[MS-CCRSOD]: Content Caching and Retrieval System Overview Document

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 Open Specification Promise or the Community Promise. 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.

DRAFT: FOR PREVIEW ONLY

This document provides an overview of the Content Caching and Retrieval System Overview Document 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

Provides an overview of the functionality and relationship of the protocols implemented in the Windows Peer Content Caching and Retrieval framework, which includes the protocols specified in [MS-FSCC], [MS-PCCRC], [MS-PCCRD], [MS-PCCRR], [MS-PCHC], [MS-PCCRTP], and [MS-SMB2]. This framework is based on a peer-to-peer discovery and distribution model designed to reduce wide-area-network (WAN) link bandwidth utilization and provide faster content downloads from a local area network (LAN) in a branch office. The Content Caching and Retrieval protocols support scenarios such as accessing content from a file server or a Web server where storing content locally from all locations in a distributed environment is not practical. If the Content Caching and Retrieval of data is unavailable or fails, normal file access would continue without caching using the SMB 2.1/2, HTTP, or HTTPS protocols.

Revision Summary

Date	Revision History	Revision Class	Comments
09/23/2011	1.0	New	Released new document.

DRAFT: FOR PREVIEW ONLY

Contents

DRAFT - FOR PREVIEW ONLY

Copyright © 2011 Microsoft Corporation.

Release: Sunday, October 16, 2011

DRAFT - FOR PREVIEW ONLY

DRAFT: FOR PREVIEW ONLY

1 Introduction

The Content Caching and Retrieval System supports **content** retrieval scenarios such as accessing content from a file or Web server. For file access scenarios, this document can be used in conjunction with the File Access Services System Overview [MS-FASOD]. [MS-FASOD] describes the protocols required for network File Access Services interoperation with Microsoft Windows® systems. This document describes the additional protocols, data structures, and mechanisms, such as security, that are required to enable a system of Content Caching and Retrieval to interoperate with Windows systems. The system is designed to support scenarios in which local storage of all possible content at all locations in a distributed network is not practicable, such as in corporate branch offices. If the Content Caching and Retrieval of data is unavailable or fails, normal file access would continue without caching using the SMB 2.1/2, HTTP, or HTTPS protocols.

Content within the system is divided up into **segments** and **blocks**, a block being a subdivision of a segment. It is segments and blocks that are stored and retrieved by the system, rather than files.

Content caching and retrieval requires at least three computers: one computer to act as a **content server** (normally located on a wide area network (WAN) link), one to act as a client that is requesting content, and a third (normally a computer on the same local area network (LAN) as the requesting client) to hold in cache some or all of the content that the client computer is requesting.

1.1 Conceptual Overview

1.1.1 Content Identifiers

For the purposes of the Content Caching and Retrieval System, content is considered to be divided into one or more segments. Segments are the unit of discovery.

Each segment is a binary string of a standard size (32 megabytes (MB)), except the last segment, which may be smaller if the content size is not a multiple of the standard segment size. Each segment is identified on the network by its Segment Identifier, also known as **HoHoDk**. Different content items can share the same segment if they contain an identical part that coincides with a complete segment.

Each segment is divided, in turn, into blocks. Each block is a binary string of a fixed size (64 kilobytes (KB)), except for the last block in the last segment, which may be shorter. Unlike segments, blocks in different segments are always treated as distinct objects, even if they are identical. Blocks within a segment are identified by their progressive index within the segment (for example, Block 0 is the first block in the segment, Block 1 is the second, and so on). Because of the fixed block size, a block's index can also be used to compute its actual byte offset in the segment. Given the standard block size of 64 KB, Block 0 is located at offset 0 in the segment, Block 1 is at offset 65536, Block 2 is at offset 131072, and so on. Blocks are the unit of download.

The following figure depicts how the content is divided into segments and blocks.

DRAFT - FOR PREVIEW ONLY

Copyright © 2011 Microsoft Corporation.

Release: Sunday, October 16, 2011

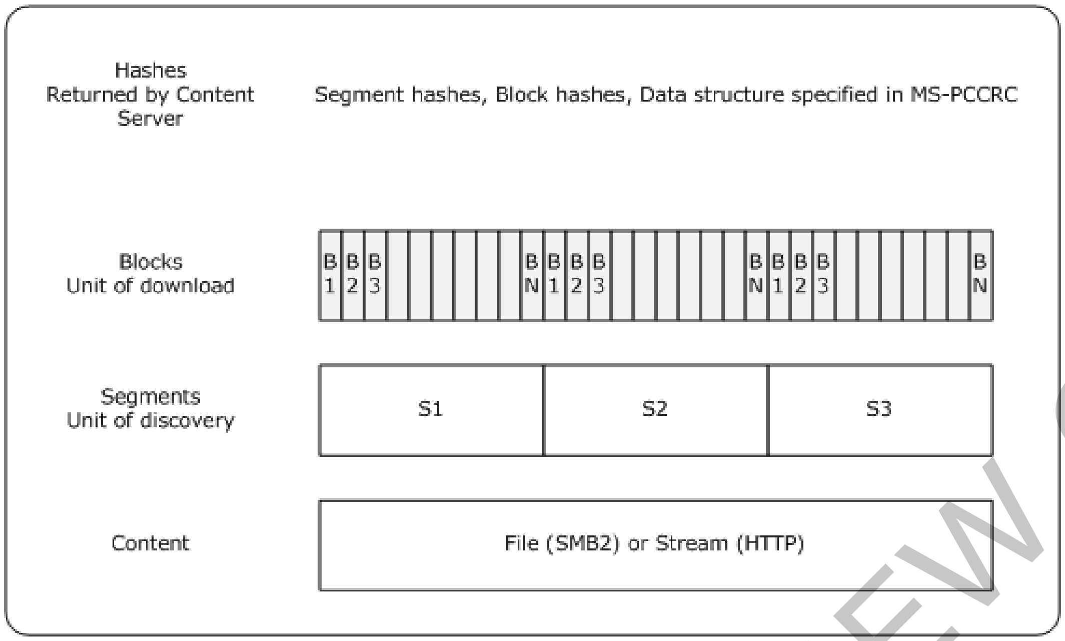

Hashes
Returned by Content Server
Segment hashes, Block hashes, Data structure specified in MS-PCCRC
Blocks
Unit of download
B 1
B 2
B 3
B N
B 1
B 2
B 3
B N
B 1
B 2
B 3
B N
Segments
Unit of discovery
S1
S2
S3
Content
File (SMB2) or Stream (HTTP)

Figure 1: Content identifiers

Note that, given the entire set of blocks for a segment, each identified by index, one can reconstruct the original segment simply by concatenating the blocks in order by index. Similarly, given the entire sequence of HoHoDk values for the successive segments in a content item, and a set of segments with matching associated HoHoDk values, one can reconstruct the original content simply by concatenating the segments in order based on value. Details about calculating the identifiers and keys can be found in [MS-PCCRC] section 2.2.

1.1.2 Client-Role Peer

This section describes how the Content Caching and Retrieval System, running on multiple **peers**, uses the Discovery Protocol [MS-PCCRD] and the Retrieval Protocol [MS-PCCRR] to allow the **client-role peer** to retrieve **content blocks** of a target segment from one or more **server-role peers**. Requests come from higher-layer applications on the client-role peer to retrieve the whole or parts of a content item, which may span multiple segments. For each target segment, the client-role peer uses the Discovery Protocol to find a server-role peer (or it directly contacts a **hosted cache** server) that has (the whole or parts) of the target segment. The client-role peer then initiates Retrieval Protocol exchanges to each server-role peer to query the block ranges held by each server-role peer, and downloads the blocks.

The Discovery Protocol and the Retrieval Protocol both operate on or within a single segment. The operations described in this section allow a client-role peer to find and retrieve blocks (parts or all) of a single target segment. This process is referred to as a **segment retrieval session**. If the content spans multiple segments, then multiple segment retrieval sessions are required to retrieve all of the content's segments and reassemble them into the complete content item.

1.2 Glossary

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

DRAFT: FOR PREVIEW ONLY

fully qualified domain name (FQDN)
handle
Server Message Block (SMB)

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

block
content server
peer
segment
segment hash of data (HoD)

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

client (client-role peer)
download schedule session
segment retrieval session
server (server-role peer)

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

PeerDist

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

HoHoDk
hosted cache

The following terms are defined in [MS-SMB2]:

Tree Connect

The following terms are specific to this document:

BranchCache™: A Windows Content Caching and Retrieval feature that enables **content** from File and Web servers on a wide area network (WAN) to be cached on computers at a local branch office. This feature is available in two modes: hosted cache and **distributed cache**.

content: When cached, **content** is identified by **segment** and downloaded in **blocks**.

content block: A **block** of data in the **content** that can be retrieved from clients.

Client-Side Caching (CSC): A local cache, also known as offline files, on a computer running Windows.

DFS File Client: See [MS-GLOS] definition of **Distributed File System (DFS) client**.

distributed cache: A cache composed of **blocks** of data hosted on multiple **peers** acting in cooperation.

File Access Protocol: A protocol that enables remote access to a portion of a local Object Store and that supports file system semantics. In this document, this term refers to the SMB 2.1/2 access protocols and HTTP/HTTPS protocols.

file client: Instance of an **NFS File Client**, **SMB File Client**, or **DFS File Client**.

file handle: A general term used to refer to the SMB2_FILEID packet ([MS-SMB2] section 2.2.14.1). It represents an open file on the server that is often referred to as File ID or file id.

DRAFT: FOR PREVIEW ONLY

A **file handle** is returned from an **SMB2 Open** or **SMB2 Create** operation and is unique within an SMB2 connection.

file server: Computer hosting one or more instances of a **File Service**.

File Service: Instance of an **NFS File Service** and/or an **SMB File Service**.

hash: A hash (such as SHA-1) on the **content** or **content block**.

hash list: A list of **hashes** that comprise the **block** **hashes** plus the **content hash**.

HMAC: Keyed-Hashing for Message Authentication. For more information, see [RFC2104].

metadata: A generic term for a **hash** or **hash list**.

NFS: Refers collectively to version 2 or version 3 of the **NFS** protocol (that is, **NFS Access Protocols**) as well as the related IETF protocols. This includes, specifically, the following standards:[RFC4506], [RFC1057], [RFC1094], [RFC1813], [RFC5531], and [RFC1833].

NFS Access Protocols: Refers collectively to **NFS** version 2 [RFC1094], **NFS** version 3 [RFC1813], and NLM/NSM [C702].

NFS File Client: A **service** that implements client side functionality of the **NFS Access Protocols** and exposes it to applications.

NFS File Service: A **service** on the **file server** that provides access to files using some version of the **NFS Access Protocols** in combination with related IETF protocols.

SMB Access Protocols: Refers collectively to protocols defined in [MS-CIFS], [MS-SMB], [MS-SMB2], and [MS-FSCC].

SMB File Client: A **service** that implements client side functionality of the **SMB Access Protocols**, and exposes it to applications.

SMB File Service: A **service** on the **file server** which provides access to files using the **SMB Access Protocols** and related protocols.

1.3 References

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

[C702] The Open Group, "Protocols for Interworking: XNFS, Version 3W", C702, February 1998, http://www.opengroup.org/public/pubs/catalog/c702.htm

[FIPS180-2] Federal Information Processing Standards Publication, "Secure Hash Standard", FIPS PUB 180-2, August 2002, http://csrc.nist.gov/publications/fips/fips180-2/fips180-2.pdf

[FIPS197] National Institute of Standards and Technology, "Federal Information Processing Standards Publication 197: Advanced Encryption Standard (AES)", November 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[MC-BUP] Microsoft Corporation, "Background Intelligent Transfer Service (BITS) Upload Protocol Specification".

[MS-AUTHSOD] Microsoft Corporation, "Authentication Services Subsystem Overview Document".

DRAFT - FOR PREVIEW ONLY

[MS-BPCR] Microsoft Corporation, "Background Intelligent Transfer Service (BITS) Peer-Caching: Content Retrieval Protocol Specification".

[MS-BPDP] Microsoft Corporation, "Background Intelligent Transfer Service (BITS) Peer-Caching: Peer Discovery Protocol Specification".

[MS-CERSOD] Microsoft Corporation, "Certificate Services Overview Document".

[MS-CIFS] Microsoft Corporation, "Common Internet File System (CIFS) Protocol Specification".

[MS-FASOD] Microsoft Corporation, "File Access System Overview Document", to be published 2011.

[MS-FSA] Microsoft Corporation, "File System Algorithms".

[MS-FSCC] Microsoft Corporation, "File System Control Codes".

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

[MS-GPSOD] Microsoft Corporation, "Group Policy System Overview", to be published 2011.

[MS-PCCRC] Microsoft Corporation, "Peer Content Caching and Retrieval: Content Identification".

[MS-PCCRD] Microsoft Corporation, "Peer Content Caching and Retrieval Discovery Protocol Specification".

[MS-PCCRR] Microsoft Corporation, "Peer Content Caching and Retrieval: Retrieval Protocol Specification".

[MS-PCCRTP] Microsoft Corporation, "Peer Content Caching and Retrieval: Hypertext Transfer Protocol (HTTP) Extensions".

[MS-PCHC] Microsoft Corporation, "Peer Content Caching and Retrieval: Hosted Cache Protocol Specification".

[MS-SMB] Microsoft Corporation, "Server Message Block (SMB) Protocol Specification".

[MS-SMB2] Microsoft Corporation, "Server Message Block (SMB) Version 2 Protocol Specification".

[MS-TLSP] Microsoft Corporation, "Transport Layer Security (TLS) Profile".

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980, http://www.ietf.org/rfc/rfc768.txt

[RFC793] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981, http://www.ietf.org/rfc/rfc0793.txt

[RFC1001] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods", STD 19, RFC 1001, March 1987, http://www.ietf.org/rfc/rfc1001.txt

[RFC1002] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications", STD 19, RFC 1002, March 1987, http://www.ietf.org/rfc/rfc1002.txt

[RFC1057] Sun Microsystems, Inc., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 1057, June 1998, http://www.ietf.org/rfc/rfc1057.txt

DRAFT - FOR PREVIEW ONLY

[RFC1094] Sun Microsystems, Inc., "NFS: Network File System Protocol Specification", RFC 1094, March 1989, http://www.ietf.org/rfc/rfc1094.txt

[RFC1813] Callaghan, B., Pawlowski, B., and Staubach, P., "NFS Version 3 Protocol Specification", RFC 1813, June 1995, http://www.ietf.org/rfc/rfc1813.txt

[RFC1833] Srinivasan, R., "Binding Protocols for ONC RPC Version 2", RFC 1833, August 1995, http://www.ietf.org/rfc/rfc1833.txt

[RFC2104] Krawczyk, H., Bellare, M., and Canetti, R., "HMAC: Keyed-Hashing for Message Authentication", RFC 2104, February 1997, http://www.ietf.org/rfc/rfc2104.txt

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000, http://www.ietf.org/rfc/rfc2743.txt

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.ietf.org/rfc/rfc2818.txt

[RFC4506] Network Appliance, Inc., "XDR: External Data Representation Standard", STD 67, RFC 4506, May 2006, http://www.ietf.org/rfc/rfc4506.txt

[RFC4559] Jaganathan, K., Zhu, L., and Brezak, J., "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006, http://www.ietf.org/rfc/rfc4559.txt

[RFC5531] Thurlow, R., "RPC: Remote Procedure Call Protocol Specification Version 2", RFC 5531, May 2009, http://www.rfc-editor.org/rfc/rfc5531.txt

[WS-Discovery] Beatty, J., Kakivaya, G., Kemp D., et al., "Web Services Dynamic Discovery (WS-Discovery)", April 2005, http://specs.xmlsoap.org/ws/2005/04/discovery/ws-discovery.pdf

If you have any trouble finding [WS-Discovery], please check here.

DRAFT: FOR PREVIEW ONLY

2 Functional Architecture

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

2.1 Overview

2.1.1 System Purpose

The purpose of the Content Caching and Retrieval System is to enable the retrieval of content from networked computers. It supports content discovery, transport, data structures used for content, and the encryption process for securing content. Within the system, caching has two distinct modes of operation. One, in which the content cache is on a single predetermined computer, is known in this document as a "hosted cache" and is a client-server model.<1> The other mode is one in which the cached content is distributed among a number of computers, known in this document as "**distributed cache**", and is a peer-to-peer caching model.<2>

The member protocols that make up the system are:

- MS-FSCC
- MS-PCCRC
- MS-PCCRD
- MS-PCCRR
- MS-PCHC
- MS-PCCRTP
- MS-SMB2

2.1.2 Functional Overview

This section describes the conceptual data organization the system maintains to provide its functionality. The data model described in this section can be implemented in a variety of ways. This document does not prescribe any specific implementation technique.

The purpose of this section is to document shared state that is maintained between the Content Caching and Retrieval System member protocols and/or external systems in order to support operations involving more than one protocol.

