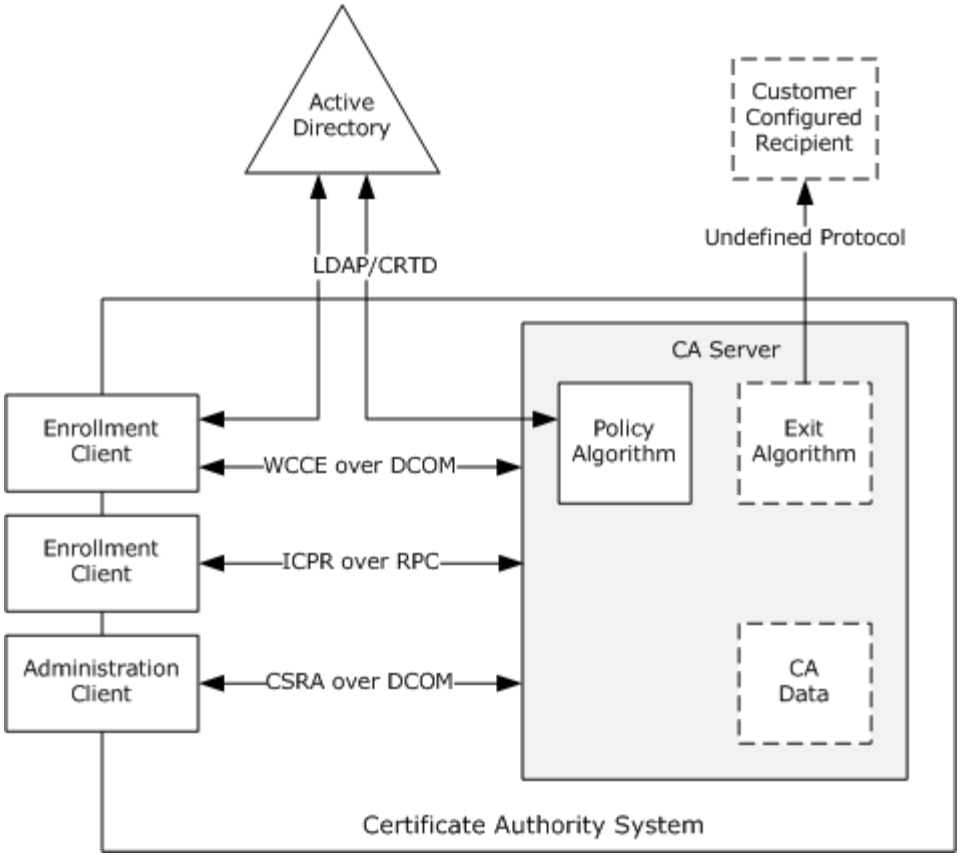

Figure 8: The CA System as an enterprise CA

5.4.1 Policy Algorithm

The CA policy algorithm is a required component of the system. Requests for new and renewed certificates are subject to the policy algorithm. It determines whether a certificate request is to be fulfilled, denied, or set to pending administrator approval. For example, a system implementing the enterprise CA functionality that is specified in [MS-WCCE] section 3.2.2 MUST verify that the requester has Enroll permission on the requested certificate template. The policy algorithm and rules for its implementation are defined in [MS-WCCE] sections 3.2.1.4.2.1.4.4 and 3.2.2.6.2.1.4.

5.4.2 Exit Algorithm

The CA exit algorithm is an optional internal component responsible for request post-processing, which MAY include communicating via some other protocol. For example, the CA could send email notifications to the end entity and system administrator when a new certificate is generated. The exit algorithm and rules for its implementation are defined in [MS-WCCE] section 3.2.1.4.2.1.4.8.

5.4.3 Data Storage

The method used for data storage is independent of the protocols and interfaces described in this document. The implementer can use a general purpose database, files stored in the operating system's native file system, or whatever is preferred. The data that needs to be stored is described in section 5.1, Abstract Data Model.

5.5 Failure Scenarios

Since the CA System operates through the use of multiple protocols, faults could occur in a variety of places. When these failures are encountered by the CA System, the protocol documents discuss error handling in the context of the use of the protocol individually. Many common failures can occur when the CA is running in enterprise mode and interacting with Active Directory. When failures occur in the context of Active Directory and domain connectivity, some errors can be mitigated through retries or redirections. The handling of domain connectivity failures is discussed in [MS-WCCE], section 3.2.2.1. Other errors or failures might be undetectable by the system, and in those scenarios, there is no real mitigation.

6 System Details

This section contains the details that complete the descriptions in earlier sections of the document. These details are needed to understand and implement this system.

6.1 Architectural Details

This section provides the following examples of the CA System certificate enrollment and administration processes:

- Enrollment from a standalone CA (basic enrollment)
- Enrollment from an enterprise CA (template based enrollment)
- Enrollment with CA Administrator approval
- Enroll on Behalf Of request and renewal
- Private key archival and recovery
- Certificate revocation
- Certificate request denied by policy algorithm
- Certificate request denied due to out-of-sync certificate templates

The processes shown in the following examples are each required for their given scenario, but not necessarily in the order shown. These are representations of typical process flows in common system usage scenarios.

6.1.1 Enrollment from a Standalone CA (Basic Enrollment)

The goal of this example is as detailed in the Enroll for a Certificate – End Entity use case, section 3.3.4.1. The simplest case of certificate enrollment is basic enrollment. In this example, the caller creates a PKCS#10 request by populating its fields as the caller chooses. The caller then uses an implementation that has a WCCE client component to submit the request to the WCCE server (CA). The CA in this example is a standalone CA configured to issue certificates for all requests so it responds to the caller's request by issuing a certificate.

Basic enrollment consists of a single message exchange between the client and the server where a client sends a certificate request to a server, which then issues the requested certificate. In this example the client implements the basic enrollment mode (in [MS-WCCE] section 3.1.1) and the server implements the standalone CA mode ([MS-WCCE] section 3.2.1).

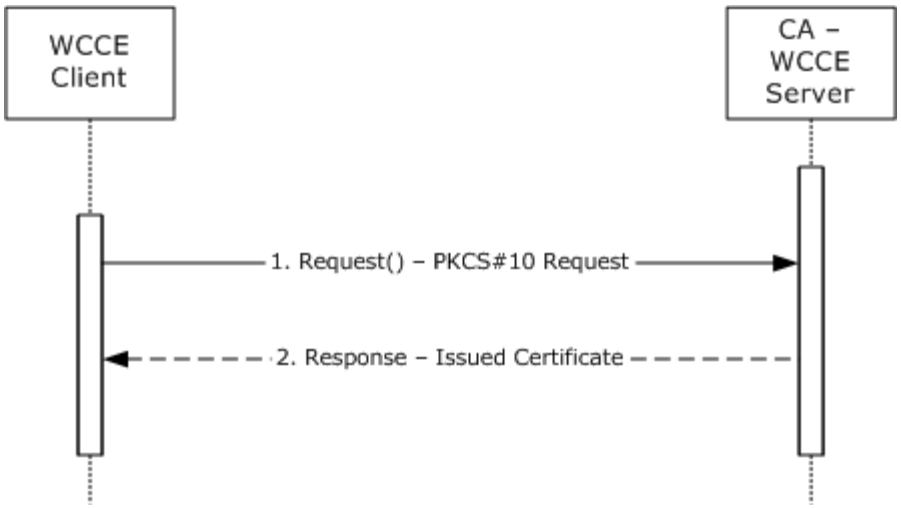

Figure 9: Basic enrollment

The message flow represented in the preceding figure is as follows:

1. The End Entity, using a WCCE client component, creates a PKCS#10 request and submits it to the CA as specified in [MS-WCCE] section 3.1.1.4.3.1.1.