The sharing of state happens internal to the server implementation. Compatible implementations can use any mechanism of their choice to perform the sharing, as long as it satisfies the requirements of this section and the protocol specifications. For example, an implementation could use a single store that is shared across all of the protocols, or it could use multiple stores (one per protocol) and build a synchronization mechanism to maintain consistency across all of the stores, as long as that synchronization mechanism ensures that all atomicity requirements are observed and that two protocols that share state always see an identical view of that shared state.

2.1.2.1 Black Box Diagram

The Content Caching and Retrieval System is shown in the following figure as a "black box", showing a high-level abstraction of the Content Caching and Retrieval System's major components and the external systems and components with which the system interacts.

DRAFT: FOR PREVIEW ONLY

The abstract components of the Content Caching and Retrieval System are as follows.

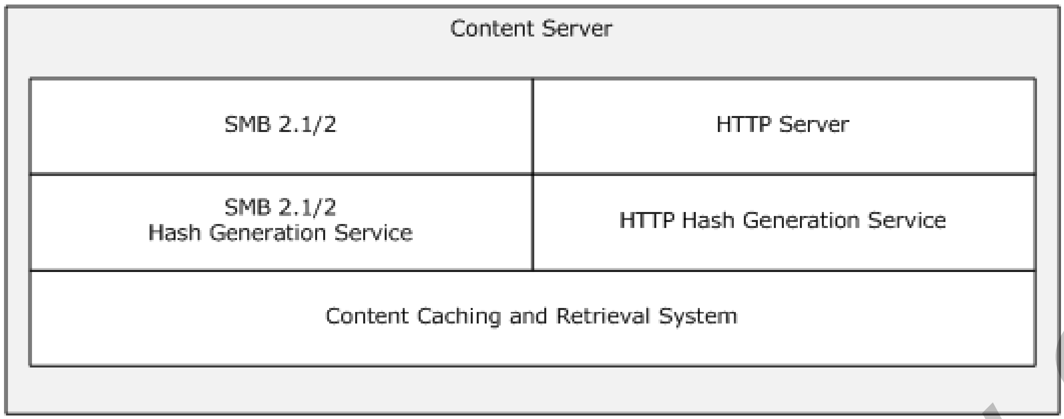

Content Server
SMB 2.1/2
HTTP Server
SMB 2.1/2 Hash Generation Service
HTTP Hash Generation Service
Content Caching and Retrieval System

Figure 2: Abstraction view of a content server

A content server file is an abstraction of the software running on a file and Web server that provides remote file access to one or more users. The abstraction can be considered to be made up of two subsystems: Block File Services (SMB 2.1/2) and Streaming File Services (HTTP). A content server can be managed by one or more Administrators. Internally, the content server contains an object store that contains the static content made available through the SMB 2.1/2 and HTTP protocols. The object store is an implementation-dependent local file system.

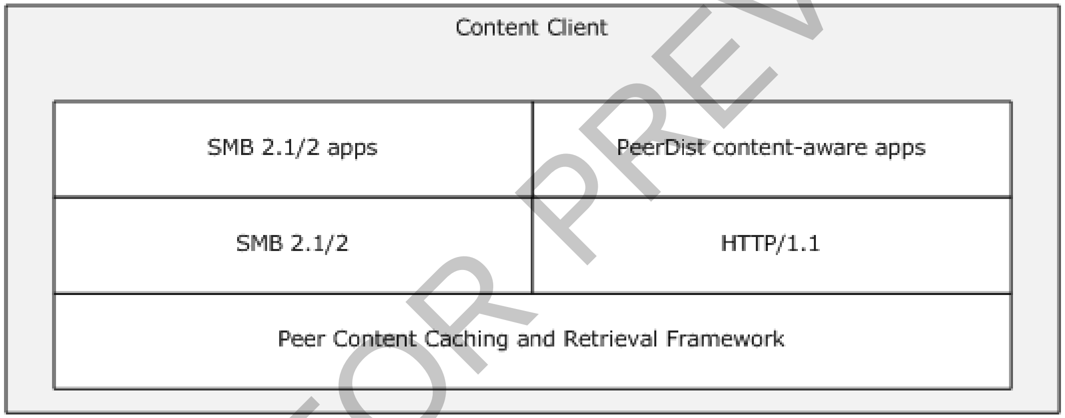

Content Client
SMB 2.1/2 apps
PeerDist content-aware apps
SMB 2.1/2
HTTP/1.1
Peer Content Caching and Retrieval Framework

Figure 3: Abstraction view of a content client

Content client and Admin Client are abstractions of low-level protocol state and operating software that runs in application and administrative nodes, respectively. These clients are considered part of the Content Caching and Retrieval System. They are used by Application and Administrator Tool software to effect communication with components of the System.

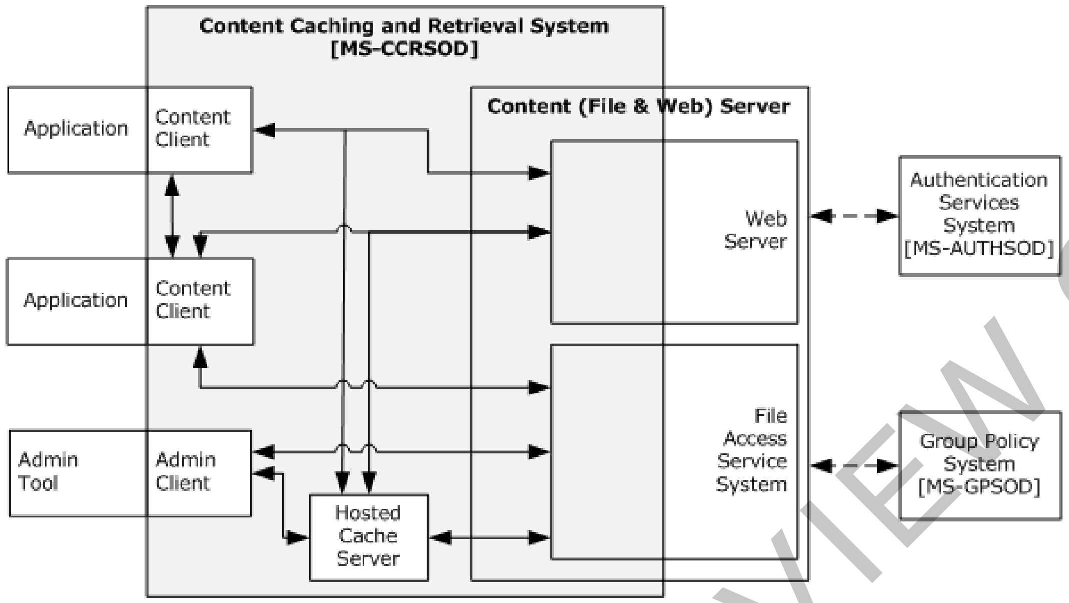
The Application and Administrator Tool represent programs with which a user and an Administrator typically interact, respectively. It should be understood that this is a somewhat arbitrary distinction, and that many programs have both Application and Administrator Tool features.

The dotted lines model dependency and influence relationships between the Content Caching and Retrieval System and external systems and components.

DRAFT: FOR PREVIEW ONLY

Copyright © 2011 Microsoft Corporation.

The Content Caching and Retrieval System has two dependencies. It depends on the File Access Services System for SMB 2.1/2, and it depends on a Web Server Service for HTTP. Other external relationships are influences, because the system may, depending on the configuration, operate with reduced functionality if the external service is not present.



Content Caching and Retrieval System [MS-CCRSOD]
Application
Content Client
Application
Content Client
Admin Tool
Admin Client
Hosted Cache Server
Content (File & Web) Server
Web Server
File Access Service System
Authentication Services System [MS-AUTHSOD]
Group Policy System [MS-GPSOD]

Figure 4: Content Caching and Retrieval System black box diagram

2.1.2.2 White Box Diagram

The following figure shows the protocol layering relationships for the Content Caching and Retrieval System member protocols. The default relationship, indicated by a solid arrow, is "is transported by". The "includes" notation means that a protocol document includes a second document by reference (for example, [MS-SMB2] includes [MS-FSCC]). Member protocols are shown in shaded boxes.

Content Caching and Retrieval can be initiated both by SMB2 and HTTP. A Background Intelligent Transfer Service (BITS) client can use the system and acts as an HTTP client.

A content client uses HTTPS for communication with a hosted cache server specifically while offering content. The content is then retrieved using the Peer Content Caching and Retrieval: Retrieval Protocol, as specified in [MS-PCCRR].

A content client uses the Peer Content Caching and Retrieval Discovery Protocol, as specified in [MS-PCCRD] (an implementation of [WS-Discovery]), to locate peer computers with cached content.

The majority of traffic in the system is performed by the Peer Content Caching and Retrieval: Retrieval Protocol [MS-PCCRR]. This protocol is used to transfer actual content regardless of the protocol that retrieved the content **metadata**.

DRAFT: FOR PREVIEW ONLY

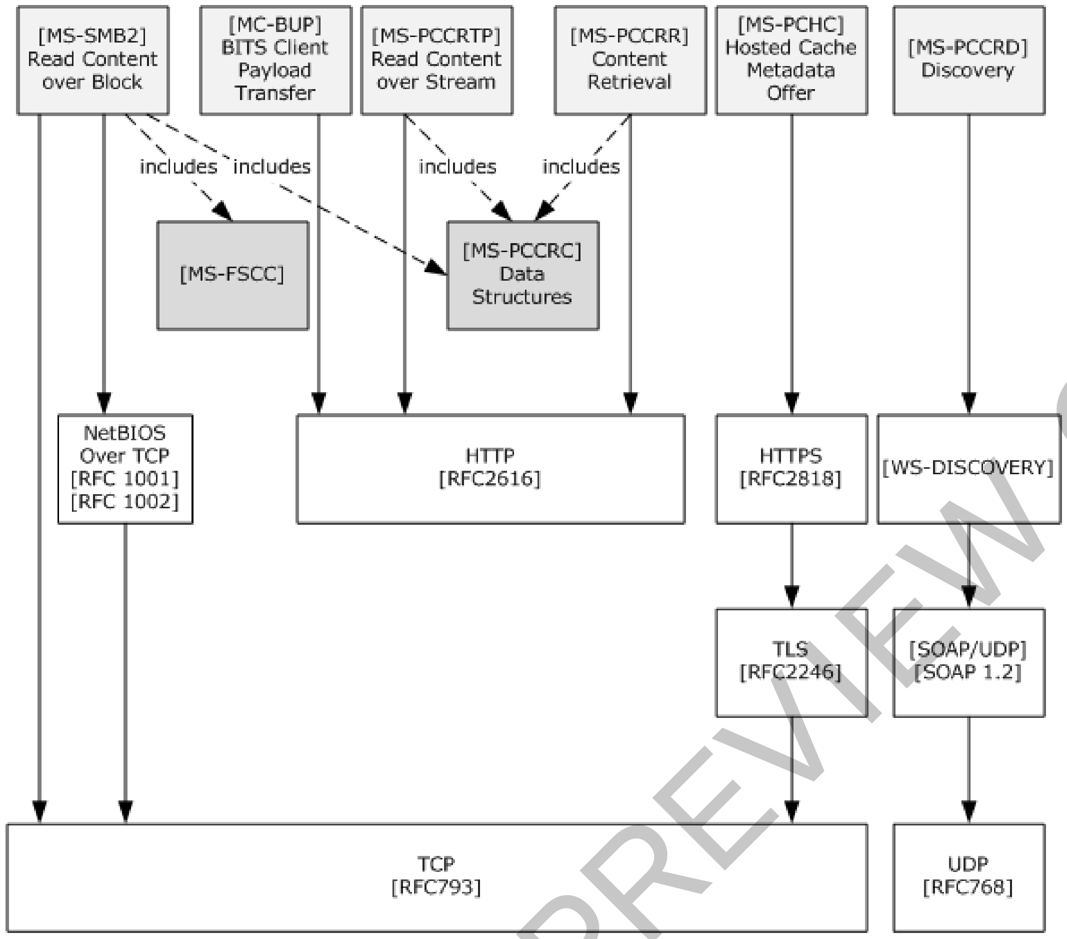

[MS-SMB2] Read Content over Block
[MC-BUP] BITS Client Payload Transfer
[MS-PCCRTP] Read Content over Stream
[MS-PCCRR] Content Retrieval
[MS-PCHC] Hosted Cache Metadata Offer
[MS-PCCRD] Discovery
includes
includes
includes
includes
[MS-FSCC]
[MS-PCCRC] Data Structures
NetBIOS Over TCP [RFC 1001] [RFC 1002]
HTTP [RFC2616]
HTTPS [RFC2818]
[WS-DISCOVERY]
TLS [RFC2246]
[SOAP/UDP] [SOAP 1.2]
TCP [RFC793]
UDP [RFC768]

Figure 5: Protocol relationships for the Content Caching and Retrieval System

DRAFT: FOR PREVIEW ONLY

2.1.2.2.1 HTTP Metadata Retrieval Integration

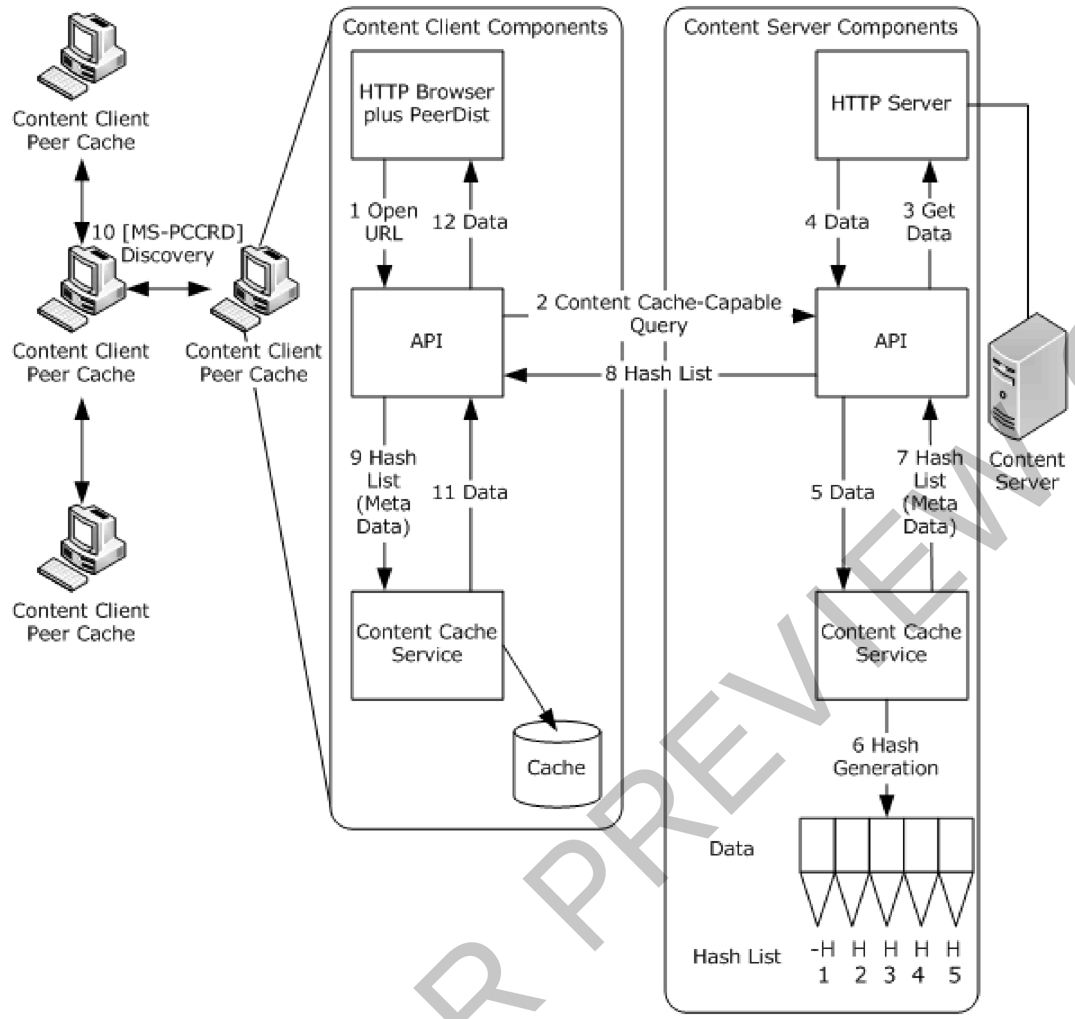


Figure 6: HTTP metadata retrieval integration

The sequence of messages for an HTTP request with Content Caching and Retrieval is shown in the preceding figure and is as follows:

1. The Client Application opens a URL.
2. The client side adds a **PeerDist** header to an outgoing HTTP request.
3. The HTTP server side performs any required access checks, and then, if valid, requests the data. If **hashes** are available, these are returned to the client.
4. If hashes are not available, a normal HTTP response is returned immediately to the client.
5. If the requested data and hashes are available, the data is placed in the distributed cache.
6. Metadata in the form of **hash lists** is generated, making the metadata available for any subsequent requests on the same data.

DRAFT: FOR PREVIEW ONLY

7. Any subsequent request for the same data identified by the URL results in metadata being returned.

8. Metadata is returned to the content client using the Peer Content Caching and Retrieval Hypertext Transfer Protocol (HTTP) Extensions, as specified in [MS-PCCRTP].

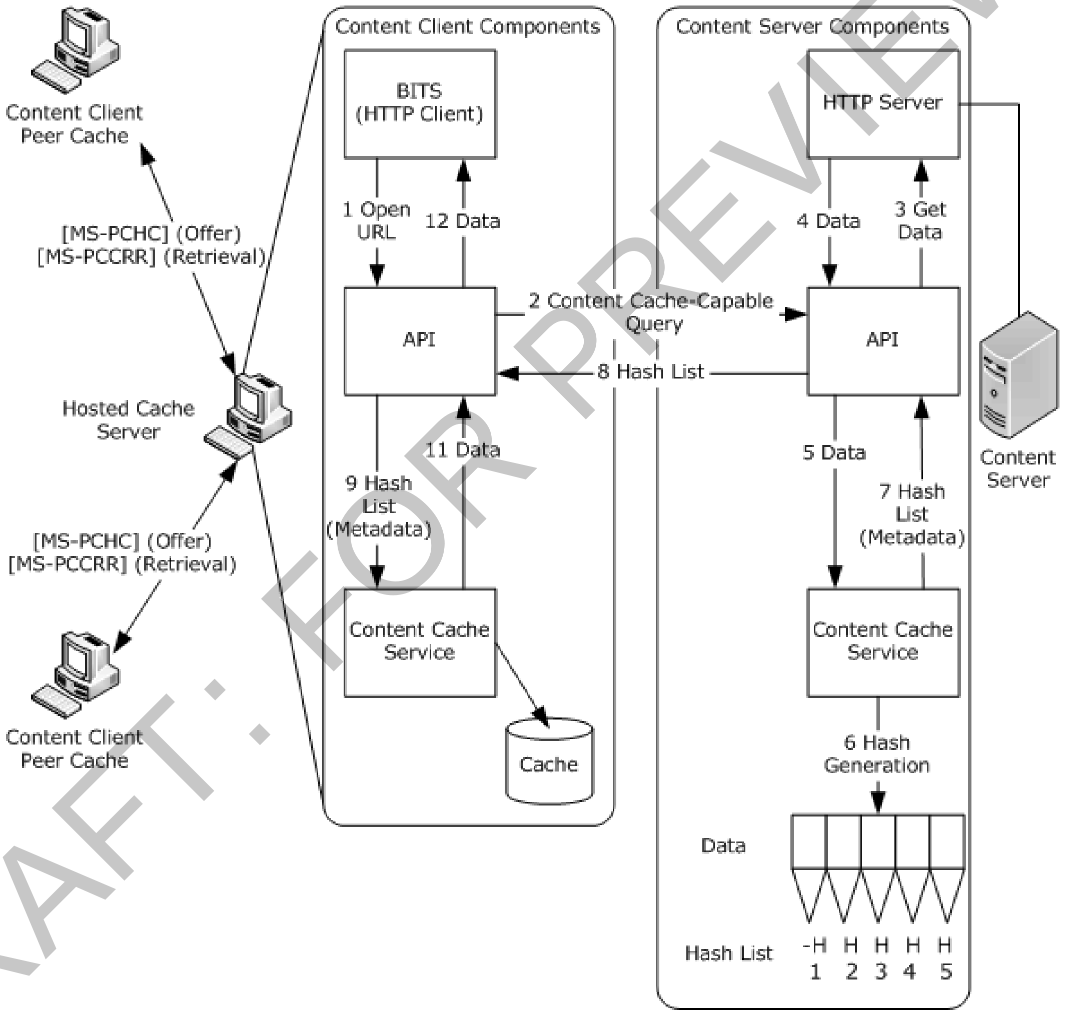
9. The hash list is passed to the BranchCache Service to look up the data.

10. For unsuccessful cache lookups (missing block), a range request, including the byte range, is broadcast using the Peer Content Caching and Retrieval Discovery Protocol, as specified in [MS-PCCRD]. If data is found in a peer client, the data is retrieved using the Peer Content Caching and Retrieval: Retrieval Protocol [MS-PCCRR].

11. If data is retrieved, it is added to the cache and the hash list made available for discovery. If the client does not receive the associated data (that is, the original content is unavailable), a suitable (HTTP) error is returned to the application.

12. If the data is retrieved, it is returned to the requesting content client.

2.1.2.2.2 BITS Integration



DRAFT: FOR PREVIEW ONLY

Figure 7: BITS integration

In the context of Content Caching and Retrieval, BITS is a client of the HTTP service. The figure in this section shows the hosted cache configuration, but distributed cache is equally valid and would be the same as the figure in the previous section, which depicts HTTP metadata retrieval integration, with the web browser replaced by a BITS client.

The sequence of messages for a BITS request with Content Caching and Retrieval are in fact HTTP requests. The main difference is that the BITS client makes extensive use of HTTP range requests (see [RFC2616] section 3.12) because BITS is mainly used for file transfer and may be subject to interruption, pause, and continuation actions. The higher-layer protocol provides BITS with the desired range(s) of the server URL; BITS then issues single- or multi-range HTTP requests that represent a subset of the desired ranges.

The following sequence is for a hosted cache configuration:

1. The Client Application opens a URL.
2. BITS, on the client side, adds a PeerDist header to the outgoing HTTP requests with.
3. The Web server performs any required access checks. If hashes are available, these are returned to the client.
4. If hashes are not available, a response is returned immediately to the client.
5. If the requested data and the hashes are available, the data is placed in the distributed cache.
6. Metadata, in the form of hash lists, is generated, making the metadata available for any subsequent requests on the same data.
7. Any subsequent request for the same data identified by the URL results in metadata being returned.
8. Metadata (hash list) is returned to the content client by using the Peer Content Caching and Retrieval Hypertext Transfer Protocol (HTTP) Extensions, as specified in [MS-PCCRTP].
9. BITS receives a response comprising a hash list. The hash list is passed to the BranchCache Service to look up the data.
10. For unsuccessful local cache lookups (missing block), a direct request for the data from the hosted cache is made using the Peer Content Caching and Retrieval: Retrieval Protocol [MS-PCCRR].
11. If BITS does not receive the associated data (because the original content is unavailable), it returns a suitable (HTTP) error to the application. If the original data is available, it is obtained from the content server.
12. The requested data is returned to the requesting content client.

Note Prior to the introduction of the Content Caching and Retrieval System, BITS clients used the Background Intelligent Transfer Service (BITS) Peer-Caching: Peer Discovery Protocol, as specified in [MS-BPDP], for discovery and the Background Intelligent Transfer Service (BITS) Peer-Caching: Content Retrieval Protocol, as specified in [MS-BPCR], for retrieval.<3> The Background Intelligent Transfer Service (BITS) Peer-Caching: Peer Discovery and Background Intelligent Transfer Service (BITS) Peer-Caching: Content Retrieval protocols cannot coexist with the Content Caching and Retrieval System.

DRAFT - FOR PREVIEW ONLY

2.1.2.2.3 SMB 2.1/2 Metadata Retrieval Integration

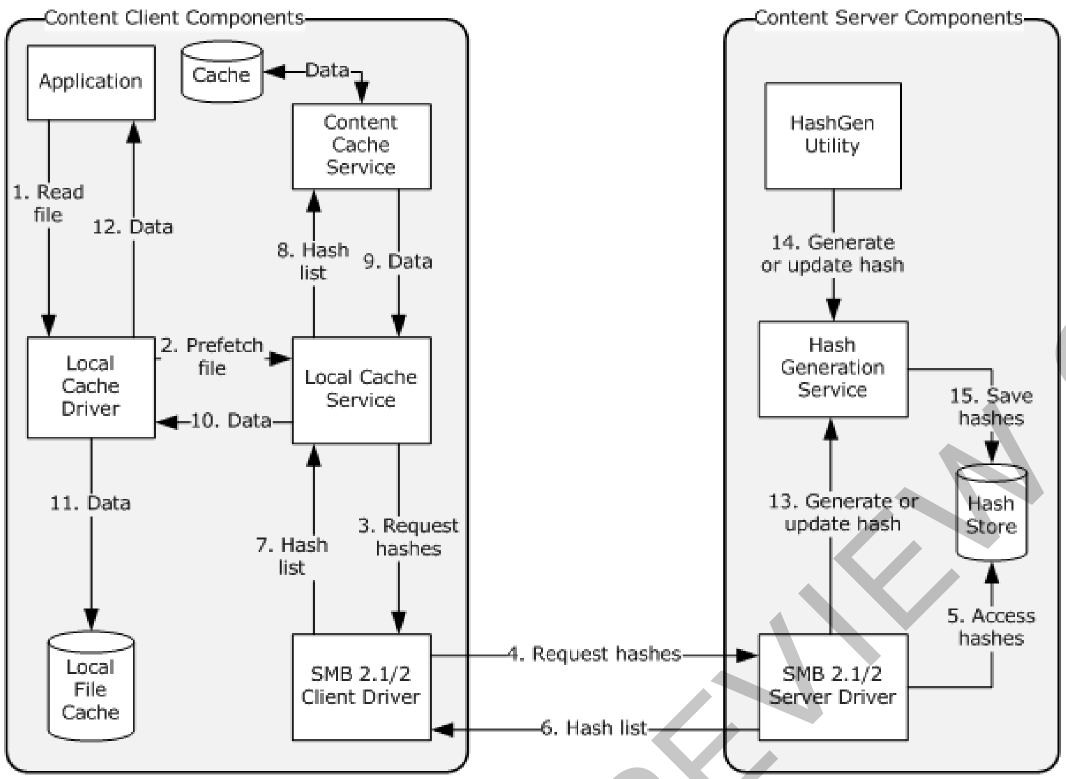


Figure 8: SMB 2.1/2 Metadata Retrieval Integration

The sequence of messages for an SMB 2.1/2 request with Content Caching and Retrieval is shown in the preceding figure and is as follows:

1. The Client Application opens a remote file. During this stage an SMB 2.1/2 **Tree Connect** allows the content client to determine whether the remote share supports hash lists.
2. An attempt is first made to retrieve the file from the local cache.
3. If the file is unavailable and the share supports hash lists, the local cache service requests the hash list.
4. The SMB 2.1/2 client requests hashes.
5. The SMB 2.1/2 Server Driver accesses the hash lists.
6. The SMB 2.1/2 Server Driver returns the hash list to the SMB 2.1/2 client.
7. The SMB 2.1/2 Client Driver returns the hash list to the local cache service.
8. The local cache service supplies the hash list to the content cache service to retrieve the data.
9. The data, if available, is returned to the local cache service. For unsuccessful local cache lookups (missing blocks), a request is broadcast using the Peer Content Caching and Retrieval Discovery Protocol [MS-PCCRD]. If data is found in a peer client, the data is retrieved using the Peer Content Caching and Retrieval: Retrieval Protocol [MS-PCCRR].

DRAFT: FOR PREVIEW ONLY

10.The data is placed in the content cache and delivered to the local cache service driver.

11.The data is placed in the local cache.

12.The data is returned to the Client Application.

13.If no hash list was available and the data was directly returned from the content server, then hash list generation is initiated on the content server.

14.The hash list may be independently created by running the hashgen utility on the content server.

15.The generated hash list is stored, making it available for future client requests.

2.1.2.2.4 PCCRD and WS-Discovery

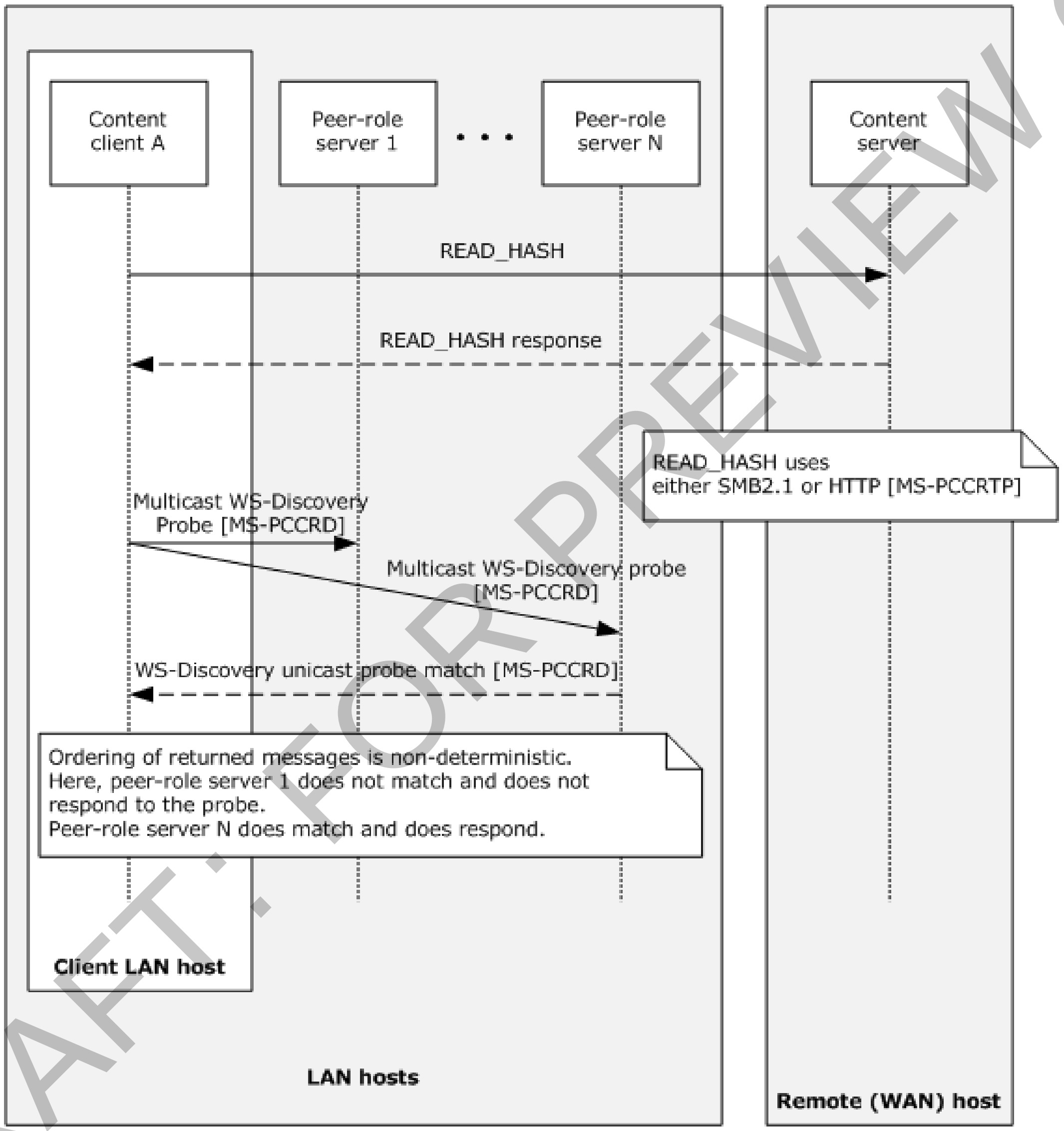

Content client A
Peer-role server 1
Peer-role server N
Content server
READ_HASH
READ_HASH response
READ_HASH uses either SMB2.1 or HTTP [MS-PCCRTP]
Multicast WS-Discovery Probe [MS-PCCRD]
Multicast WS-Discovery probe [MS-PCCRD]
WS-Discovery unicast probe match [MS-PCCRD]
Ordering of returned messages is non-deterministic. Here, peer-role server 1 does not match and does not respond to the probe. Peer-role server N does match and does respond.
Client LAN host
LAN hosts
Remote (WAN) host

Figure 9: PCCRD and WS-Discovery

DRAFT: FOR PREVIEW ONLY

The figure in this section shows the ordering of messages when the service is operating in distributed cache mode. The initial request (READ_HASH) for the content takes place in either SMB 2.1/2 or HTTP. The response with the hashes comes back in the same protocol that makes the request. After the content hashes have been received by the content client, the content client can determine whether any of the content exists within the LAN where it resides. The content client does this by checking any local machine cache; then, if the data is not available locally, it performs a Multicast WSD/MS-PCCRD Probe message with the hashes of the content.

Any server-role peers on the LAN that receive the Multicast Probe message and have the matching content via the Peer Content Caching and Retrieval Discovery Protocol ([MS-PCCRD]) respond with a Unicast Probe-Match message.

After the content client receives a match for content, it will initiate one or more Peer Content Caching and Retrieval: Retrieval Protocol ([MS-PCCRR]) sessions (not shown in the figure) to retrieve the content from the server-role peer.