2. The CA responds by issuing a certificate as specified in [MS-WCCE] section 3.2.1.4.2.1.4.1.1.

6.1.2 Enrollment from an Enterprise CA (Template Based Enrollment)

The goal of this example is as detailed in the Enroll for a Certificate – End Entity use case, section 3.3.4.1. This example builds on the example in section 6.1.1 by introducing an enterprise CA. An enterprise CA uses certificate templates for all certificate enrollments. Certificate templates, as defined in [MS-CRTD], contain data for requesting and issuing certificates. Policy algorithms use certificate templates to determine how to respond to certificate requests. In this example, the caller creates a certificate request PKCS#10, as specified in [MS-WCCE] section 3.1.1.4.3.1.1, based on the certificate template. The enterprise CA then uses the template information to decide whether to issue the certificate, and if it does, how to construct the certificate.

This example of certificate enrollment is based on the following assumptions:

- The End Entity operates in the client mode specified in [MS-WCCE] section 3.1.2 and the server implements the enterprise CA mode as specified in [MS-WCCE] section 3.2.2.

- It is also assumed that certificate templates are stored in Active Directory as specified by [MS-CRTD].

The process and specific message flows for this example are as follows:

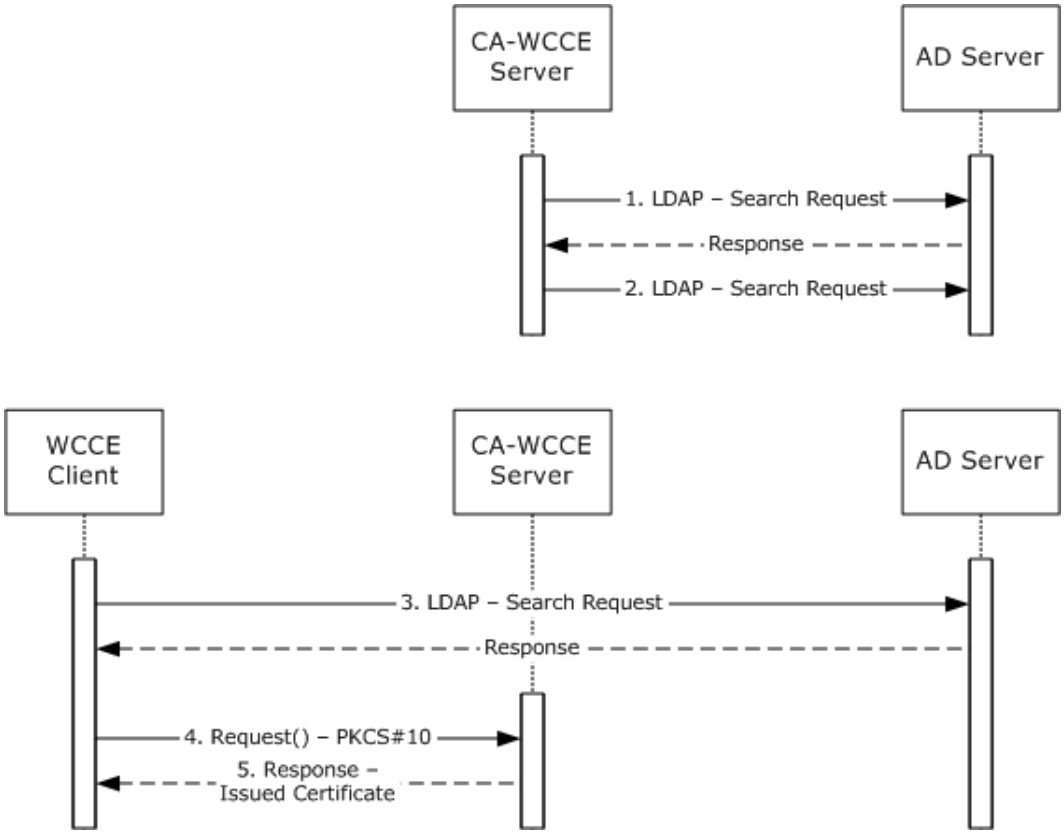

Figure 10: Template based enrollment

1. Upon startup, the CA retrieves certificate template data from the Active Directory server in the format specified in [MS-WCCE] section 3.2.2.1.1.

2. The CA registers itself to receive change notifications, as specified in [MS-ADTS] section 3.1.1.3.4.1.9, when an attribute of a certificate template is being modified in order to stay up to date with any changes and avoid having to retrieve the templates for each request.

3. The client retrieves the certificate templates from the Active Directory server via an LDAP search.

4. The client creates a PKCS#10 request based on one of the certificate templates and submits it to the CA by calling the Request method specified in [MS-WCCE] section 3.1.2.4.2.

5. The CA checks the policy defined in the certificate template and concludes that it is appropriate to issue the certificate (see [MS-WCCE] section 3.2.2.6.2.1.4). The CA constructs a new certificate as defined by the certificate template (see [MS-WCCE] section 3.2.2.6.2.1.4) and sends a new certificate to the client.

6.1.3 Enrollment with CA Administrator Approval

The goal of this example is as detailed in the Enroll for a Certificate – End Entity use case, section 3.3.4.1. This example builds on the example in section 6.1.2 by introducing a CA Administrator who modifies and approves the certificate request before the certificate is issued. One possible context for this scenario is where the certificate that is being requested requires a higher level of scrutiny before it can be issued, or needs input from someone other than the requestor.

This example of certificate enrollment is based on the following additional assumption:

- A certificate template has been defined in Active Directory that has the msPKI-Enrollment-Flag CT_FLAG_PEND_ALL_REQUESTS bit set (see [MS-WCCE] section 3.2.2.6.2.1.4.5.6.

The process and specific message flow for this example are as follows:

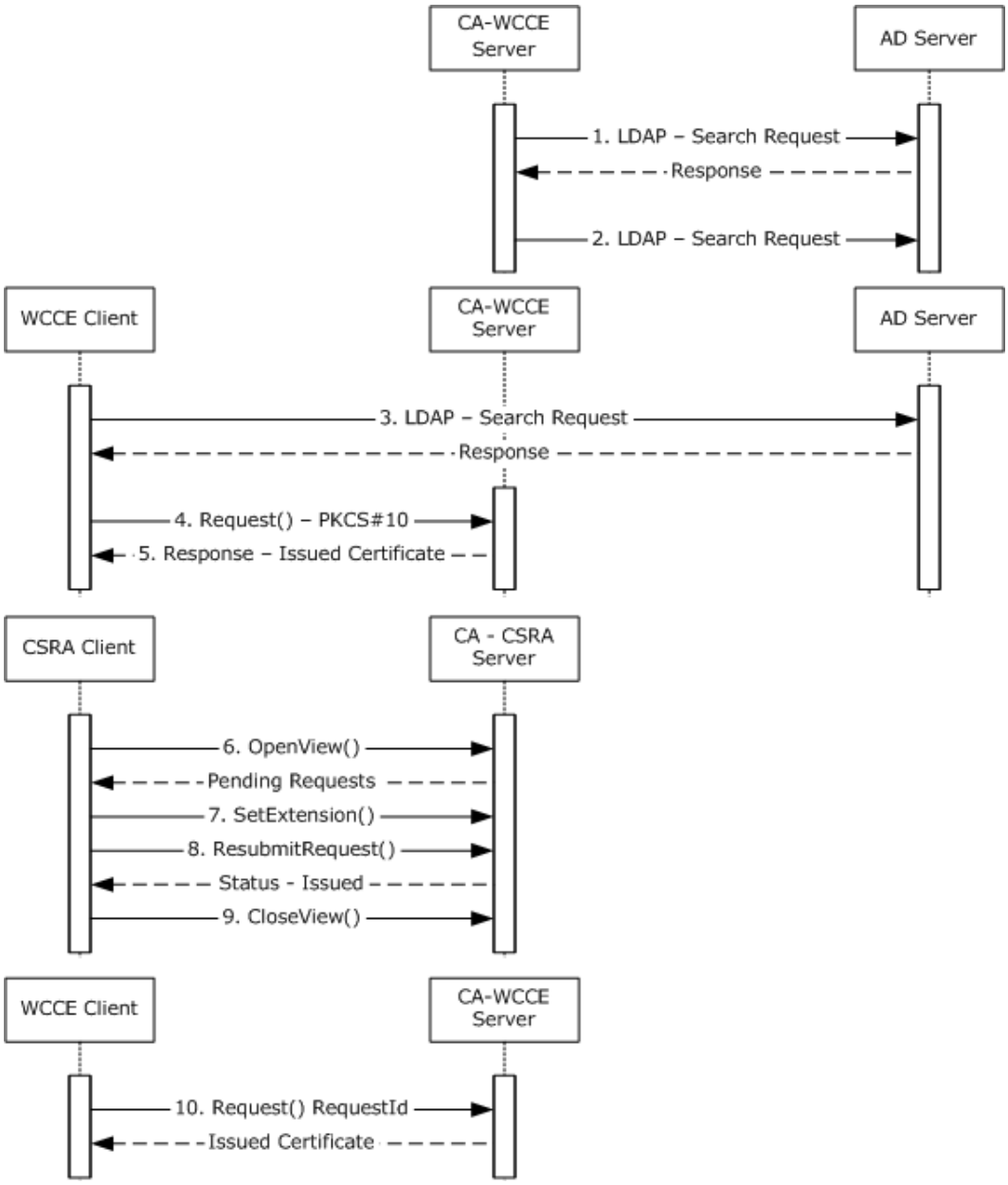

Figure 11: Enrollment with CA Administrator approval

1. Upon startup, the CA retrieves the certificate template data from the Active Directory server as specified in [MS-WCCE] section 3.2.2.1.1.

2. The CA registers itself to receive change notifications, as specified in [MS-ADTS] section 3.1.1.3.4.1.9, when an attribute of a certificate template is being modified in order to stay up to date with any changes and avoid having to retrieve the templates for each request.

3. The client retrieves the certificate templates from the Active Directory server via an LDAP search.

4. The client creates a PKCS#10 request based on one of the certificate templates and submits it to the CA by calling the Request method specified in [MS-WCCE] section 3.1.2.4.2.

5. The CA checks the certificate template and because the msPKI-Enrollment-Flag has the CT_FLAG_PEND_ALL_REQUESTS bit set (see [MS-WCCE] section 3.2.2.6.2.1.4.5.6) it records this request in its database and tells the client that the request's status is set to pending (see [MS-WCCE] section 3.2.1.4.2.1).

6. The CA Administrator, using an implementation that has a CSRA client component, queries the CA database to obtain information about pending requests by calling the OpenView method as specified in [MS-CSRA] section 3.1.4.1.12.

7. The CA Administrator adds a certificate extension to the request by calling the SetExtension method as specified in [MS-CSRA] section 3.1.4.1.1. Note that this step modifies the certificate request, but does not imply manager approval—which is done explicitly in the next step.

8. The CA Administrator approves the request by calling the ResubmitRequest method as specified in [MS-CSRA] section 3.1.4.1.3 and the certificate is created.

9. The CA Administrator closes the CA database view by calling the CloseView method as specified in [MS-CSRA] section 3.1.4.1.14.

10. After the certificate has been approved by the CA Administrator, the caller of the certificate retrieves the issued certificate from the CA by calling the Request method as specified in [MS-WCCE] section 3.1.1.4.3.5.

6.1.4 Enroll on Behalf of Request and Renewal

The goal of this example is as detailed in the Enroll for a Certificate – End Entity use case, section 3.3.4.1. This example builds on the example in section 6.1.2 by introducing a cosigner for the certificate request. In this example, the enrollment agent creates and signs the initial certificate request. The enrollment agent then submits the signed request to the CA. The CA returns the issued certificate to the enrollment agent, who then provides the issued certificate to the end entity via an out of band process.

Later, when a certificate needs to be renewed, the end entity creates a renewal request and signs it with the key associated with their current certificate. The certificate template is configured to allow renewals when the request is signed by the end entity's existing valid certificate that is based on the same template. The end entity submits the renewal request to the CA. If the certificate used for the signature is still valid, the CA automatically renews the certificate and returns the issued certificate to the client.

Smart card certificates are typically provisioned in this way. The smart card user might visit an enrollment agent in person so that their identity can be verified. The enrollment agent can then submit the certificate request on their behalf. The end entity, however, is allowed to renew their certificate without again requiring the involvement of the enrollment agent. By signing the renewal request with their existing valid certificate, they are providing evidence that their identity has already been verified.

This example has the following additional assumptions:

- A certificate template has been defined in Active Directory that has the msPKI-RA-Application-Policies attribute set with some enhanced key usage (EKU), for example, 1.3.6.1.4.1.311.20.2.1, Certificate Request Agent. The certificate template also has 0x00000040 (CT_FLAG_PREVIOUS_APPROVAL_VALIDATE_REENROLLMENT) bit set on the msPKI-Enrollment-Flag field.
- The enrollment agent has a certificate containing an EKU with the same object identifier (OID) as defined in the previous template msPKI-RA-Application-Policies attribute.

The process and specific message flow for this example are as follows:

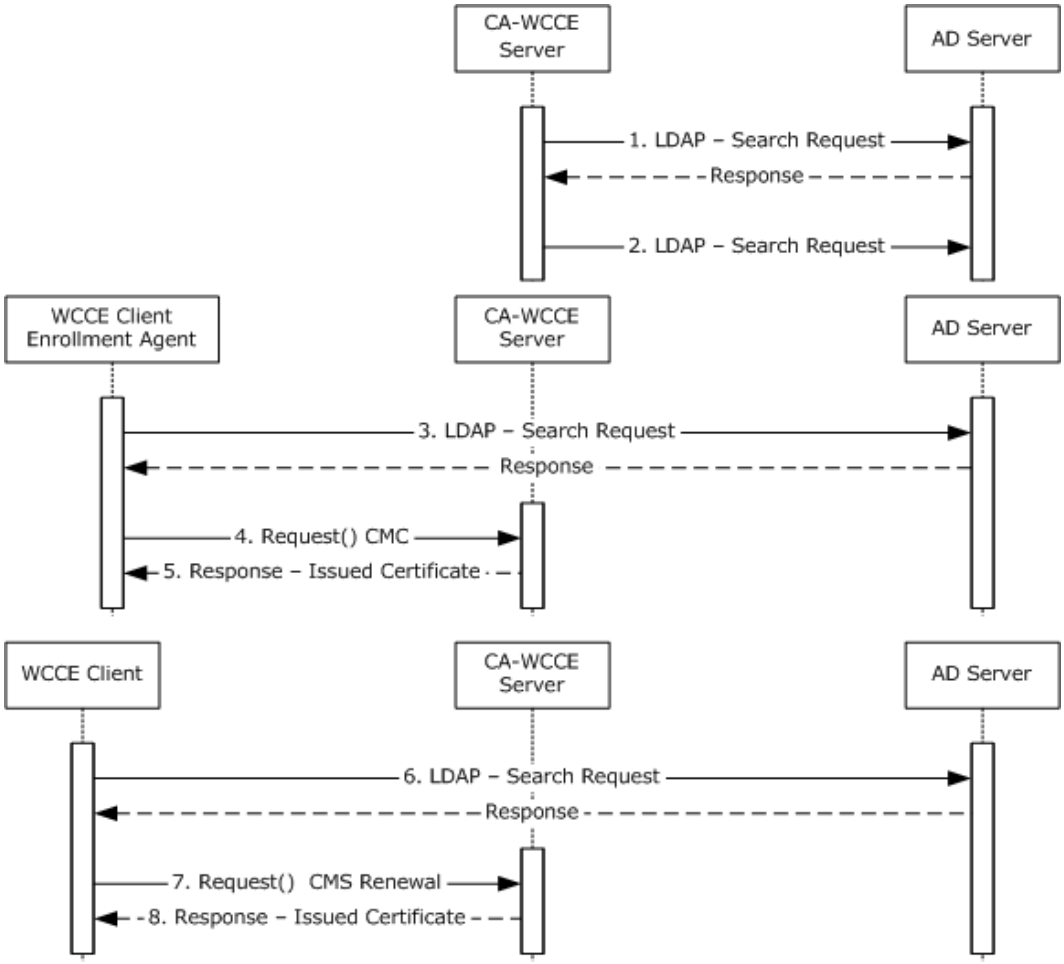

Figure 12: Enroll on Behalf of request and renewal

1. Upon startup, the CA retrieves certificate template data from the Active Directory server in the format specified in [MS-WCCE] section 3.2.2.1.1.
2. The CA registers itself to receive change notifications, as specified in [MS-ADTS] section 3.1.1.3.4.1.9, when an attribute of a certificate template is being modified in order to stay up to date with any changes and avoid having to retrieve the templates for each request.
3. The enrollment agent, using a WCCE client component, retrieves the certificate templates from the Active Directory server via an LDAP search.

4. The enrollment agent generates a Cryptographic Message Syntax (CMS) request on behalf of another user, as specified in [MS-WCCE] section 3.1.1.4.3.3, and submits it to the CA by calling the Request method as specified in [MS-WCCE] section 3.1.2.4.2.

5. The CA determines that the certificate template that corresponds to the request requires the enrollment agent's signature. It validates the signature and verifies that the certificate associated with the signature has the required EKUs as specified in [MS-WCCE] section 3.2.2.6.2.1.2.1. When validation is complete it issues the certificate and sends it to the enrollment agent.

6. The enrollment agent then transfers the new certificate to the end entity via an out of band process. The process for this communication is not defined within this document.

7. When it is time to renew the certificate, the end entity uses a WCCE client component to retrieve the certificate templates from the Active Directory server via an LDAP search.

8. The client creates a CMS renewal request and sends it to the CA as specified in [MS-WCCE] section 3.1.1.4.3.2.

9. When the CA receives the request, it checks the certificate template for the msPKI-Enrollment-Flag and confirms that the 0x00000040 (CT_FLAG_PREVIOUS_APPROVAL_VALIDATE_REENROLLMENT) bit is set, therefore allowing the use of the previous certificate to sign the request as specified in [MS-WCCE] section 3.2.2.6.2.1.2.3. The CA issues the renewed certificate and sends it to the End Entity.

6.1.5 Private Key Archival and Recovery

This example is the combination of two separate use cases. The first is the Enroll for a Certificate – End Entity use case, section 3.3.4.1, and the second is the Recover Archived Certificate and Key – CA Administrator use case, section 3.3.4.4. This example builds on the example in section 6.1.2 by introducing private key archival and recovery. A CA Administrator configures the CA to be able to archive private keys. An end entity enrolls for an encryption certificate, based upon a template that has been configured for archival. Later, a CA Administrator recovers the archived certificate and a private key from the CA database.

This example of key archival and recovery is based on the following additional assumptions:

- The certificate template has been configured in Active Directory with the msPKI-Private-Key-Flag attribute with the 0x00000001 (CT_FLAG_REQUIRE_PRIVATE_KEY_ARCHIVAL) bit as specified in [MS-CRTD].
- A CA Administrator is acting as a KRA and has obtained a KRA encryption certificate.

The process and specific message flow in this example are as follows:

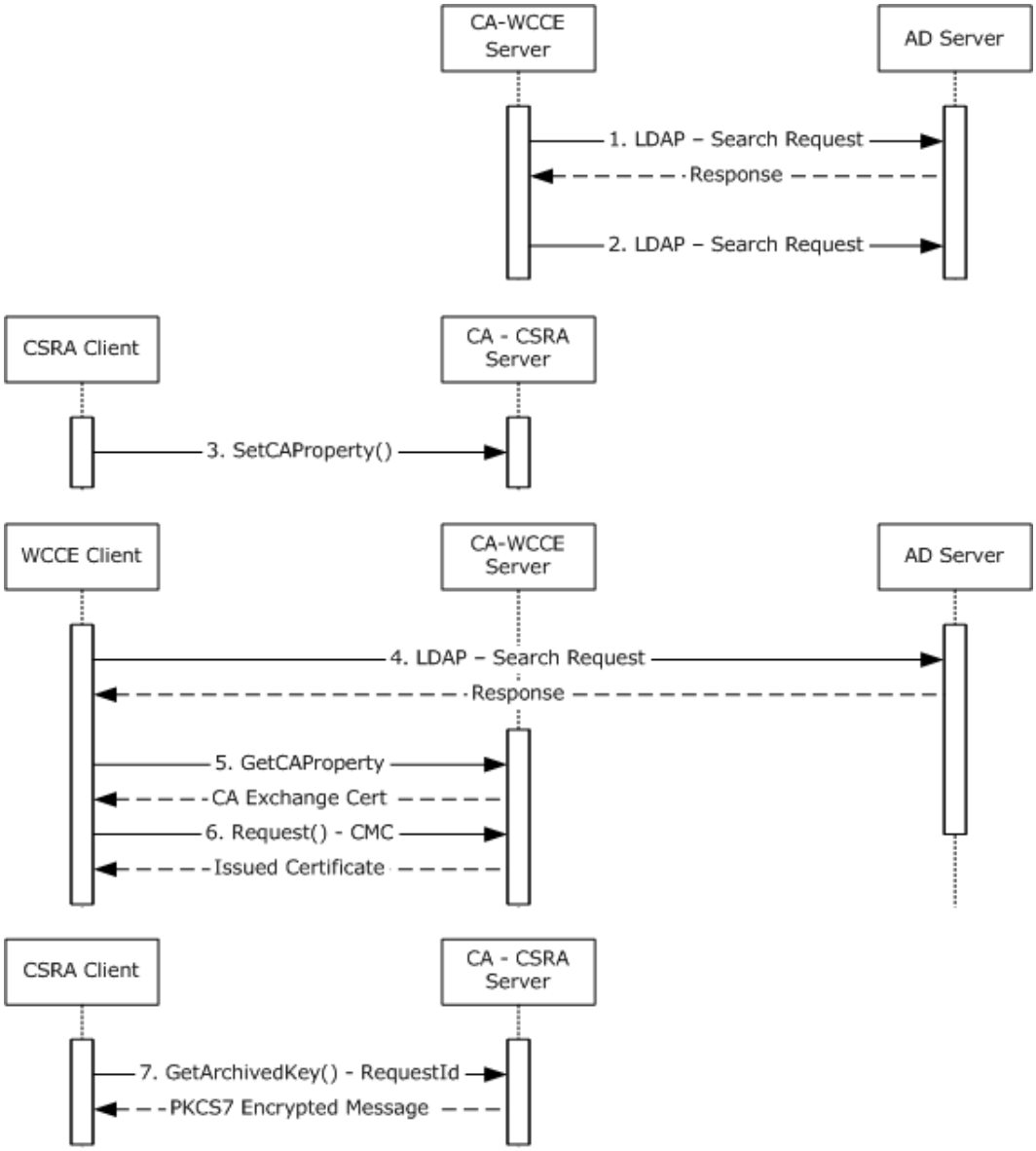

Figure 13: Private key archival and recovery

1. Upon startup, the CA retrieves certificate template data from the Active Directory server as specified in [MS-WCCE] section 3.2.2.1.1.