2.1.3 Applicability

The Content Caching and Retrieval System is supplementary to the File Access Services; it is a form of WAN link acceleration. The goal is to increase network utilization. The major principle employed by Content Caching and Retrieval is the reduction of traffic across a WAN. Content caching enables content from file and Web servers on one end of a WAN to be cached on computers at the other end of the WAN.

2.1.4 Relevant Standards

Advanced Encryption Standard as specified in [FIPS197].

Hypertext Transfer Protocol - HTTP/1.1 as specified in [RFC2616].

Hypertext Transfer Protocol - HTTP/1.1 over TLS as specified in [RFC2818].

Protocol Standard for a NetBIOS Service on a TCP/UDP Transport (NetBIOS over TCP), as specified in [RFC1001] and [RFC1002].

Secure Hash Standard as specified in [FIPS180-2].

Transmission Control Protocol (TCP) as specified in [RFC793].

User Datagram Protocol (UDP) as specified in [RFC768].

Web Services Dynamic Discovery as specified in [WS-Discovery].

SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows as specified in [RFC4559].

2.1.5 Protocol Summary

The following table provides a comprehensive list of the Member Protocols of the Content Caching and Retrieval System.

Protocol name	Description	Short name
Peer Content Caching and Retrieval: Content	Specifies a binary data structure used in the Content Caching and Retrieval. The primary role in the Content Caching and Retrieval	[MS-PCCRC]

DRAFT: FOR PREVIEW ONLY

Protocol name	Description	Short name
Identification	System is Content Identification.	
Peer Content Caching and Retrieval Discovery Protocol	Specifies a multicast to discover and locate services based on the Web Services Dynamic Discovery (WS-Discovery) protocol [WS-Discovery]. There are two modes of operations in WS-Discovery: client-initiated probes and service-initiated announcements; both are sent through IP multicast to a predefined group. The primary role in the Content Caching and Retrieval System is Content Discovery.	[MS-PCCRD]
Peer Content Caching and Retrieval: Retrieval Protocol	Specifies the messages that are necessary for querying Peer-role servers or a **Hosted cache** server for the availability of certain content, and for retrieving the content. The primary role in the Content Caching and Retrieval System is Content Retrieval.	[MS-PCCRR]
Peer Content Caching and Retrieval: Hosted Cache Protocol	Specifies an HTTPS-based mechanism for clients to notify a hosted cache server regarding the availability of content and for a hosted cache server to indicate interest in the content. The primary role in the Content Caching and Retrieval System is Content Notification.	[MS-PCHC]
Peer Content Caching and Retrieval: Hypertext Transfer Protocol (HTTP) Extensions	Specifies a content encoding known as PeerDist that is used by an HTTP/1.1 client and an HTTP/1.1 server to communicate content to each other. The primary role in the Content Caching and Retrieval System is metadata (hash) retrieval.	[MS-PCCRTP]
Server Message Block (SMB) Version 2.1 Protocol	Specifies a metadata retrieval mechanism. Version 2.1 of this protocol has enhancements for the detection of content caching-enabled shares and retrieval of metadata related to content caching. The primary role in the Content Caching and Retrieval System is metadata (hash) retrieval.	[MS-SMB2]

The following table provides a comprehensive list of the Member Protocols of the Content Caching and Retrieval System. The Member Protocols are grouped according to their primary purpose.

Protocol name	Group description	Short name
	Metadata (Hash) Retrieval	
Server Message Block (SMB) Version 2.1 Protocol	SMB 2.1 and HTTP are the current protocols that are enabled for metadata retrieval.	[MS-SMB2]
Peer Content Caching and Retrieval: Hypertext Transfer Protocol (HTTP) Extensions	Specifies a content encoding known as PeerDist that is used by an HTTP/1.1 client and an HTTP/1.1 server to communicate content to each other. The primary role in the Content Caching and Retrieval System is metadata (hash) retrieval.	[MS-PCCRTP]
	File Retrieval	
Server Message Block (SMB) Version 2.1 Protocol	SMB2 version 2.1 and HTTP (as specified in [MS-PCCRTP] specifies content encoding over HTTP) are the two protocols used for file retrieval that are content caching aware.	[MS-SMB2]
Peer Content Caching and	Specifies a content encoding known as PeerDist that is used by	[MS-

DRAFT: FOR PREVIEW ONLY

Protocol name	Group description	Short name
Retrieval: Hypertext Transfer Protocol (HTTP) Extensions	an HTTP/1.1 client and an HTTP/1.1 server to communicate content to each other. The primary role in the Content Caching and Retrieval System is metadata (hash) retrieval.	PCCRTP]
	Content Identification	
Peer Content Caching and Retrieval: Content Identification	A binary data structure that is used for content identification.	[MS-PCCRC]
	Content Discovery	
Peer Content Caching and Retrieval Discovery Protocol	Based on the Web Services Dynamic Discovery (WS-Discovery) protocol [WS-Discovery]. There are two modes of operations in WS-Discovery: client-initiated probes and service-initiated announcements; both are sent through IP multicast to a predefined group.	[MS-PCCRD]
	Content Retrieval	
Peer Content Caching and Retrieval: Retrieval Protocol	Content in the form of blocks is transferred using HTTP, with the message format specified in [MS-PCCRR].	[MS-PCCRR]
	Authentication	
Transport Layer Security (TLS)	Hosted Server Authentication uses HTTPS for secure transport of hosted content as specified in [MS-PCHC]. [MS-TLSP] describes the Microsoft Windows® implementation of TLS.	[MS-TLSP]
RFC 4559	Client Authentication [RFC4559]. When a hosted cache server is used, client authentication can be enabled.	[RFC4559]

2.2 Environment

The following sections identify the context in which the system exists. This includes the systems that use the interfaces provided by this system of protocols, other systems that depend on this system, and, as appropriate, how components of the system communicate.

2.2.1 Dependencies on This System

There are no systems that depend on the Content Caching and Retrieval System.

2.2.2 Dependencies on Other Systems/Components

Network Infrastructure: This system requires access to network services that support:

- TCP/IP (IPv4 or IPv6).
- UDP/IP (IPv4 or IPv6).
- Domain Naming System (DNS) name resolution.

DRAFT: FOR PREVIEW ONLY

Cache mode	Protocol	Activity	Port
Distributed and Hosted	HTTP	Content Retrieval (uses HTTP)	TCP Port 80
Hosted	HTTPS	Content Offering (HTTPS)	TCP Port 443
Distributed	WS-Discovery	Peer Discovery (Uses WSD)	UDP port 3702

Object Store: Both SMB2 and HTTP File Services require access to a hierarchical object store for persistence of files and namespace. The Object Store is typically built on file system(s) that are available in the host operating system, but the Object Store may also need to include functionality in addition to that provided by the file system. A File Service may need to implement additional logic to convert between the semantics of a particular **File Access Protocol** and the semantics of the available Object Store. Some semantics implemented in Microsoft Windows® file systems are directly visible when using the SMB Access Protocols, and such wire-visible behaviors are documented in [MS-FSA].

Case Sensitivity: The Object Store supports case-insensitive operation for the SMB File Service and should support case-sensitive operation for the HTTP File Service.

Accounts: If computers hosting file clients and file services are not joined to a domain, then user accounts must be configured across computers by some external mechanism.

Hosted Cache: If a hosted cache is to be used, the location of the Hosted Cache (DNS Name or IP Address) and the Hosted Cache Listen and Connections Ports is configured on the client computers.

2.2.3 System Influences

The Content Caching and Retrieval System can be influenced by the external systems and components shown in the following table.

External entity	Content Caching and Retrieval System depends on external entity for	Consequences if absent
Authentication Services System	Authenticating client and server principals.	Centralized identity management does not work if the Domain Interaction System is not available.
Group Policy System	Configuration of individual capabilities within the Content Caching and Retrieval System.	Cannot centrally configure some functionality of the system.
Domain Interaction System	Content Caching and Retrieval can operate either in a Domain or in a Workgroup. Domain services are required for Hosted Cache client authentication.	Hosted Cache client authentication cannot function.
File Access System	File Access services to which Content Caching and Retrieval is an adjunct.	No file access and therefore no Content Caching and Retrieval.
Web Server	HTTP Content Caching is dependent on the availability of HTTP served content.	No HTTP content caching can be performed.
Certification Authority System	Hosted Cache mode makes use of SSL; this requires an X.509 certificate.	Hosted Cache communication will fail.

Specific-system influences are as follows:

Copyright © 2011 Microsoft Corporation.

Release: Sunday, October 16, 2011

DRAFT: FOR PREVIEW ONLY

Group Policy enables a client to interact with the Content Cache Service, which is off by default. This enables a computer to act as both a client-role peer and client-role server, thereby publishing and retrieving content and metadata obtained from a content server.

The Background Intelligent Transfer Service (BITS) Upload Protocol, as defined in [MC-BUP], specifies an HTTP 1.1-based upload protocol. This protocol is used to transfer large payloads from a client to a server or a server to a client over networks with frequent disconnections and to send notifications about the availability of uploaded payloads. This protocol is a client of the Content Caching and Retrieval System.

2.3 Assumptions and Preconditions

The following assumptions and preconditions need to be satisfied for the Content Caching and Retrieval System to operate successfully:

System Availability:

The File Access Services System must be installed on all the computers involved in content caching.

A PeerDist-capable (see[MS-PCCRTP]) web server must be installed and configured on the content server for HTTP caching.

A PeerDist-capable (see[MS-PCCRTP]) web browser or BITS client must be installed on client computers for HTTP caching.

The Content Caching and Retrieval System components (**BranchCache**) must be installed on the computers involved.

Authentication Services: Authentication services as described in [MS-AUTHSOD] are available to all file clients and file services.

Network Configuration: In order for system components running on different computers to communicate, the network services and infrastructure must be functional and correctly configured.

Domain Configuration: In a domain configuration, file clients and file services have access to directory services provided by the domain.

Domain Functionality: Domain functionality is not a requirement of the Content Caching and Retrieval System.

2.4 Use Cases

2.4.1 Actors

The actors that participate in the Content Caching and Retrieval System are:

User: The User is the principal that requires file access in order to read files on another computer. The User is referred to using the qualifiers "SMB2.1", "HTTP", or "BITS" when it is necessary to distinguish User instances. The User is external to the File Services System and the Content Caching and Retrieval System, and interacts through the Application. The Content Caching and Retrieval System only applies to the reading of existing files.

Administrator: The Administrator is the person who administers the content server and hosted cache server. The Administrator is interested in organizing content, setting access rights, and enabling content caching. The Administrator is external to the Content Caching and Retrieval System, and interacts with the System through the Administrator Tool.

DRAFT: FOR PREVIEW ONLY

Administrator Tool: The Administrator Tool is a program that offers management functionality to the Administrator by means of the Admin Client. Typical Administrator Tools are command line and graphical shells, management utilities, and graphical management programs. The Administrator Tool is external to the Content Caching and Retrieval System and makes use of the Admin Client to accomplish its work.

Application: The Application is a program that consumes file-reading services by means of the content client. Applications (where caching applies) need to open, read, and close files. The Application is external to the File Services System and the Content Caching and Retrieval System. The Application interacts with System through the content client.

Content Client: The content client implements client-side protocol components and consumes the file services that are offered by the content server. The content client can be referred to using the qualifier "SMB2.1", "HTTP", or "BITS" when it is necessary to distinguish Client instances. The content client is internal to the Content Caching and Retrieval System. A content client may additionally act as distributed cache peer.

Content Server: The content server's interest is to provide and maintain a secure and consistent File Access Service (see [MS-FASOD]) and to provide content metadata as part of the Content Caching and Retrieval System.

Hosted Cache Server: The Hosted Cache Server's interest is to cache content and to receive metadata regarding the availability of content segments and blocks, and then, as required, to download the segments and blocks from clients that have the relevant data. Later, when another client requests the content through a secure mechanism, the content can be retrieved from the **Hosted Cache** rather than from a content server.

Distributed Cache Peer: A Distributed Cache Peer's interest is to cache and distribute data and respond to queries regarding the availability of data segments and blocks. Then, when a client requests the content through a secure mechanism, the content can be obtained from the distributed cache rather than from a content server.

Object Store: The File Access Services System (and therefore the Content Caching and Retrieval System) is dependent on an external Object Store for storing files and directories.<4>

Wire-visible behavior of File Access Services protocols is not specified by the protocols themselves and is dependent on Object Store behavior.

2.4.2 Supporting Actors and System Interests Summary

File Access Services System [MS-FASOD]: The purpose of the File Access Services System is to allow a set of actors (people or processes) to access and share files located on a file server using a network between them in a secure and managed environment. The files may be distributed among a number of computers in a workgroup or domain, or centralized in one or more file server computers. The File Access Services System can optionally use the Content Caching and Retrieval System to cache content.

Group Policy System [MS-GPSOD]: The Group Policy System enables an administrator to maintain standard operating environments in domains for specific groups of users and computers. As software changes and policies change over time, Group Policy can be used to update an already-deployed standard operating environment. Computers participating in the Content Caching and Retrieval System within a domain can be expected to participate in, and be influenced by, the Group Policy System.

Certificate Services protocols system [MS-CERSOD]: The Certificate Authority component enables an administrator to request a certificate.

DRAFT: FOR PREVIEW ONLY

Background Intelligent Transfer Service (BITS) Upload Protocol, as defined in [MC-BUP], specifies an HTTP 1.1-based upload protocol. This protocol is used to transfer large payloads from a client to a server or server to client over networks with frequent disconnections, and to send notifications about the availability of uploaded payloads. "BITS" is a PeerDist-enabled HTTP client and can therefore act as a client of the Content Caching and Retrieval System.

2.4.3 Use Case Diagrams

The following table groups use cases that span the functionality of the Content Caching and Retrieval System. Detailed descriptions for these use cases are provided in section 2.4.4.

Use case group	Use cases
Configuring Content Caching and Retrieval Components	Configuring SMB 2.1/2 Content Server Caching Configuring HTTP Content Server Caching Configuring Content Client Caching Mode Configuring a Hosted Cache Server
Metadata Retrieval	Using SMB 2.1/2 Metadata Retrieval HTTP Metadata Retrieval BITS-HTTP Metadata Retrieval
Content Discovery and Retrieval	Content Discovery and Retrieval with Hosted Cache (Cached Data Unavailable) Content Discovery and Retrieval with Hosted Cache (Cached Data Available) Content Discovery and Retrieval with Distributed Cache (Cached Data Unavailable) Content Discovery and Retrieval with Distributed Cache (Cached Data Available)

The following use case diagrams illustrate the separate groups of use cases described in this section.

DRAFT: FOR PREVIEW ONLY

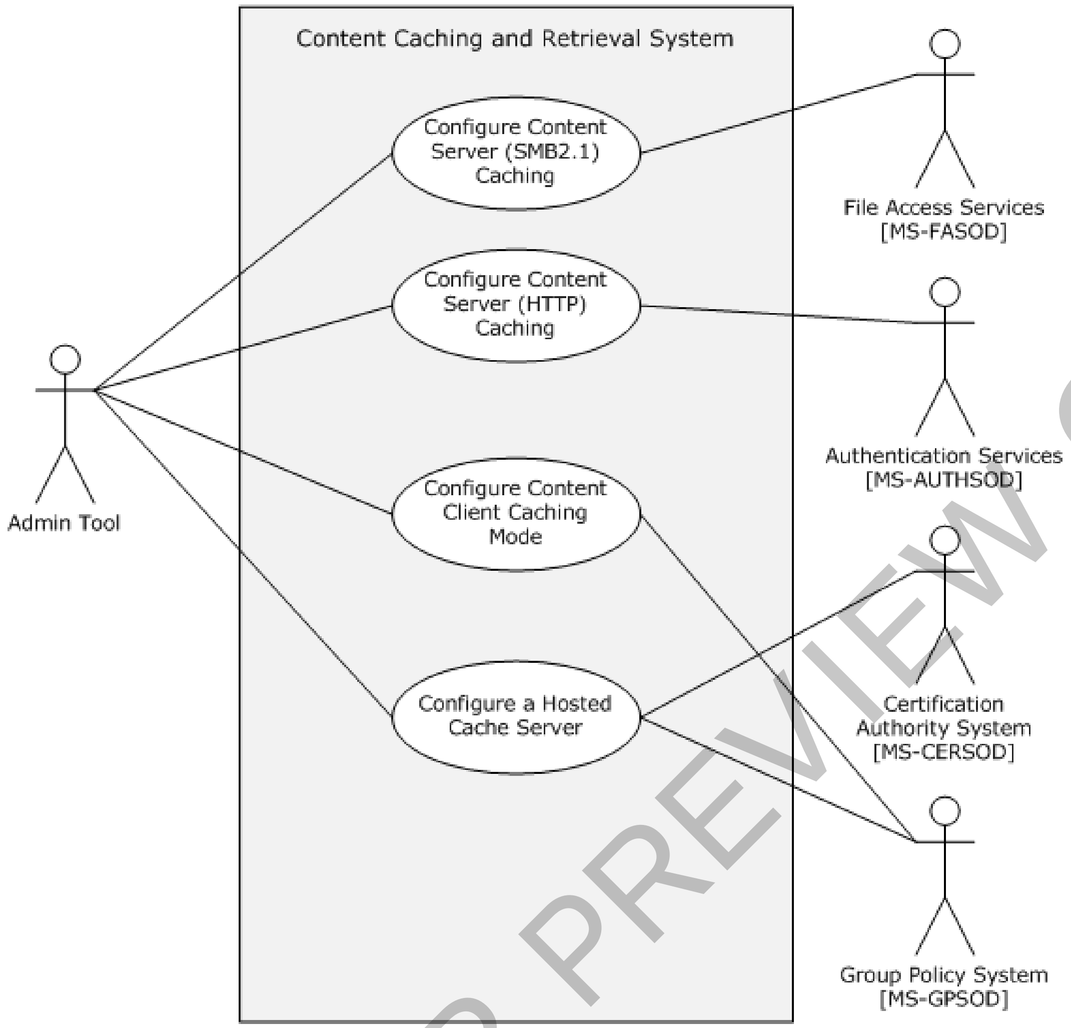

Content Caching and Retrieval System
Configure Content Server (SMB2.1) Caching
Configure Content Server (HTTP) Caching
Configure Content Client Caching Mode
Configure a Hosted Cache Server
Admin Tool
File Access Services [MS-FASOD]
Authentication Services [MS-AUTHSOD]
Certification Authority System [MS-CERSOD]
Group Policy System [MS-GPSOD]

Figure 10: Use cases included in the Configuring Content Caching and Retrieval Components summary use case

DRAFT: FOR PREVIEW ONLY

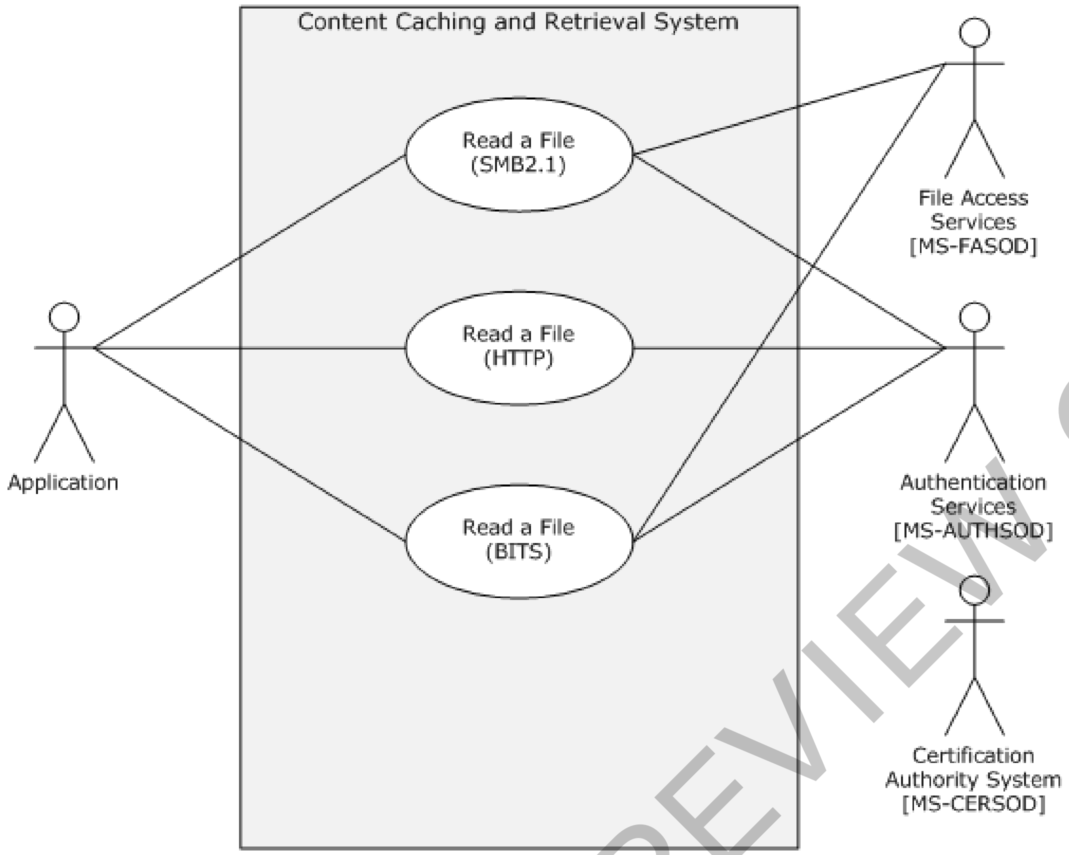

Content Caching and Retrieval System
Read a File (SMB2.1)
Read a File (HTTP)
Read a File (BITS)
Application
File Access Services [MS-FASOD]
Authentication Services [MS-AUTHSOD]
Certification Authority System [MS-CERSOD]

Figure 11: Use cases included in the Reading Content summary use case

2.4.4 Summary Use Case Descriptions

Note that the configuration tools on Microsoft Windows® are local only and therefore do not use protocols.

2.4.4.1 Configuring Content Caching and Retrieval Components

This section summarizes use cases for configuring caching on content servers, content clients, and hosted cache servers.

2.4.4.1.1 Configuring SMB 2.1/2 Content Server Caching

Goal: To enable Content Caching and Retrieval on a content server share directory.

Context of Use: The Administrator is configuring a content server and is either adding a shared directory or modifying a shared directory.

Direct Actor: The direct actor is the Administrator Tool. The Administrator Tool's interest is to correctly interpret, execute, and display the results of the commands issued by the Administrator.

Primary Actor: The primary actor is the Administrator. The Administrator's interest is expressed in administrative privileges and responsibility for using the File Access Services System and the Content Caching and Retrieval System to provide file services.

Copyright © 2011 Microsoft Corporation.

Release: Sunday, October 16, 2011

DRAFT: FOR PREVIEW ONLY

Supporting Actors: The supporting actors for this use case are as follows:

- Administrator Client: Maintains a consistent access mechanism to the SMB 2.1 File Service and Content Caching and Retrieval configuration.
- SMB File Service: Provides and maintains a secure and consistent file service by enforcing directory quotas and file screens.
- Object Store: Stores files and directories.

Preconditions: The Administrator has identified a content server and a shared directory on its Object Store, and wishes to enable Content Caching and Retrieval on that shared directory.

Minimal Guarantees: Content Caching and Retrieval is not enabled.

Success Guarantee: Content Caching and Retrieval is enabled on the shared directory identified on the content server.

Trigger: The Administrator Tool receives a request from the Administrator to configure Content Caching and Retrieval.

2.4.4.1.1.1 Main Success Scenario

1. The Administrator Tool (Share and Storage Management Console) creates or modifies an SMB 2.1/2 share with the specified SMB 2.1/222 share name and directory on the content server.
2. The Administrator Tool enables Content Caching and Retrieval on the SMB 2.1/2 share.

2.4.4.1.2 Configuring HTTP Content Server Caching

Goal: To enable Content Caching and Retrieval on a content server-hosted Web site.

Context of Use: The Administrator is configuring an HTTP server and is either adding a Web site or modifying an existing site.

Direct Actor: The direct actor is the Administrator Tool (netsh). The Administrator Tool's interest is to correctly interpret, execute, and display the results of the commands issued by the Administrator.

Primary Actor: The primary actor is the Administrator. The Administrator's interest is expressed in administrative privileges and responsibility for using the Web site.

Supporting Actors: The supporting actors for this use case are as follows:

- Admin Client: Maintains a consistent access mechanism to the HTTP server and Content Caching and Retrieval configuration.
- HTTP Server: Provides and maintains a secure and consistent file service.
- Object Store: Stores files and directories.

Preconditions: The Administrator has identified a content server that is hosting a Web site on its Object Store, and wishes to enable Content Caching and Retrieval on that site.

Minimal Guarantees: Content Caching and Retrieval is not enabled.

Success Guarantee: Content Caching and Retrieval is enabled on the Web site identified on the content server.

DRAFT: FOR PREVIEW ONLY

Trigger: The Administrator Tool receives a request from the Administrator to configure a Web site.

2.4.4.1.2.1 Main Success Scenario

- The Administrator Tool creates or modifies a Web site hosted on the content server.

2.4.4.1.3 Configuring Content Client Caching Mode

Goal: To configure content caching mode on a client peer.

Context of Use: The Administrator is configuring a content client to use either hosted cache or distributed cache, which are mutually exclusive roles and specified by the ADM element Server Role.

Direct Actor: The direct actor is the Administrator Tool (netsh). The Administrator Tool's interest is to correctly interpret, execute, and display the results of the commands issued by the Administrator.

Primary Actor: The primary actor is the Administrator. The Administrator's interest is expressed in administrative privileges and responsibility for the content client using the Content Caching and Retrieval System.

Supporting Actors: The supporting actor for this use case is as follows:

- Admin Client: Maintains a consistent access mechanism to the content client and the Content Caching and Retrieval configuration of the client.

Preconditions: The Administrator has identified a client peer (content client) and wishes to configure the caching mode. The client peer is Content Caching and Retrieval-capable.

Minimal Guarantees: No action is taken that affects systems on the client peer.

Success Guarantee: The caching mode is set to DISTRIBUTED or HOSTED CLIENT.

Trigger: The Administrator Tool receives a request from the Administrator to configure the caching mode.

2.4.4.1.3.1 Main Success Scenario

1. The Administrator Tool establishes a connection to the Content Caching and Retrieval System configuration.
2. The Administrator Tool changes context to Caching and Retrieval.
3. The Administrator Tool sets the Server Role to HOSTEDCLIENT or DISTRIBUTED.
4. The firewall allows DISTRIBUTED or HOSTEDCLIENT mode protocols.

2.4.4.1.4 Configuring a Hosted Cache Server

Goal: To configure content-caching mode on a server.

Context of Use: The Administrator is configuring a hosted cache server.

Direct Actor: The direct actor is the Administrator Tool. The Administrator Tool's interest is to correctly interpret, execute, and display the results of the commands issued by the Administrator.

DRAFT: FOR PREVIEW ONLY

Primary Actor: The primary actor is the Administrator. The Administrator's interest is expressed in administrative privileges and responsibility for the hosted cache server using the Content Caching and Retrieval System.

Supporting Actors: The supporting actor for this use case is as follows:

- Admin Client: Maintains a consistent access mechanism to the hosted cache server and the Content Caching and Retrieval configuration.

Preconditions: The Administrator has identified a server and wishes to configure it as a hosted cache server. The identified server is Content Caching and Retrieval-capable. The server has a fully qualified domain name (FQDN) and, if Client Authentication is to be used, is a member of a domain.

Minimal Guarantees: No action is taken that affects systems on the server.

Success Guarantee: The Server Role is set to HOSTEDSERVER, and the Client Authentication is set to DOMAIN or NONE.

Trigger: The Administrator Tool receives a request from the Administrator to configure the hosted cache server.

2.4.4.1.4.1 Main Success Scenario

1. The Administrator Tool establishes a connection with the Content Caching and Retrieval System configuration of the Server.
2. The Administrator Tool changes context to Caching and Retrieval.
3. The Administrator Tool sets the Server Role to HOSTEDSERVER, and the Client Authentication is set to DOMAIN or NONE.
4. The firewall allows HOSTEDSERVER protocols.

2.4.4.2 Initial Reading and Caching of a File from a Content Server

Goal: To read content from a content server with Caching and Retrieval in effect.

Context of Use: A number of users are attempting to read content that resides on a slow link. When some or all of the content is transferred to the LAN, Content Caching and Retrieval will operate to improve the performance of the File Access Service for later users.

Direct Actor: The direct actor is the Application. The Application's interest is to consume file-reading services by means of the content client and to correctly execute and display the results of commands issued by a user.

Primary Actor: The primary actor is the User. The User's interest is to access content on a content server by using the File Access Services System and the Content Caching and Retrieval System to provide file services.

Supporting Actors: The supporting actors for this use case are as follows:

- Content client: Maintains a consistent access mechanism to the SMB 2.1/2 File Service and Content Caching and Retrieval System.
- SMB 2.1/2 File server: Provides and maintains a secure and consistent file service by enforcing directory quotas and file screens. Provides metadata for content to requesting content clients using the SMB 2.1/2 protocol as a transport.

DRAFT: FOR PREVIEW ONLY

- HTTP Server: Provides and maintains a secure and consistent file streaming service. Provides metadata for content to requesting content clients using HTTP (PeerDist) as a transport.
- Object Store: Stores files and directories.

Minimal Guarantees: No action is taken that affects other directories or shares on the content server.

Success Guarantee: The Application obtains a **handle** to the requested file.

Trigger: The application receives a request from the user to read a file.

2.4.4.2.1 Main Success Scenario

1. The Application establishes a communication channel to the content server.
2. The content server authorizes the User.
3. The content server provides metadata in the format specified in [MS-PCCRC] to the Content Client.
4. The client searches attempts to locate content within the branch office.
5. The content server returns the data requested by the client (either Block SMB 2.1/2 or stream HTTP) to the Application.
6. The client caches the data so that it can be made available to peers.

2.4.4.3 Metadata Retrieval

2.4.4.3.1 Using SMB 2.1/2 Metadata Retrieval

Goal: Use SMB 2.1/2 for metadata retrieval.

Context of use: The User has located a file on a content server on a WAN link and wants to read that file.

Minimal Guarantees: No action is taken that affects other files exposed on the content server as a result of this operation. Files Access Services operate with no content caching.

Success Guarantee: The User obtains the metadata required to open the communication channel to retrieve the file.

Trigger: The application receives a request from the user to read a file.

2.4.4.3.1.1 Main Success Scenario

1. The User requests the Application to read a file.
2. The Application uses the content client to open a file handle to the required existing file on the content server using the mechanisms of SMB 2.1/2. This operation completes successfully.
3. The Application directs the content client to read data from the file represented by the file handle. As part of that operation, the content client requests content information (metadata--block hashes, segment hashes, and HoD and private segment keys (Kp) for the data) as specified in [MS-SMB2].

DRAFT. FOR PREVIEW ONLY

4. The content server sends content identifiers on the same channel to the content client (in this case, SMB version 2.1) as specified in [MS-SMB2].

5. The content client computes segment identifiers (HoHoDk) for the data as specified in [MS-PCCRC].

2.4.4.3.1.2 System Assumptions and Preconditions

The following preconditions must be satisfied for successful operation:

Preconditions: The client computers are configured to use Content Caching and Retrieval. A shared folder with the desired file has been created on the content server with content caching support enabled. The user has located the URL of the file on the content server.

Note that the specific URLs (that is, \\server\share\file and \\192.168.0.3\share\file) will be considered different for local caching but not for Content Caching and Retrieval.

System Availability: Appropriate content caching components must be installed and enabled on all of the computers involved.

Domain Configuration: In a domain configuration, clients and servers have access to directory services provided by the domain.

Note that domain configuration is not a requirement but can be used.

Authentication Services: Authentication services as described in [MS-AUTHSOD] are available to all clients and servers.

Network Configuration: In order for system components running on different computers to communicate, the network services and infrastructure must be functional and configured.

2.4.4.3.2 HTTP Metadata Retrieval

Goal: Obtain metadata from a content server Web site with content caching support enabled using HTTP.

Context of Use: The User has located a file on a content server (a Web server on a WAN link) and wants to read the file.

Minimal Guarantees: No action is taken that affects other files exposed on the content server as a result of this operation. HTTP access operates with no content caching.

Success Guarantee: The User obtains the metadata required to open the communication channel to retrieve the file.

Trigger: The application receives a request from the user to read a file.

2.4.4.3.2.1 Main Success Scenario

1. The User requests the Application to read a file.

2. The Application establishes an HTTP connection to the content server.

3. The content server authenticates the User through the mechanisms of [MS-AUTHSOD], if required.

DRAFT: FOR PREVIEW ONLY

4. The Application performs an HTTP GET request, as specified in [RFC2616], with PeerDist encoding as specified in [MS-PCCRTP].
5. The content server checks the authorization of the User to perform the action, if required [MS-AUTHSOD].
6. The content server retrieves metadata (block hashes, segment hashes, and private segment key) for the data as specified in [MS-PCCRC].
7. The content server sends metadata on the same application channel to the Content Client, in this case HTTP, as specified in [MS-PCCRTP].
8. The Content Client computes a segment discovery key as specified in [MS-PCCRC].

2.4.4.3.2.2 System Assumptions and Preconditions

The following preconditions must be satisfied for successful operation:

Preconditions: The content client and content server computers are configured to use content caching. The client is configured with the network address of the content server. A Web site with a file has been created on the content server with caching support enabled. The user has located the URL of the file on the content server.

System Availability: Content (BranchCache) caching must be installed and enabled on all of the computers involved.

Domain Configuration: In a domain configuration, client and servers have access to directory services provided by the domain.