2. The CA registers itself to receive change notifications ([MSDN-LDAPSync]) when an attribute of a certificate template is being modified in order to stay up to date with any changes and avoid having to retrieve the templates for each request.

3. The CA Administrator configures the CA to use the KRA certificates when archiving private keys by calling the SetCAProperty method and setting the 0x0000001a (CR_PROP_KRACERT) and 0x00000018 (CR_PROP_KRACERTUSEDCOUNT) properties as specified in [MS-CSRA] section 3.1.4.2.3.

4. The end entity, using a WCCE client component, retrieves the certificate templates from the Active Directory server via an LDAP search.

5. Because the template specified by the caller has the 0x00000001 (CT_FLAG_REQUIRE_PRIVATE_KEY_ARCHIVAL) bit set on the msPKI-Private-Key-Flag attribute, the client calls GetCAProperty to retrieve the CA exchange certificate. The CA exchange certificate is required to construct a request with private key archival as specified in [MS-WCCE] section 3.1.1.4.3.4.

6. The client constructs a CMS request and sends it to the CA as specified in [MS-WCCE] section 3.1.1.4.3.4. When the CA receives the request it issues a new certificate and archives the private key by using the KRA certificate that the administrator configured in step 3 as specified in [MS-WCCE] section 1.3.2.1.

7. Later, when the CA Administrator needs to recover the archived private key, using a CSRA client component he calls the GetArchivedKey method to retrieve it as specified in [MS-CSRA] section 3.1.4.2.9.

6.1.6 Certificate Revocation

The goal of this example is as detailed in the Revoke a Certificate – CA Administrator use case. This example builds on the example in section 6.1.2 by adding the process of revoking a previously issued certificate. A CA Administrator is able to revoke certificates for different reasons, such as private key compromise or cessation of operation.

This example of key archival and recovery is based on the following additional assumptions:

- The Config.CA.CDP.Publish.To datum of the CA's ADM (see [MS-WCCE] section 3.2.1.1.4) is configured with valid paths.

The process and specific message flow in this example are as follows:

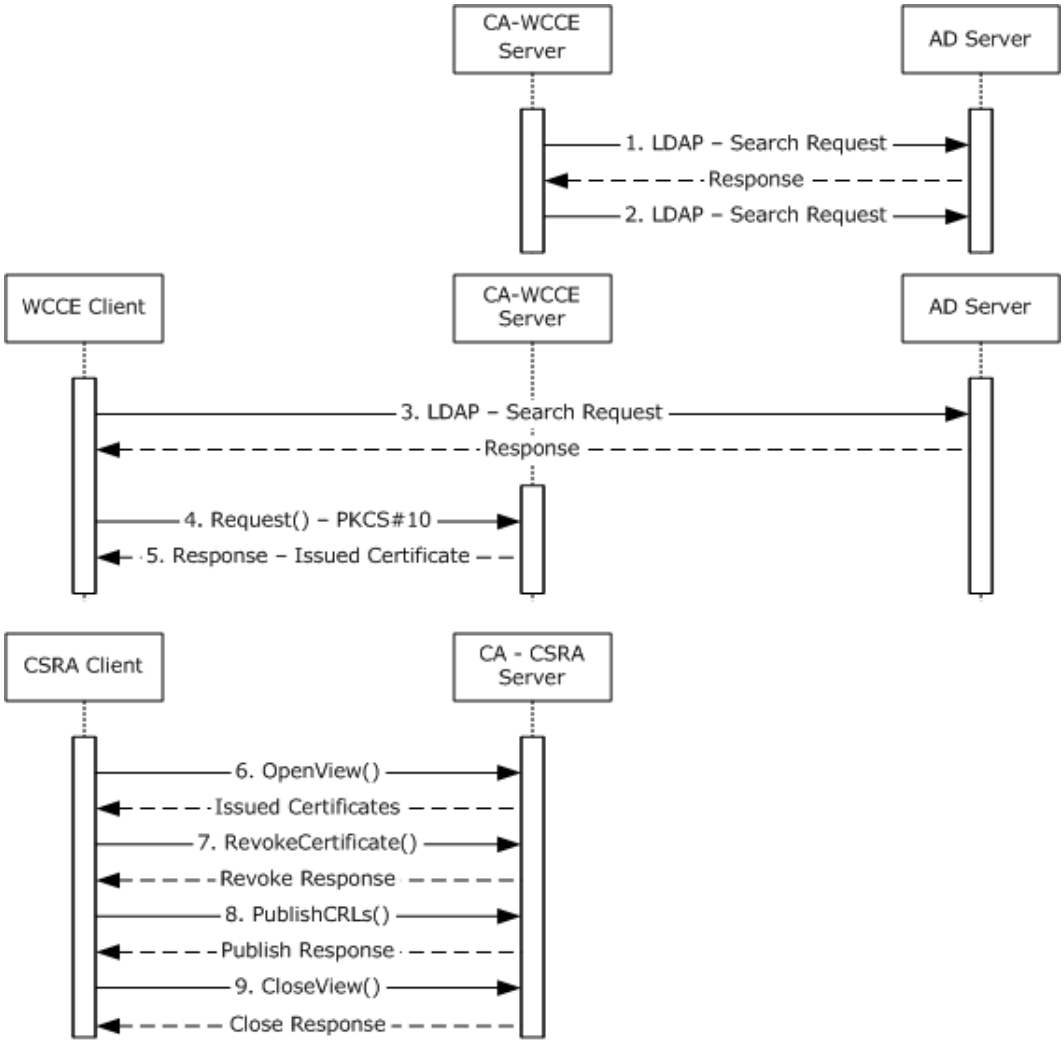

Figure 14: Certificate revocation

1. Upon startup, the CA retrieves certificate template data from the Active Directory server as specified in [MS-WCCE] section 3.2.2.1.1.

2. The CA registers itself to receive change notifications, as specified in [MS-ADTS] section 3.1.1.3.4.1.9, when an attribute of a certificate template is being modified in order to stay up to date with any changes and avoid having to retrieve the templates for each request.

3. The end entity, using a WCCE client component, retrieves the certificate templates from the Active Directory server via an LDAP search.

4. The client creates a PKCS#10 request based on one of the certificate templates and submits it to the CA by calling the Request method specified in [MS-WCCE] section 3.1.2.4.2.

5. The CA checks the policy defined by the certificate template and concludes that it is appropriate to issue the certificate (see [MS-WCCE] section 3.2.2.6.2.1.4).

6. At some point later, it is determined that the issued certificate is to be revoked. The CA Administrator, using a CSRA client component, queries the CA database to obtain the serial number of the certificate to be revoked by calling the OpenView method as specified in [MS-CSRA] section 3.1.4.1.12.