Note that domain configuration is not a requirement.

Authentication Services: Authentication services as described in [MS-AUTHSOD] are available to all clients and servers.

Network Configuration: In order for system components running on different computers to communicate, the network services and infrastructure must be functional and configured.

2.4.4.3.3 BITS--HTTP Metadata Retrieval

Goal: Retrieve metadata from a content server Web site with content caching support enabled using BITS.

Context of Use: The user has located a file on a content server (a Web server on a WAN link) and wants to read that file. The file is located on a Web site with caching support enabled. The file is not initially on the LAN and needs to be retrieved across a WAN link.

Minimal Guarantees: No action is taken that affects other files exposed on the content server as a result of this operation. HTTP access operates with no content caching.

Success Guarantee: The User opens the channel to access the file.

Trigger: An Application performs a read operation on content located on a WAN link.

2.4.4.3.3.1 Main Success Scenario

1. The User requests the Application to read a file.

DRAFT: FOR PREVIEW ONLY

2. The Application establishes an HTTP connection to the content server.

3. The content server authenticates the User through the mechanisms of [MS-AUTHSOD], if required.

4. The Application performs an HTTP GET request, as specified in [RFC2616], decorated with [MS-PCCRTP] extensions.

5. The content server checks the authorization of the User to perform the action [MS-AUTHSOD], if required.

6. The content server retrieves metadata (block hashes, segment hashes, and private segment key) for the data as specified in [MS-PCCRC].

7. The content server sends metadata on the same application channel to the content client as specified in [MS-PCCRTP].

8. The content client computes a segment discovery key [MS-PCCRC].

2.4.4.3.3.2 System Assumptions and Preconditions

The following preconditions must be satisfied for BITS metadata retrieval to operate successfully:

Preconditions: The content client and content server computers are configured to use content caching. The client is configured with the network address of the content server. A Web site with a file has been created on the content server with caching support enabled.

System Availability: Content (BranchCache) caching must be installed and enabled on all of the computers involved.

Domain Configuration: In a domain configuration, clients and servers have access to directory services provided by the domain.

Note that domain configuration is not a requirement.

Authentication Services: Authentication services as described in [MS-AUTHSOD] are available to all clients and servers.

Network Configuration: In order for system components running on different computers to communicate, the network services and infrastructure must be functional and correctly configured.

2.4.4.4 Content Discovery and Retrieval

2.4.4.4.1 Content Discovery and Retrieval with Hosted Cache (Cached Data Unavailable)

Goal: Read a file from a content server with content caching support enabled when the caching mode is hosted.

Context of Use: The User has located a file on a content server on a WAN link and wants to read that file. The file is located on a shared folder with content caching support enabled. The file is not initially on the LAN and needs to be retrieved from the remote end of a WAN.

Minimal Guarantees: No action is taken that affects other files exposed on the content server as a result of this operation. Files Access Services operate with no content caching.

Success Guarantee: The User reads the file.

DRAFT: FOR PREVIEW ONLY

Trigger: An Application performs a read operation on content located on a WAN link.

2.4.4.4.1.1 Actors

The Actors and their associated interests are as follows:

User: The User's interest is to use the Application to access files on a content server.

Application: The Application's interest is to consume file reading services by means of the content client.

Content Client: The content client's interest is to use client-side protocol components and consume the File Services that are offered by the content server and to offer received content to the hosted cache server.

Content Server: The content server's interest in this use case is to provide and maintain a secure and consistent File Service [MS-FASOD] and provide content metadata.

Hosted Cache Server: The hosted cache server's interest is to provide two mechanisms: one for querying for the availability of certain content and the other for retrieving content from a content client.

2.4.4.4.1.2 Main Success Scenario

1. The User requests the Application to read a file.
2. The Application obtains content identifiers using one of the access mechanisms described in section 2.4.4.3.
3. The content client computes segment identifiers (HoHoDk) for the data.
4. The Content Client queries the hosted cache server for the availability of blocks from the target segments using Peer Content Caching and Retrieval: Retrieval Protocol as specified in [MS-PCCRR].
5. The hosted cache server indicates that it does not have the required blocks as specified in [MS-PCCRR].
6. The Content Client retrieves the content from the content server using the mechanism specified in [MS-FASOD].
7. The retrieved data is placed in the local cache of the content client computer.
8. The content client retrieves the data from the local cache and returns it to the Application.
9. The Application delivers the file contents to the User.
10. The content client offers the metadata content to the Hosted Cache server as specified in [MS-PCHC].
11. The hosted cache server optionally authenticates the content client.
12. The retrieved data is placed in the distributed cache of the content client computer.
13. The hosted cache server requests from the content client any desired segments and blocks [MS-PCCRR].

DRAFT: FOR PREVIEW ONLY

14. The content client sends the requested segments and blocks to the hosted cache server [MS-PCCRR].

2.4.4.4.1.3 System Assumptions and Preconditions

The following preconditions must be satisfied for hosted cache mode to operate successfully:

Preconditions: The client computers are configured to use Content Caching and Retrieval. A shared folder with the desired file has been created on the content server with content caching support enabled. The user has located the URL of the file on the content server. The client computers are configured with the location of the hosted cache server.

Note that the specific URL (that is, \\server\share\file and \\192.168.0.3\share\file) will be considered different for local caching but not for Content Caching and Retrieval.

System Availability: Appropriate content caching components must be installed and enabled on all of the computers involved.

Domain Configuration: In a domain configuration, clients and servers have access to directory services provided by the domain.

Note that domain configuration is not a requirement but can be used.

Authentication Services: Authentication services as described in [MS-AUTHSOD] are available to all clients and servers.

Network Configuration: In order for system components running on different computers to communicate, the network services and infrastructure must be functional and correctly configured.

2.4.4.4.2 Content Discovery and Retrieval with Hosted Cache (Cached Data Available)

Goal: Read a file from a content server with content caching support enabled when the caching mode is hosted.

Context of Use: The User has located a file on a content server on a WAN link and wants to read that file. The file is located on a shared folder with content caching support enabled. The file has previously been retrieved across the WAN link and is cached on the LAN.

Minimal Guarantees: No action is taken that affects other files exposed on the content server as a result of this operation. Files Access Services operate with no content caching.

Success Guarantee: The User reads the file.

Trigger: An Application performs a read operation on content located on a WAN link.

2.4.4.4.2.1 Actors

The Actors and their associated interests are as follows:

User: The User's interest is to use the Application to access files on a content server.

Application: The Application's interest is to consume file reading services by means of the content client.

DRAFT: FOR PREVIEW ONLY

Content Client: The content client's interest is to use client-side protocol components and consume the file services that are offered by the content server and to offer received content to the Hosted Cache Server.

Content Server: The content server's interest in this use case is to provide and maintain a secure and consistent File Service [MS-FASOD] and provide content metadata.

Hosted Cache Server: The Hosted Cache Server's interest is in providing two mechanisms: one for querying for the availability of certain content and the other for retrieving content from a content client.

2.4.4.4.2.2 Main Success Scenario

1. The User requests the Application to read a file.
2. The Application obtains content identifiers using one of the access mechanisms described in section 2.4.4.3.
3. The content client computes segment identifiers (HoHoDk) for the data as specified in [MS-PCCRC] section 2.2.
4. The content client queries the hosted cache server for the availability of blocks from the target segments using the Peer Content Caching and Retrieval Content Identification Protocol as specified in [MS-PCCRC].
5. The hosted cache server returns the requested data using the Peer Content Caching and Retrieval: Retrieval Protocol, as specified in [MS-PCCRR].
6. The retrieved data is placed in the distributed cache of the content client computer.
7. The retrieved data is placed in the local cache of the content client computer.
8. The content client retrieves the data from the local cache and supplies that data to the Application.
9. The Application delivers the file contents to the User.

2.4.4.4.2.3 System Assumptions and Preconditions

The following preconditions must be satisfied for hosted cache mode to operate successfully:

Preconditions: The client computers are configured to use Content Caching and Retrieval. A shared folder with the desired file has been created on the content server with content caching support enabled. The user has located the URL of the file on the content server. A prior client within the local network has retrieved the data, enabling some or all of the content to be stored in a distributed cache peer.

Note that the specific URLs (that is, \\server\share\file and \\192.168.0.3\share\file) will be considered different for the local file cache but not for Content Caching and Retrieval.

System Availability: Appropriate content caching components must be installed and enabled on all of the computers involved.

Domain Configuration: In a domain configuration, clients and servers have access to directory services provided by the domain.

Note that domain configuration is not a requirement but can be used.

DRAFT: FOR PREVIEW ONLY

Authentication Services: Appropriate Authentication services as described in [MS-AUTHSOD] are available to all clients and servers. Domain membership is not a requirement for clients and servers.

Network Configuration: In order for system components running on different computers to communicate, the network services and infrastructure must be functional and correctly configured.

2.4.4.4.3 Content Discovery and Retrieval with Distributed Cache (Cached Data Unavailable)

Goal: To read a file from a content server with content caching support enabled, when the caching mode is distributed.

Context of Use: The User has located a file on a content server on a WAN link and wants to read that file. The file is located on a shared folder with content caching support enabled. The file is not initially on the LAN and needs to be retrieved from the remote end of a WAN.

Minimal Guarantees: No action is taken that affects other files exposed on the content server as a result of this operation.

Success Guarantee: The User reads the file.

Trigger: User action in the Application.

2.4.4.4.3.1 Actors

The Actors and their associated interests are as follows:

User: The User's interest is to use the Application to access files on a content server.

Application: The Application's interest is to consume file reading services by means of the content client.

Content Client: The content client's interest is to use client-side protocol components and consume the file services that are offered by the content server.

Content Server: The content server's interest in this use case is to provide and maintain a secure and consistent File Service as described in [MS-FASOD] and to provide content metadata.

Distributed Cache Peers: A distributed cache peer's interest is to provide two mechanisms: one responding to broadcasts for cached content and the other for delivering the cached content.

2.4.4.4.3.2 Main Success Scenario

1. The User requests the Application to read a file.
2. The Application obtains content identifiers using one of the mechanisms in section 2.4.4.3.
3. The client broadcasts a probe message as specified in [MS-PCCRD] to discover if any distributed cache peers have the required content.
4. No distributed cache peer responds with an indication of content.

   Note that successful discovery would be via the Peer Content Caching and Retrieval Discovery Protocol, as specified in [MS-PCCRD].

5. The content client retrieves the data from the content server.

DRAFT: FOR PREVIEW ONLY

6. The retrieved data is placed in the local cache of the content client computer.
7. The content client retrieves the data from the local cache and supplies that data to the Application.
8. The Application delivers the file contents to the User.
9. The retrieved data is placed in the distributed cache of the content client computer.

2.4.4.4.3.3 System Assumptions and Preconditions

The following preconditions must be satisfied for distributed cache mode to operate successfully:

Preconditions: The client computers are configured to use Content Caching and Retrieval. A shared folder with the desired file has been created on the content server with content caching support enabled. The user has located the URL of the file on the content server.

Note that the specific URL (that is, \\server\share\file and \\192.168.0.3\share\file) will be considered different for local caching but not for Content Caching and Retrieval.

System Availability: Appropriate content caching components must be installed and enabled on all of the computers involved.

Domain Configuration: In a domain configuration, clients and servers have access to directory services provided by the domain.

Note that domain configuration is not a requirement but can be used.

Authentication Services: Authentication services as described in [MS-AUTHSOD] are available to all clients and servers.

Network Configuration: In order for system components running on different computers to communicate, the network services and infrastructure must be functional and correctly configured.

2.4.4.4.4 Content Discovery and Retrieval with Distributed Cache (Cached Data Available)

Goal: Read a file from a content server with content caching support enabled, when the caching mode is distributed.

Context of Use: The User has located a file on a content server on a WAN link and wants to read the file. The file is located on a shared folder with content caching support enabled. The file has previously been retrieved across the WAN link and is cached on the LAN.

Minimal Guarantees: No action is taken that affects other files exposed on the content server as a result of this operation.

Success Guarantee: The User reads the file.

Trigger: User action in the Application.

2.4.4.4.4.1 Actors

The Actors and their associated interests are as follows:

User: The User's interest is to use the Application to read files on a content server.

DRAFT: FOR PREVIEW ONLY

Application: The Application's interest is to consume file reading services by means of the content client.

Content Client: The content client's interest is to use client-side protocol components and consume the file services that are offered by the content server.

Content Server: The content server's interest is to provide and maintain a secure and consistent File Service as described in [MS-FASOD] and provide content metadata using the SMB 2.1/2 protocol as transport.

Distributed Cache Peers: A distributed cache peer's interest is to respond to broadcasts for cached content and to deliver the cached content.

2.4.4.4.4.2 Main Success Scenario

1. The User requests the Application to read a file.
2. The Application obtains content identifiers using one of the mechanisms shown in section 2.4.4.3.
3. The content client broadcasts a Probe message as specified in [MS-PCCRD].
4. A distributed cache peer with the required content responds with a Probe Match message as specified in [MS-PCCRD].
5. The content client requests the content from the distributed cache peer as specified in [MS-PCCRR].
6. The content client receives the data and decrypts the data as specified in [MS-PCCRR].
7. The content client validates the block data against the block hash as specified in [MS-PCCRC].
8. The retrieved data is placed in the distributed cache of the content client computer.
9. The retrieved data is placed in the local cache of the content client computer.
10. The content client retrieves the data from the local cache and supplies that data to the Application.
11. The Application delivers the file contents to the User.

2.4.4.4.4.3 System Assumptions and Preconditions

The following preconditions must be satisfied for distributed cache mode to operate successfully:

Preconditions: The client computers are configured to use Content Caching and Retrieval. A shared folder with the desired file has been created on the content server with content caching support enabled. The user has located the URL of the file on the content server. Note that the specific URLs (that is, \\server\share\file and \\192.168.0.3\share\file) will be considered different for local caching but not for Content Caching and Retrieval.

A prior client within the LAN has retrieved the data, enabling some or all of the content to be stored in a distributed cache peer.

System Availability: Appropriate content caching components must be installed and enabled on all of the computers involved.

Domain Configuration: In a domain configuration, clients and servers have access to directory services provided by the domain.

DRAFT - FOR PREVIEW ONLY

Note that domain configuration is not a requirement but can be used.

Authentication Services: Authentication services as described in [MS-AUTHSOD] are available to all clients and servers.

Network Configuration: In order for system components running on different computers to communicate, the network services and infrastructure must be functional and correctly configured.

2.5 Versioning, Capability Negotiation, and Extensibility

The Content Caching and Retrieval System does not define any versioning or capability negotiation beyond what is described in the member protocol specifications, as listed in section 2.2. The Background Intelligent Transfer Service (BITS) Upload Protocol, as specified in [MC-BUP], the Peer Content Caching and Retrieval: Hypertext Transfer Protocol (HTTP) Extensions, as specified in [MS-PCCRTP], and the Peer Content Caching and Retrieval: Retrieval Protocol, as specified in [MS-PCCRR] have version-specific capability.<5>

2.6 Error Handling

This section describes the common failure scenarios and specifies the system behavior in such conditions.

2.6.1 Connection Disconnected

A common failure scenario is an unexpected connection breakdown between the system and external entities. A disconnection can be caused by the network not being available, or by one of the communicating participants becoming unavailable. In the case where the network is not available, both participants remain active and expect the other party to continue the communication pattern specified by the protocol being executed at the time of the failure. Similarly, in the case where one of the participants is not available, the active participant expects the communication to proceed as specified by the protocol being executed.

Generally, a protocol detects a connection breakdown failure through either of the following methods:

- By using a timer object that generates an event if the corresponding participant has not responded within a reasonable time span.
- By being notified by the underlying protocol that the connection is disconnected.

When a connection disconnected event is detected, it causes the protocol to tear down all related communications and update any necessary data structures to maintain the system state.

Details about how each protocol detects a connection disconnected event, and how it behaves under this scenario, are provided in the specifications of the member protocols, as listed in section 2.1.5.

A content client that fails to find content on a hosted cache server is not considered a common failure, but rather a normal operation. The first time any content client attempts to retrieve data from a hosted cache server, it will fail to find the content, and the client will simply request the full content from the content server.

2.6.2 Internal Failures

The Content Caching and Retrieval System is dependent on the File Access Service System, which is not defensive against internal failures of its state other than that described in [MS-FASOD] and the specifications of its member protocols.

DRAFT: FOR PREVIEW ONLY

2.6.3 System Configuration Corruption or Unavailability

The Content Caching and Retrieval System relies on the availability and consistency of its configuration data. Configuration consists of the data that determines the behavior of the system under specific conditions or for specific functionality. In the event that the Content Caching and Retrieval System fails in some manner and the File Access Service System is still functioning, the system will operate without content caching.

During content retrieval malformed messages received by the content client and messages of unknown type are quietly discarded; see [MS-PCCRR] section 3.1.1.5.4.

2.7 Coherency Requirements

2.7.1 Timers

This section explains the timers that are significant to the state of the entire system. The system is dealt with in terms of a Client Framework that describes the peer-role and server-role peers. It also includes a summary of the timers for each member protocol.

2.7.1.1 Member Protocol Timer Summary

Member protocol	Timer summary
Peer Content Caching and Retrieval: Hypertext Transfer Protocol (HTTP) Extensions [MS-PCCRTP]	None.
Peer Content Caching and Retrieval Discovery Protocol [MS-PCCRD]	Two timers are associated with the Discovery Protocol operations: back off and request. See [MS-PCCRD] section 3.1.2.
Peer Content Caching and Retrieval: Hosted Cache Protocol [MS-PCHC]	None.
Peer Content Caching and Retrieval: Retrieval Protocol Specification [MS-PCCRR]	Associated with the request timer and upload timer. See [MS-PCCRR] section 3.1.2.

2.7.1.2 Client Framework

2.7.1.2.1 Hosted Cache Mode

The client sets the **Upload Timer** as specified in [MS-PCCRR] section 3.1.2.

When the **Upload Timer** expires, the client aborts the segment retrieval session.

2.7.1.2.2 Distributed Cache Mode

The following timers are associated with the client framework operations:

- **Upload Timer**:

  Set by the higher-layer applications in the client at the beginning of each segment retrieval session. The segment retrieval session aborts when the Download Timer expires.<6>

- **Server List Timer**:

DRAFT: FOR PREVIEW ONLY

A separate instance of this timer is started for each empty **Server Information List** that is created and for each previously populated **Server Information List** that becomes empty.

It is disabled if the **Server Information List** becomes non-empty.

An empty **Server Information List** is removed when its **Server List Timer** expires. The default timeout value is set to 15 seconds.<7>

When the **Server List Timer** for a **Server Information List** of a segment expires, the list is deleted.

- **Server Entry Timer**:

  The timer for each server entry in a **Server Information List**.

  It is started after an entry is added into the **Server Information List**. An entry is removed from the list after its timer expires. The default timeout value is set to 15 seconds.<8>

  When the **Server Entry Timer** for an entry in a segment ID's **Server Information List** expires, the entry is deleted from the cache.

2.7.2 Non-Timer Events

This section explains the non-timer events that are significant to the state of the entire system. The system is dealt with in terms of a Client Framework that describes the peer-role and server-role peers. It also includes a summary of the non-timers events for each member protocol.

2.7.2.1 Member Protocol Non-Timer Events Summary

Member protocol	Non-timer event summary
Peer Content Caching and Retrieval: Retrieval Protocol Specification [MS-PCCRTP]	None.
Peer Content Caching and Retrieval Discovery Protocol [MS-PCCRD]	Receive Probe, Receive Probe-Match; see [MS-PCCRD] section 3.1.5.
Peer Content Caching and Retrieval: Hosted Cache Protocol [MS-PCHC]	None.
Peer Content Caching and Retrieval: Retrieval Protocol [MS-PCCRR]	GetBlockList Initiation (see [MS-PCCRR] section 3.1.1.4.2). GetBlocks Initiation (see [MS-PCCRR] section 3.1.1.4.3).

2.7.2.2 Client Framework - Hosted Cache Mode, Higher-Layer Triggered Events

2.7.2.2.1 Content Retrieval Request

When the client instance of the framework receives a Content Retrieval Request from a higher-layer application, it adds the list of segments and the corresponding block ranges for each segment to its content cache. The framework then initiates a segment retrieval session for each segment in the requested content.

DRAFT: FOR PREVIEW ONLY

2.7.2.2.2 Segment Retrieval Session Initiation

A client in Hosted Cache mode (either a peer downloading content from a hosted cache, or vice versa) performs the following actions when a segment retrieval session is initiated:

- The client starts the **Download Timer** for that segment retrieval session.
- Add the server into the **Server Information List** of the segment ID, and set the corresponding server status as "free".
- If the requested block ranges in the segment consist of three or fewer consecutive blocks, the client:
  - Starts a Download Schedule Session if the **Download Initiated Flag** has not been set.
- Otherwise, initiates a Retrieval Protocol GetBlockList request (MSG_GETBLKLIST) to the server.

2.7.2.3 Client Framework - Distributed Cache Mode, Higher-Layer Triggered Events

2.7.2.3.1 Content Retrieval Request

When the client instance of the framework receives a content retrieval request from a higher-layer application, it receives a list of segments and block ranges for each segment. The framework concurrently initiates a segment retrieval session for each segment in the requested content.

2.7.2.3.2 Segment Retrieval Session Initiation

The client performs the following actions in the order specified when a segment retrieval session is initiated for a **segment ID**:

1. Starts a **Download Timer** for that segment retrieval session.
2. Checks the status of the **Server Information List** of the segment ID:
   1. If the **Server Information List** for the segment ID exists and is empty, then the client aborts the segment retrieval session.
   2. If the **Server Information List** for the segment ID exists and is not empty, then the client:
      1. Clears the available block ranges for every server in the list and sets each server status as "free".
      2. If the requested block ranges of the segment consist of three or fewer consecutive blocks, the client sets the **Download Initiated Flag** and starts a **download schedule session**.
      3. Otherwise, the client initiates a Retrieval Protocol GetBlockList request to each server in the list.
   3. Otherwise (the **Server Information List** does not exist), the client:
      1. Creates a **Server Information List** for this segment ID.
      2. Starts the **Server List Timer** for the list.
      3. If the maximum number of **Server Information Lists** has been reached, the client deletes the least recently used **Server Information List**.

DRAFT: FOR PREVIEW ONLY

4. If the **Discovery Frequency Counter** is greater than or equal to the threshold, the client aborts the segment retrieval session. Otherwise, the client starts a Discovery Protocol instance (as specified in [MS-PCCRD]) and passes it the segment ID.

2.7.2.4 Client Framework - Hosted Cache Mode, Other Local Events

2.7.2.4.1 Download Schedule Session

When a Download Schedule Session is started, then:

- If all requested blocks are marked as "downloaded", then the client returns all blocks in the requested block ranges and exits the segment retrieval session.
- If none of the requested blocks in the segment is marked as "idle", the client clears the **Download Initiated Flag** and exits the Download Schedule Session.
- If the server status is marked as "free" in the **Server Information List** of the segment ID:
  - If the available block ranges of the server are empty:
    - If the requested block ranges in the segment consist of three or less consecutive blocks, then the client:
      - Locates the first block that is marked as "idle" and changes it to "downloading".
      - Changes the server status to "busy" and initiates a Retrieval Protocol GetBlocks request to the server for that block.
      - Clears the **Download Initiated Flag** and exits the Download Schedule Session.
    - If the server has a set of available block ranges of the segment:
      - The client locates the first block in the host cache's available block ranges that is marked as "idle", changes the **Block Download Status** of that block to "downloading", changes the server status to "busy", and initiates a Retrieval Protocol GetBlocks request to the server for that block. Then the client clears the **Download Initiated Flag** and exits the Download Schedule Session.
- (No free server) If the server is marked as "complete", the client aborts the segment retrieval session and notifies the framework of missing blocks.
- Otherwise (server is busy), the client clears the **Download Initiated Flag** and exits the Download Schedule Session.

2.7.2.4.2 Retrieval Protocol GetBlockList Succeeds

When a Retrieval Protocol GetBlockList exchange returns valid block ranges of the requested segment, the client:

- Records the block ranges into the corresponding server entry in the **Server Information List** of the segment ID, and sets the server status as "free".
- If the **Download Initiated Flag** is not set, sets the flag and starts the Download Schedule Session.

DRAFT: FOR PREVIEW ONLY

2.7.2.4.3 Retrieval Protocol GetBlocks Succeeds

When a Retrieval Protocol GetBlocks exchange returns a valid block of the requested segment block ranges, the client:

- Stores the block in the content cache and marks the block status as "downloaded".
- If all blocks in the available block ranges of the server are all completed, marks the server status as "complete". Otherwise, marks the server status as "free" in the **Server Information List**.
- If the **Download Initiated Flag** is not set, sets the flag and starts the Download Schedule Session.

2.7.2.4.4 Retrieval Protocol Failure (GetBlockList or GetBlocks)

When a Retrieval Protocol GetBlockList request (see [MS-PCCRR] section 2.2.4.2) fails, the client:

- Removes the server from the **Server Information List** of the segment ID.
- If the **Download Initiated Flag** is not set, sets the flag and starts a Download Schedule Session.

When a Retrieval Protocol GetBlocks request (see [MS-PCCRR] section 2.2.4.3) fails, the client:

- Sets the status of the requested block to idle.
- Removes the server from the **Server Information List** of the segment ID.
- If the **Download Initiated Flag** is not set, sets the flag and starts a Download Schedule Session.

2.7.2.5 Client Framework - Distributed Cache Mode, Other Local Events

2.7.2.5.1 Server Peer Discovered by the Discovery Protocol

When a discovered peer is passed to the client by the Discovery Protocol, the client performs the following actions:

- If the **Server Information List** of the segment ID contains the maximum number of server entries, the client deletes the least recently used server.<9>
- Adds the newly discovered server into the **Server Information List** of the segment ID and sets the corresponding server status as free.
- If the requested block ranges in the segment consist of three or fewer consecutive blocks,<10> the client starts a Download Schedule Session with the newly discovered server if the **Download Initiated Flag** has not been set.
- If the requested ranges consist of disjoint blocks or more than three consecutive blocks, the client initiates a Retrieval Protocol GetBlockList request (see [MS-PCCRR] section 2.2.4.2) to the newly discovered server.

2.7.2.5.2 Discovery Protocol Failure - No Server Found

If the Discovery Protocol instance returns without a server being found, the client aborts the segment retrieval session.

DRAFT . FOR PREVIEW ONLY

2.7.2.5.3 Download Schedule Session

When a Download Schedule Session is started, then:

- If all requested blocks are marked as "downloaded", then the client returns all blocks in the requested block ranges and exits the segment retrieval session.
- If none of the requested blocks in the segment is marked as "idle", the client clears the **Download Initiated Flag** and exits the Download Schedule Session.
- For every server entry marked as "free" in the **Server Information List** for the Segment Id:
  - If the list of available block ranges for the server in the **Server Information List** is empty, then:
    - If the requested block ranges in the segment consist of three or less consecutive blocks, then the client:
      - Locates the first block that is marked as "idle" and changes it to "downloading".
      - Changes the server status to "busy" and initiates a Retrieval Protocol GetBlocks request (see [MS-PCCRR] section 2.2.4.3) to the server for that block.
    - Otherwise, the client skips to the next free server.
  - If there is a list of available block ranges for the server in the **Server Information List** for the segment, then:
    - The client locates the first block in the server's available block ranges that is marked as "idle", changes the **Block Download Status** of that block to "downloading", changes the server status to "busy", and initiates a Retrieval Protocol GetBlocks request to the server for that block.
    - Skips to the next free server.
- (No free server) If all the servers are marked as "complete", the client aborts the segment retrieval session and notifies the framework of missing blocks.
- Otherwise (still some busy servers), the client clears the **Download Initiated Flag** and exits the Download Schedule Session.

2.7.2.5.4 Retrieval Protocol GetBlockList Succeeds

When a Retrieval Protocol GetBlockList exchange (see [MS-PCCRR] section 2.2.4.2) returns valid block ranges of the requested segment, the client:

- Records the block ranges into the corresponding server entry in the **Server Information List** of the segment ID, and sets the server status as "free".
- If the **Download Initiated Flag** is not set, sets the flag and starts the Download Schedule Session.

2.7.2.5.5 Retrieval Protocol GetBlocks Succeeds

When a Retrieval Protocol GetBlocks exchange (see [MS-PCCRR] section 2.2.4.3) returns a valid block of the requested segment block ranges, the client:

DRAFT. FOR PREVIEW ONLY

- Stores the block in the content cache and marks the block status as "downloaded".
- If all blocks in the available block ranges of the server are all completed, marks the server status as "complete". Otherwise, marks the server status as "free" in the **Server Information List**.
- If the **Download Initiated Flag** is not set, sets the flag and starts the Download Schedule Session.

2.7.2.5.6 Retrieval Protocol Failure (GetBlockList or GetBlocks)

The cause of Retrieval Protocol failure could be that the exchange is aborted (see [MS-PCCRR] section 3.1.1.5), or that the **Request Timer** for the Retrieval Protocol expires.<11> This section describes the client action when each type of request--GetBlockList (MSG_GETBLKLIST) ([MS-PCCRR] section 2.2.4.2) or GetBlocks (MSG_GETBLKS) ([MS-PCCRR] section 2.2.4.3) fails.

When a Retrieval Protocol GetBlockList request fails, the client:

- Removes the server from the **Server Information List** of the segment ID if the number of failures exceeds the maximum number allowed.<12>
- If the **Download Initiated Flag** is not set, sets the flag and starts a Download Schedule Session.

When a Retrieval Protocol GetBlocks request fails, the client:

- Sets the status of the requested block to "idle".
- Removes the server from the **Server Information List** of the segment ID.
- If the **Download Initiated Flag** is not set, sets the flag and starts a Download Schedule Session.

2.7.3 Initialization and Reinitialization Procedures

The HTTP Client is initialized as per [MS-PCCRTP] section 3.1.4.

Probe and Probe-Match messages are initialized as per [MS-PCCRD] section 3.1.3.

The hosted cache is initialized as per [MS-PCHC] section 3.1.3.

The peer role server side is initialized as per [MS-PCCRR] section 2.1.1.

2.7.3.1 Client Framework

2.7.3.1.1 Hosted Cache Mode

A hosted cache client is configured with the server's address.

2.7.3.1.2 Distributed Cache Mode

No specific initialization required.

DRAFT: FOR PREVIEW ONLY

2.8 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.

The following figure illustrates how the content is encrypted by the system.

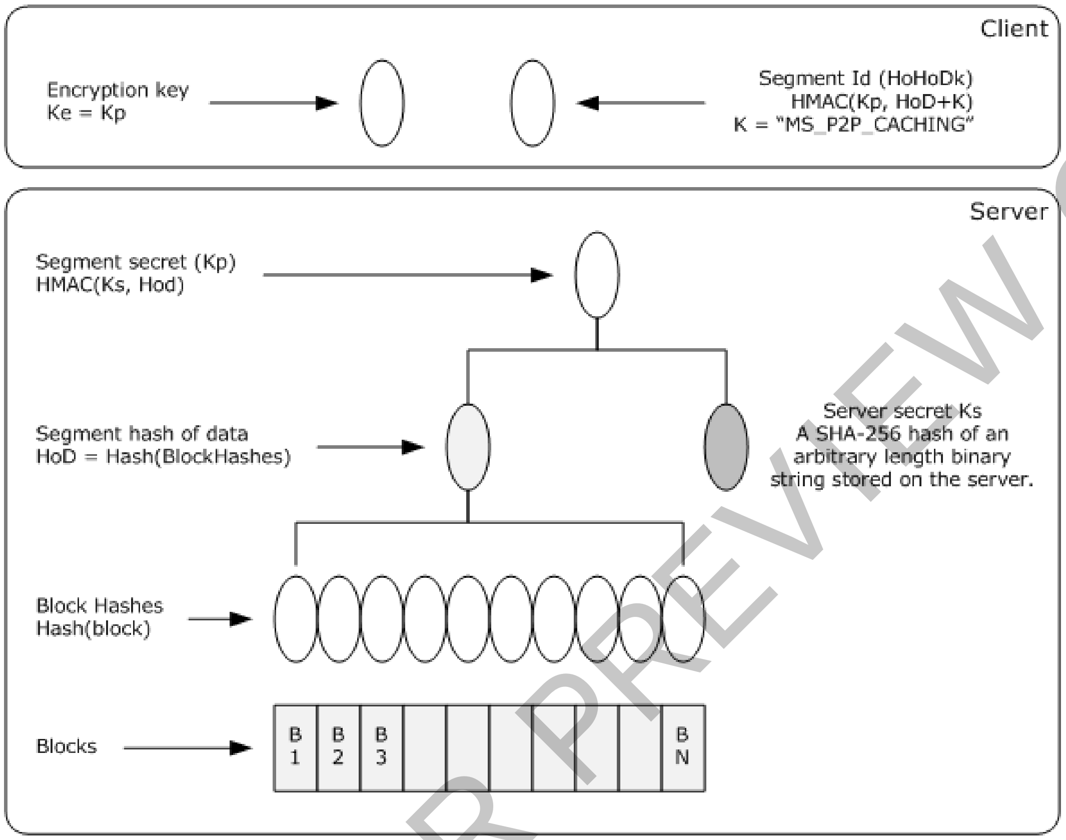


Figure 12: Content security

The following typical sequence of events describes the security of data during transportation and at rest (in the client cache or Hosted Cache).

A content client requests data from a content server, and by requesting metadata (note that there are different mechanisms for SMB 2.1/2 and HTTP) indicates that it is capable of understanding content caching.