7. The CA Administrator revokes the certificate by calling RevokeCertificate method as specified in [MS-CSRA] section 3.1.4.1.8.

8. The CA Administrator instructs the CA to publish the CRL by calling the PublishCRLs method as specified in [MS-CSRA] section 3.1.4.2.1.

9. The CA Administrator closes the CA database view by calling the CloseView method as specified in [MS-CSRA] section 3.1.4.1.14.

6.1.7 Certificate Denied by Policy Algorithm

This example represents a failure scenario for the Enroll for a Certificate – End Entity use case. This example builds on the example in section 6.1.2, however, in this scenario, the caller requests a certificate based on a template for which it does not have permission to enroll. In this case the CA's policy algorithm denies the request after examining the permissions applied to the template.

The process and specific message flow in this example are as follows:

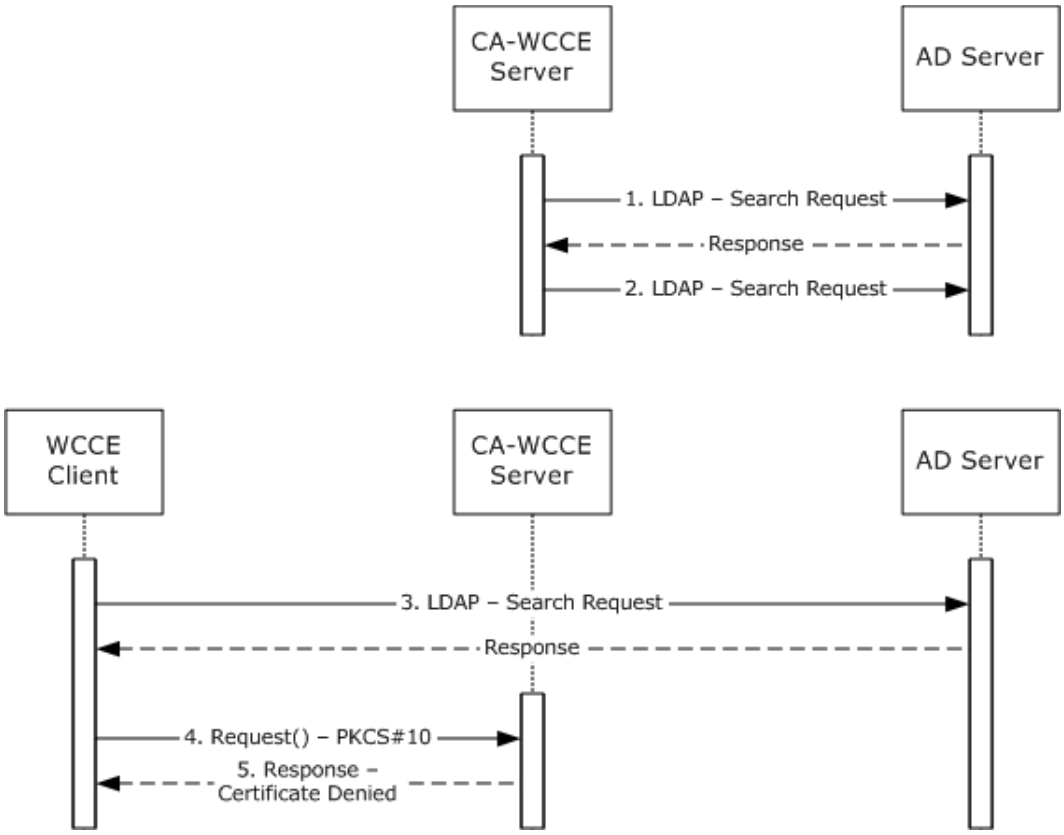

Figure 15: Certificate request denied by policy

1. Upon startup, the CA retrieves certificate template data from the Active Directory server as specified in [MS-WCCE] section 3.2.2.1.1.

2. The CA registers itself to receive change notifications, as specified in [MS-ADTS] section 3.1.1.3.4.1.9, when an attribute of a certificate template is being modified in order to stay up to date with any changes and avoid having to retrieve the templates for each request.

3. The end entity, using a WCCE client component, retrieves the certificate templates from the Active Directory server via an LDAP search.

4. The client creates a PKCS#10 request based on one of the certificate templates and submits it to the CA by calling the Request method specified in [MS-WCCE] section 3.1.2.4.2.

5. When the CA receives the request, the policy algorithm is checked to see if it is to be issued. The CA examines ntSecurityDescriptor of the certificate template that corresponds to the request to determine if the caller has the permissions required to enroll for that template as specified in [MS-WCCE] section 3.2.2.6.2.1.4.3, and [MS-CRTD] section 2.5. In this example, the caller does not have permission, so the error 0x80094012L (CERTSRV_E_TEMPLATE_DENIED) is returned.

6.1.8 Certificate Denied Due to Out-of-Sync Certificate Templates

This example represents another failure scenario for the Enroll for a Certificate – End Entity use case. This example builds on the example in section 6.1.2 and illustrates a situation where two Active Directory servers are out of sync, resulting in a version mismatch between the certificate templates used by client and server. Due to this mismatch, the server rejects the request. At a later time, after the directory has synchronized, the client submits another request that results in the certificate being issued.

This example of key archival and recovery is based on the following additional assumptions:

- There is more than one Active Directory server on this network that replicates periodically.
- Active Directory replication occurs as discussed in [MS-DRDM].
- A CA Administrator has the appropriate security permissions to make modifications to certificate templates stored within Active Directory. Modifications made to Active Directory are performed as specified within [MS-ADTS].

The process and specific message flow in this example are as follows:

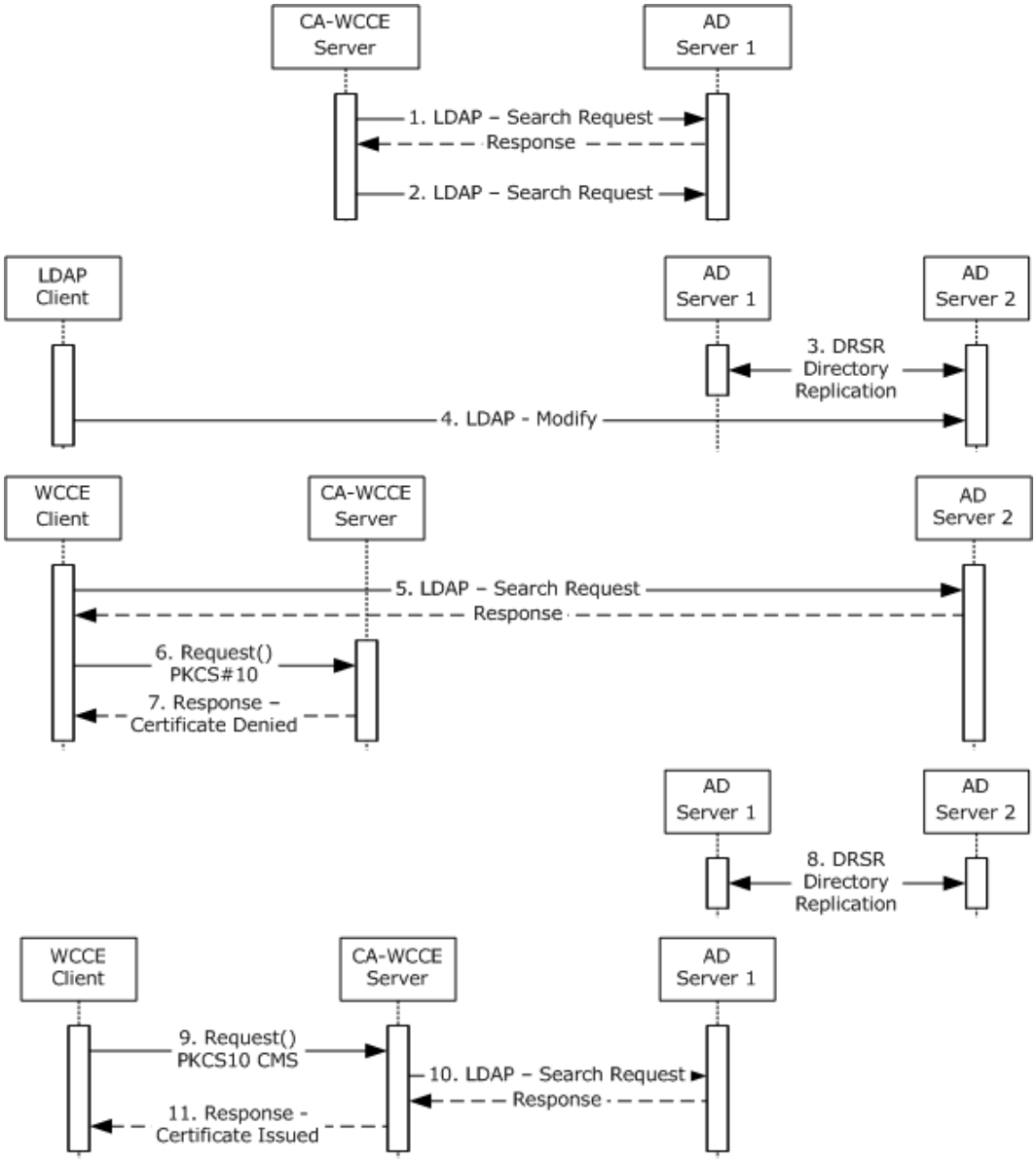

Figure 16: Certificate denied due to out-of-synch certificate templates

1. Upon startup, the CA retrieves certificate template data from the Active Directory Server 1 as specified in [MS-WCCE] section 3.2.2.1.1.

2. The CA registers itself with Active Directory (AD) Server 1 to receive change notifications, as specified in [MS-ADTS] section 3.1.1.3.4.1.9, when an attribute of a certificate template is being modified in order to stay up to date with any changes and avoid having to retrieve the templates for each request.

3. At some later time, the two Active Directory servers replicate their information between each other as specified in [MS-DRDM].

4. After the replication, a CA Administrator modifies some of the certificate templates with new policies on Active Directory Server 2. Modifications to Active Directory are performed as detailed in [MS-ADTS].

5. The end entity, using a WCCE client component, retrieves the certificate templates from the Active Directory Server 2 via an LDAP search.

6. The client creates a PKCS#10 request based on one of the certificate templates and submits it to the CA by calling the Request method specified in [MS-WCCE] section 3.1.2.4.2.

7. As described in [MS-WCCE] section 3.2.2.6.2.1.4.2, the CA's policy algorithm verifies the certificate template version. The changes made on Active Directory Server 2 have not yet replicated to Active Directory Server 1. Since the CA has not been notified of the change to the template and the CA's certificate template instance is of an older version, the CA rejects a request, and replies with error (CERTSRV_E_BAD_TEMPLATE_VERSION).

8. At some later time, the two Active Directory servers replicate their information between each other as specified in [MS-DRDM].

9. At some time later, the client attempts again the request for the same certificate in the same way as step 6.

10. This time the CA knows about the change in Active Directory because it has registered for asynchronous notifications in step 2. The CA retrieves the updated certificate template data from the Active Directory Server 1 as specified in [MS-WCCE] section 3.2.2.1.1.

11. The CA checks policy defined by the certificate and issues the certificate (see of [MS-WCCE]). The CA constructs the new certificate as defined by the certificate template (see [MS-WCCE] section 3.2.2.6.2.1.4) and returns the new certificate to the client.

6.2 Communication Details

The system components communicate though the shared ADM elements that are listed in the Abstract Data Model Section 6.1.

All messages and their processing rules are documented in the protocol specifications for the protocols that this system implements as specified in section 2.2

6.3 Transport Requirements

The system specific transport requirements are defined in [MS-WCCE] section 2.1, [MS-ICPR] section 2.1, and [MS-CSRA] section 2.1.

6.4 Timers

There are no timer events for this system.

6.5 Non-Timer Events

The system-specific events are defined in [MS-WCCE] section 3.2.1.4, [MS-ICPR] section 3.2.4, and [MS-CSRA] section 3.1.4.

A system implementing the enterprise CA mode SHOULD register for the asynchronous change notifications to the certificate template objects in Active Directory. See [MS-WCCE] section 3.2.2.8.

6.6 Initialization and Reinitialization Procedures

The initialization processes are defined in [MS-WCCE] sections 3.2.1.3 and 3.2.2.3, [MS-ICPR] section 3.2.3, and [MS-CSRA] section 3.1.3.

6.7 Status and Error Returns

The error handling requirements are defined in [MS-WCCE] section 2.2.4 and [MS-CSRA] section 2.2.5.

7 Security

This section documents system-wide security issues that are not otherwise described in the Technical Documents (TDs) for the Member Protocols. It does not duplicate what is already in the Member Protocol TDs unless there is some unique aspect that applies to the system as a whole.

Security is paramount for a CA because it stores sensitive data and issues certificates that are used for processes that can involve the organization's most important data. Therefore, it is critical that the system implementation be robust and resistant to attack. The security considerations include protection of the CA's signing and key exchange keys, protection of the requestor's data and private key that is being archived, and enforcement of certificate issuance policies that have been configured. These considerations call for the implementation of suitable protection for the storage of the CA's data, suitable protection of key recovery procedures, and the use of certificate templates for policy enforcement

The CA serves as the foundation for authentication, authorization, encryption, and digital signatures. In other words, it is the cornerstone for many of an organization's information security capabilities. A good implementation of this system will include robust protection of data that is stored locally and transmitted to remote clients. [MS-CSRA] section 5 and [MS-WCCE] section 5 discuss security issues specific to the individual protocols.

7.1 Internal Security

There are several internal areas of the CA that have security considerations worth mentioning. This section discusses these in greater detail.

7.1.1 CA Signing Key

The CA uses its signing key to sign all certificates that it issues and all the **CRLs** that it publishes. This key is bound to the CA signing certificate. Therefore, there are several important properties to consider:

- **Strength of the key**

Acceptable algorithms and key lengths are to be specified by enterprise security policy.

- **Lifetime of the key**

The CA signing keys are long-lived keys that exceed the lifetime of the certificates that they sign because, when that key expires, all certificates signed with that key are no longer considered valid by others.

- **Key storage**

If the CA signing key is compromised, certificates that were signed with that key can no longer be trusted. This is because an attacker could issue certificates that appear to originate from that CA.

- **CA signing certificate revocation**

Organizations need to have a documented process to handle the compromise of CA signing keys. For example, if the CA is subordinate to another CA, then it would make sense to revoke the compromised certificate on the parent CA and publish a new CRL. An even more dire situation occurs when the signing key of a **root CA** is compromised. In this situation, the only way to stop it from being **trusted** is to reconfigure all of the client computers to no longer trust it.

7.1.2 CA Data

Attackers could interfere with CA operations or tamper with certificate **revocation** information if they were able to access the CA data defined in section 5.1, Abstract Data Model. Therefore, it is a good idea to implement strong controls to protect this data and ensure that only authorized administrators are able to manage it.

Much of the data stored in the database is provided by the caller requesting a certificate. This caller could actually be an attacker. Therefore, it is recommended that each incoming request be validated before it is processed by the system. That is, a CA might inspect each incoming request to ensure that each field within the request is formatted correctly and that it does not exceed a reasonable size [HOWARD].

7.1.3 Certificate Templates

[MS-WCCE] section 5.1.11 describes data consistency considerations for the **certificate templates**. Additionally, it is reasonable to restrict write access to a certificate template to the administrators. Certificate templates define a policy by which certificates are issued. Therefore, an attacker who can modify certificate templates could potentially obtain certificates that would otherwise have been unobtainable.

7.1.4 Certificates for Special Roles

Although not required by the protocol, it is recommended to restrict the use of certificates that are issued for **KRAs** and enrollment agents by requiring explicit **CA administrator** approval. These certificates have special purposes in some of the scenarios for this system, as illustrated in examples in sections 6.1.4 and 6.1.5.

7.1.5 Caller Authentication

As specified in [MS-CSRA] section 1.4 and in [MS-WCCE] section 2.1, the CA depends on a component implementing the server role of DCOM authentication to identify the caller of the **DCOM** interfaces that it implements.

7.2 External Security

There are several external areas of the CA that have security considerations worth mentioning. These external areas are discussed in this section.

7.2.1 Private Key Archival

There are several considerations for key archival. These considerations include transporting the private key from the client to the CA, storing the private key on the server, and recovering lost keys. Note that, while message formats and specific processing rules are described in [MS-CSRA] and [MS-WCCE], only security considerations are discussed here. [MS-WCCE] section 5.1.10 also addresses security considerations for the key archival.

7.2.2 CA Exchange Certificate

The public key in the CA exchange certificate can be used to encrypt end entities' private keys when requests for new certificates are sent to the CA (see [MS-WCCE] section 3.1.1.4.3.4). The concerns for key length that were presented for the CA signing key apply to this key as well. However, the lifetime of this private key might be shorter than the lifetime of the CA signing key. Also, this private key is not required to extend the validity of the certificates that the CA issues.

If this key is compromised, all of the certificates and private keys that were processed using the key can no longer be trusted since an attacker who possesses the private key could intercept and decrypt the end entity's private key.

Key storage considerations are the same as for the CA signing key. These certificates can be revoked and not used by the CA if they are compromised.

Storage and transmission of the Exchange public key is important because an attacker might generate its own key pair and if it could substitute its public key for a CA's Exchange public key, the client would be induced into encrypting a private key using that key for which the attacker has the private key.

7.2.3 Archived Key Storage

Neither the protocols nor the CA mandates any particular protection mechanism for the private keys archived by a CA. When choosing an algorithm and key sizes for the key protection, it is recommended that an implementer consider the lifetime of the key that is being protected and document its strength to set expectations for the clients of the system. For more information about the key archival and recovery process in the Windows platform, see [MSFT-ARCHIVE].

7.2.4 Key Recovery Agent Certificates

KRA certificates and the private keys associated with them can be used to protect and recover end entities' private keys. The CA does not need to possess the KRA's private key to archive keys, so the storage responsibility for KRA keys is solely on KRAs themselves. However, the CA Administrator who defines policies about what types of KRA certificates are issued and configured on the CA can ensure that they are appropriate for this purpose.

The KRA public key needs to be protected from tampering and especially replacement, because an attacker that could substitute its own public key for the KRA public key would potentially have access to all private keys encrypted under the KRA public key.

7.2.5 Transport Security

The CA uses the DCOM and **RPC** protocols for transport. Both DCOM and RPC provide authentication, data integrity, and **encryption** capabilities. Although those modes are not required by the CA protocols themselves, it is strongly recommended to both authenticate and encrypt at the DCOM and RPC layers.

7.2.6 Privacy

The CA stores data that was submitted by the client when the certificate was requested. Some of these data can be considered private by law in many jurisdictions, so it is important to provide access protection to the CA database for compliance.

8 Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs:

- Microsoft Windows® 2000 operating system
- Windows® XP operating system
- Windows Server® 2003 operating system
- Windows Server® 2003 R2 operating system
- Windows Vista® operating system
- Windows Server® 2008 operating system
- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

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

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms SHOULD or SHOULD NOT implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that the product does not follow the prescription.

<1> Section 2.2: The ICertPassage Remote Protocol was first introduced in Windows NT 4.0 as the first certificate enrollment protocol for the Windows operating system. The Windows Client Certificate Enrollment Protocol, as specified in [MS-WCCE], was introduced in Windows 2000 server as a replacement for the ICertPassage Remote Protocol. While WCCE is the preferred certificate enrollment protocol and the default used in all version since Windows 2000, ICPR is still present in all Windows versions.

<2> Section 4.3.2: Windows 2000, Windows Server 2003 and Windows XP do not recognize nor honor the group policy setting allowing an administrator to define the certificate template that EFS will attempt to enroll for. This functionality was introduced in Windows Vista.

<3> Section 4.3.2: Windows 2000, Windows Server 2003 and Windows XP do not recognize nor honor the group policy setting allowing an administrator to disable the ability for EFS to create a self signed certificate. This functionality was introduced in Windows Vista.

<4> Section 5.3.2: Only Windows Server 2008 R2 supports the use of the certificate template flag CT_FLAG_DONOTPERSISTINDB, as specified in [MS-WCCE] section 3.2.2.6.2.1.4.4.1. If this flag is set and if the certificate has been issued, the CA SHOULD NOT persist the information about the request in the Request table.

<5> Section 5.3.2: Windows 2000 does not allow the deletion of records from the CA database.

9 Change Tracking

This section identifies changes that were made to the [MS-CASO] protocol document between the January 2011 and February 2011 releases. Changes are classified as New, Major, Minor, Editorial, or No change.

The revision class **New** means that a new document is being released.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements or functionality.
- An extensive rewrite, addition, or deletion of major portions of content.
- The removal of a document from the documentation set.
- Changes made for template compliance.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **Editorial** means that the language and formatting in the technical content was changed. Editorial changes apply to grammatical, formatting, and style issues.

The revision class **No change** means that no new technical or language changes were introduced. The technical content of the document is identical to the last released version, but minor editorial and formatting changes, as well as updates to the header and footer information, and to the revision summary, may have been made.

Major and minor changes can be described further using the following change types:

- New content added.
- Content updated.
- Content removed.
- New product behavior note added.
- Product behavior note updated.
- Product behavior note removed.
- New protocol syntax added.
- Protocol syntax updated.
- Protocol syntax removed.
- New content added due to protocol revision.
- Content updated due to protocol revision.
- Content removed due to protocol revision.
- New protocol syntax added due to protocol revision.

- Protocol syntax updated due to protocol revision.
- Protocol syntax removed due to protocol revision.
- New content added for template compliance.
- Content updated for template compliance.
- Content removed for template compliance.
- Obsolete document removed.

Editorial changes are always classified with the change type **Editorially updated.**

Some important terms used in the change type descriptions are defined as follows:

- **Protocol syntax** refers to data elements (such as packets, structures, enumerations, and methods) as well as interfaces.
- **Protocol revision** refers to changes made to a protocol that affect the bits that are sent over the wire.

The changes made to this document are listed in the following table. For more information, please contact protocol@microsoft.com.

Section	Tracking number (if applicable) and description	Major change (Y or N)	Change type
5.4.1 Policy Algorithm	63741 Updated reference to [MS-WCCE] to point to the "CA Policy Algorithm" section.	Y	Content updated.

10 Index