1. The content server authenticates and authorizes the client in exactly the same way it would if caching were not being used. That is, authentication and authorization of a client to access data are independent of content caching.

2. The content server recognizes that the content client can utilize content caching, and checks to make sure that the stored metadata is up-to-date with the content.

3. The content server then sends the metadata on the same channel that data normally would have been sent. If an SSL connection has been established between the client and the server, then the hashes are sent back over this encrypted SSL connection.

DRAFT. FOR PREVIEW ONLY

4. The content client that is requesting content obtains the metadata and uses it to discover local availability.

5. The content client establishes a connection with the caching computer (a Hosted Cache server when Hosted Cache mode is used, or a peer caching computer when distributed cache mode is used), and requests the blocks that it requires.

6. The caching computer encrypts the blocks with an encryption key that is derived from the content metadata (using AES 128 by default) and sends it to the content client. The details of the encryption process for AES 128 can be found in [FIPS197].

7. The content client decrypts the data by using the same encryption key as the caching computer. The content client and the caching computer compute the same encryption key because they derive it from the same content metadata, which is sent by the content server.

8. After the content client decrypts the data, it validates the data. To do this, the content client computes the block hashes on the blocks received, and then compares them to the block hashes received in the content metadata from the server. If the hashes do not match, the content client discards the data.

The data in the cache is accessible. The data is stored in the clear in the distributed cache and the Hosted Cache, which is similar to other caches and data on the system.

2.8.1 Client-Side Content Security

In the figure that illustrates content security in the preceding section, Ke represents an encryption key derived from the segment secret (Kp). A sending peer encrypts data with Ke, but Ke is never disclosed between peers. The receiving client must already have obtained enough information to compute the value of Ke from a content server in order to decrypt the peer-supplied data. Ke = Kp.

2.8.2 Server-Side Content Security

The hash algorithm used is the SHA-256 hash as specified in [FIPS180-2]. Further details can be found in [MS-PCCRC] section 2.2. The **HMAC** function is defined in [RFC2104].

2.8.3 Use of Cryptography

Cryptographic algorithms used within the Content Caching and Retrieval System are AES-128 and SHA256 ([FIPS197] and [FIPS180-2]).

2.9 Additional Considerations

The Content Caching and Retrieval System supports the SMB 2.1/2 and HTTP 1.1 protocols. Applications do not need to directly communicate with the system although they can do so if they need to.<13> However, applications accessing SMB 2.1/2 and HTTP interfaces in both the Windows® 7 operating system and the Windows Server® 2008 R2 operating system transparently benefit from the Content Caching and Retrieval System when it is enabled.

During content retrieval, if the Content is made up of multiple segments, the content client must manage that retrieval and concatenate blocks into segments to reconstruct the original content. The unit of retrieval is the block.

DRAFT - FOR PREVIEW ONLY

3 Examples

The following examples illustrate two different methods of metadata retrieval and file retrieval, along with the cases of content available and unavailable.

3.1 Example 1: Reading a File Using SMB 2.1/2 as Metadata Channel in Distributed Cache Mode (Cached Content Available)

This example illustrates multiple use cases, including the following: Configuring SMB 2.1/2 Content Server Caching (section 2.4.4.1.1), Configuring Content Client Caching mode (section 2.4.4.1.3), Using SMB 2.1/2 Metadata Retrieval (section 2.4.4.3.1), and Content Discovery and Retrieval (section 2.4.4.4.4).

The SMB 2.1/2 protocol has two distinct roles to play. The first role is to act as a member of the File Access System by providing the usual file-sharing resources between machines. The second role is to act as a transport for metadata from a content server to a content client, thereby allowing the content client to participate in the Content Caching and Retrieval System.

The initial stages of the read operation from an SMB 2.1/2 protocol operation perspective are the same, regardless of the caching mode (Hosted, Distributed, or none). After hashes have been identified as available (SMB 2.1/2 is required for this), the retrieval mechanism will vary depending on the mode of caching in effect. The caching mode is configurable, either manually or through the use of group policy. The following description has an SMB 2.1/2 stage and then a specific stage for distributed cache mode.

The basic scenario is that two peers, content client (Peer1) and peer role server (Peer2), within a LAN will obtain content by reading, in turn, a file from a content server that is reached over a network link with a latency value greater than the value of the ADM element Network Latency. When the first peer (peer role server) reads the file, there will be no cached content available within the LAN, and therefore the file will need to be retrieved using an SMB 2.1/2 read operation as specified in [MS-FASOD] and [MS-SMB2]. The Content Caching and Retrieval System does not change the requirements related to authentication and access.

After at least 32 MB (1 MB equals, one whole segment) of content has been retrieved for the first time and is available on the LAN, there is potential for the content to be cached and made available to other peers on the LAN.

Initial System State and Prerequisites

- Content Caching and Retrieval is enabled on the content server as described in the Configuring SMB 2.1/2 Content Server Caching (section 2.4.4.1.1) use case.
- The mode of the content caching as distributed cache is configured on the content client (Peer1) and peer role server (Peer2) as described in the Configuring Content Client Caching Mode (section 2.4.4.1.3) use case.
- The desired content is not available on the local cache of the content client (Peer1).
- The cached content available on the peer role server (Peer2) for the desired content of the content client (Peer1).

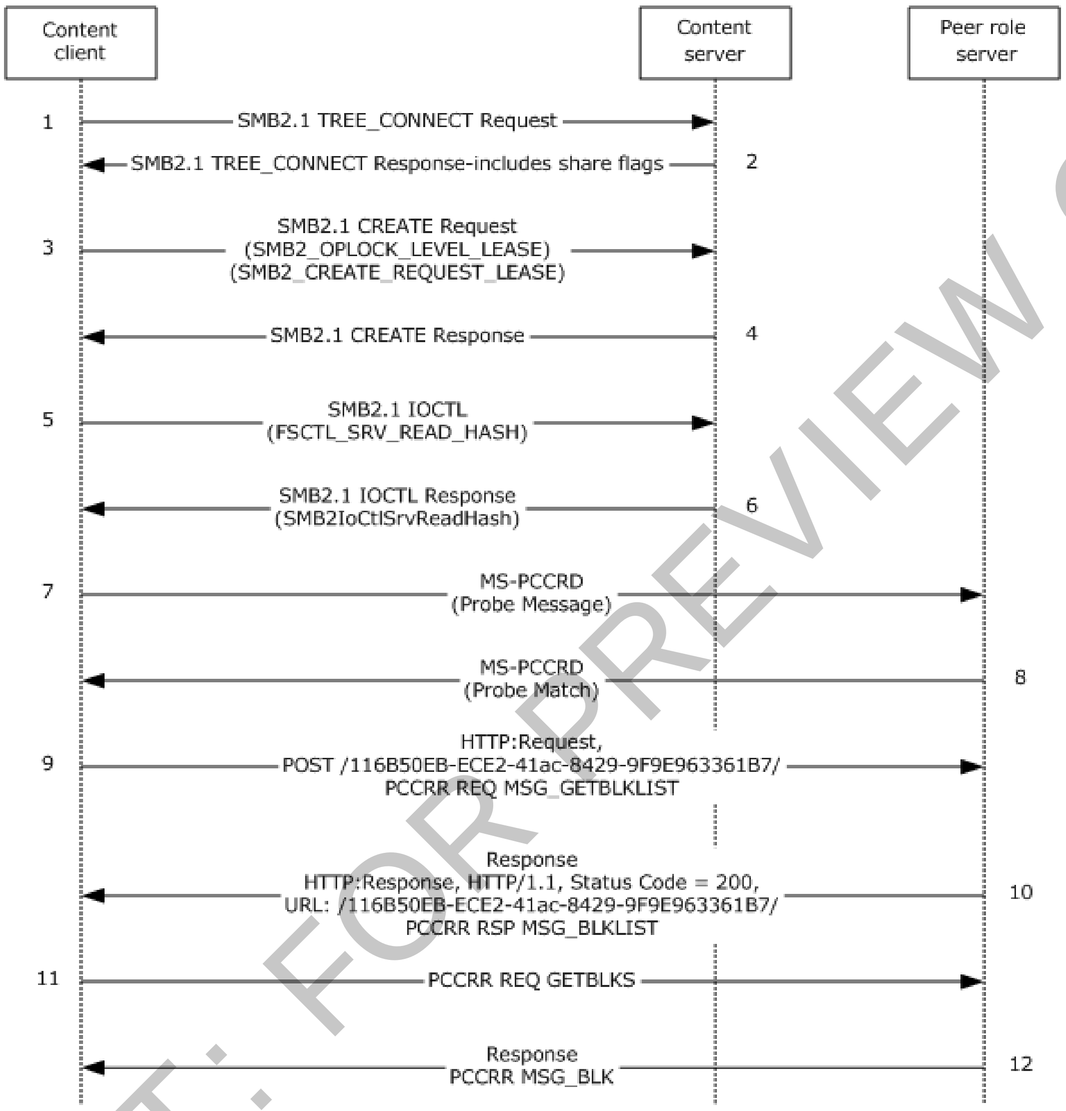
Message Sequence

This example begins with the normal sequence of events that would occur when reading a file using the SMB 2.1/2 protocol. The specification for the SMB 2.1/2 protocol can be found in [MS-SMB2],

DRAFT: FOR PREVIEW ONLY

and a description of that protocol's interaction with other protocols can be found in [MS-FASOD]. Details about opening a file using SMB 2.1/2 can be found in [MS-FASOD] and [MS-SMB2].

The initial sequence of afile read is an open operation followed by a SESSION_SETUP and **TREE_CONNECT**. An SMB 2.1/2 **TREE_CONNECT** operation is required to access any share on an SMB 2.1/2 server. The activities up to this point are independent of the Content Caching and Retrieval System hence these are not illustrated in this example.



Content client
Content server
Peer role server
1
SMB2.1 TREE_CONNECT Request
SMB2.1 TREE_CONNECT Response-includes share flags
2
3
SMB2.1 CREATE Request (SMB2_OPLOCK_LEVEL_LEASE) (SMB2_CREATE_REQUEST_LEASE)
SMB2.1 CREATE Response
4
5
SMB2.1 IOCTL (FSCTL_SRV_READ_HASH)
SMB2.1 IOCTL Response (SMB2IoCtlSrvReadHash)
6
7
MS-PCCRD (Probe Message)
MS-PCCRD (Probe Match)
8
9
HTTP:Request, POST /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR REQ MSG_GETBLKLIST
Response HTTP:Response, HTTP/1.1, Status Code = 200, URL: /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR RSP MSG_BLKLIST
10
11
PCCRR REQ GETBLKS
Response PCCRR MSG_BLK
12

Figure 13: SMB 2.1/2 read file stages and distributed mode discovery and retrieval

1. The content client (Peer1) sends an SMB 2.1/2 TREE_CONNECT request to access the requested share on content server.

2. The content server responds with an SMB 2.1/2 **TREE_CONNECT** response with ShareFlags field, SHI1005_FLAGS_ENABLE_HASH 0x00002000 (per [MS-SMB2] section 2.2.10), and indicates that the share supports hash generation for content cache retrieval of data, that caching is enabled on the server share, and that the server can respond to requests for metadata (subject to the

DRAFT: FOR PREVIEW ONLY

service being installed and configured). The flag informs the content client that it can make requests for metadata but does not guarantee that it will be available.

3. The content client attempts to obtain a lease that allows it to cache reads and writes to the data (see [MS-SMB2] section 3.2.4.3.8). The lease is requested as part of the CREATE operation by sending an SMB2_CREATE_REQUEST_LEASE with a RequestedOplockLevel of SMB2_OPLOCK_LEVEL_LEASE.

4. The server replies with a SMB 2.1/2 CREATE response with an SMB2_CREATE_RESPONSE_LEASE that has the requested Lease State value, and a lease is obtained as part of the SMB 2.1/2 CREATE request.

5. The SMB 2.1/2 content client performs a SRV_READ_HASH request (see [MS-SMB2] section 2.2.31.2) on the file with CtlCode set to FSCTL_SRV_READ_HASH to obtain the hash (metadata) for the target file.

6. The content server sends an SMB2 IOCTL response with a SRV_READ_HASH response. If the hash is out of date, an error is returned to the client; if no hash exists for the specified file, an error is returned to the client.<14> A server can choose to update the hash for either of these situations. This example covers the case where no error is returned.

7. After successfully obtaining a file hash, the client calculates the segment identifiers (HoHoDk); see [MS-PCCRC] section 2.2 for the required content. The content client performs a multicast broadcast Probe message (see [MS-PCCRD] section 2.2.1) for the required content. (The broadcast is targeted at any peers listening on the local subnet).

8. Peers that have the required content respond with a unicast ProbeMatch message (see [MS-PCCRD] section 2.2.2.2). In this example, since the peer role server (Peer2) has the required content, it responds with ProbeMatch message. The ProbeMatch message includes segment identifiers and the address of the peer that holds the content. The content client (Peer1) verifies that the response was sent by a peer on the local subnet and checks the HoHoDks to verify that at least one is associated with a Probe message on the Outstanding Probe List (see [MS-PCCRD]).

9. The content client (Peer1) then retrieves data by initiating the transport with the peer by sending an HTTP POST request to the root path of {116B50EB-ECE2-41ac-8429-9F9E963361B7/} of the peer with the data (see [MS-PCCRR]). The initial HTTP POST request carries a REQUEST-MESSAGE (MSG_GETBLKLIST), as specified in [MS-PCCRR] section 2.2.4.2 (carried as an entity body of the HTTP POST request). This is a request for a download of a block list.

10. The peer responds with an HTTP status code of 200 (OK) for the URL /116B50EB-ECE2-41ac-8429-9F9E963361B7/ (see [MS-PCCRR]); the response carries a RESPONSE-MESSAGE (MSG_BLKLIST), as specified in [MS-PCCRR] section 2.2.5.2, indicating the blocks currently available for download from the peer.

11. The content client then sends a number of REQUEST_MESSAGE (GETBLKS) as specified in [MS-PCCRR] section 2.2.4.3, to the peer role server (Peer2) to retrieve all of the required data, as described in [MS-PCCRR].

12. The peer role server (Peer2) responds with RESPONSE_MESSAGE(MSG_BLK) as described in [MS-PCCRR] section 2.2.5.3. Steps 11 and 12 are repeated until all of the required blocks within the identified segment have been retrieved. If multiple segments are required, multiple segment-retrieval sessions may be initiated (steps 9-12) with one or more peer role servers.

Final System State

- Content client (Peer1) has acquired the desired content.

DRAFT: FOR PREVIEW ONLY

- The local cache of the content client (Peer1) has been updated with the desired content.

3.2 Example 2: Reading a File Using HTTP as the Metadata Channel in Hosted Cache Mode

This example illustrates the multiple use cases, including the following: Configure HTTP Content Server Caching (section 2.4.4.1.2), Configure a Hosted Cache Server (section 2.4.4.1.4), Configure Content Client Caching Mode (section 2.4.4.1.3), HTTP Metadata Retrieval (section 2.4.4.3.2), Content Discovery and Retrieval (cached data unavailable) (section 2.4.4.4.1), and Content Discovery and Retrieval with Hosted Cache (Cached Data Available) (section 2.4.4.4.2) use cases.

In this example, the HTTP protocol has two distinct roles to play. The first role is to act in the usual manner as a file retrieval protocol. The second role is through the use of PeerDist, as described in [MS-PCCRTP], to act as a transport for metadata from a content server to a content client, thereby allowing the content client to participate in the initial stages of the Content Caching and Retrieval System.

The message sequence steps for this example are organized into the following sections:

Peer1:

A. Metadata retrieval over HTTP.

B. Content discovery in hosted cache mode (content unavailable).

C. Return trip to the content server for content download.

D. Content client offers the content to the hosted cache server.

E. Hosted cache server downloads the content from the content client.

Peer2:

F. Metadata retrieval over HTTP.

G. Content discovery and download from the hosted cache server (content available).

As shown in the preceding list, Peer1 is the content client for messages of sections A, B, C, D, and E, whereas Peer2 is the content client for messages of sections F and G.

Initial System State and Prerequisites

- The Content Caching and Retrieval is enabled on the content server as described in the Configure HTTP Content Server caching (section 2.4.4.1.2) use case.
- Hosted cache server configured with required X.509 certificate to enable the server authentication with SSL/TLS.
- The mode of the content caching as "hosted cache" is configured on the server as described in the Configuring a Hosted Cache Server (section 2.4.4.1.4) use case.
- The desired content is not available on the hosted cache server but is available on the content server.
- The hosted cache server is configured to require client authentication.

DRAFT: FOR PREVIEW ONLY

- The mode of the content caching as "hosted client" is configured on the content clients (for example, Peer1 and Peer 2), as described in the Configure Content Client Caching Mode (section 2.4.4.1.3) use case.
- The content clients (for example, Peer1 and Peer2) are configured with the **FQDN** of the hosted cache server.
- On both content clients (Peer1 and Peer2), the content transfer is initiated by the use of Internet Explorer.

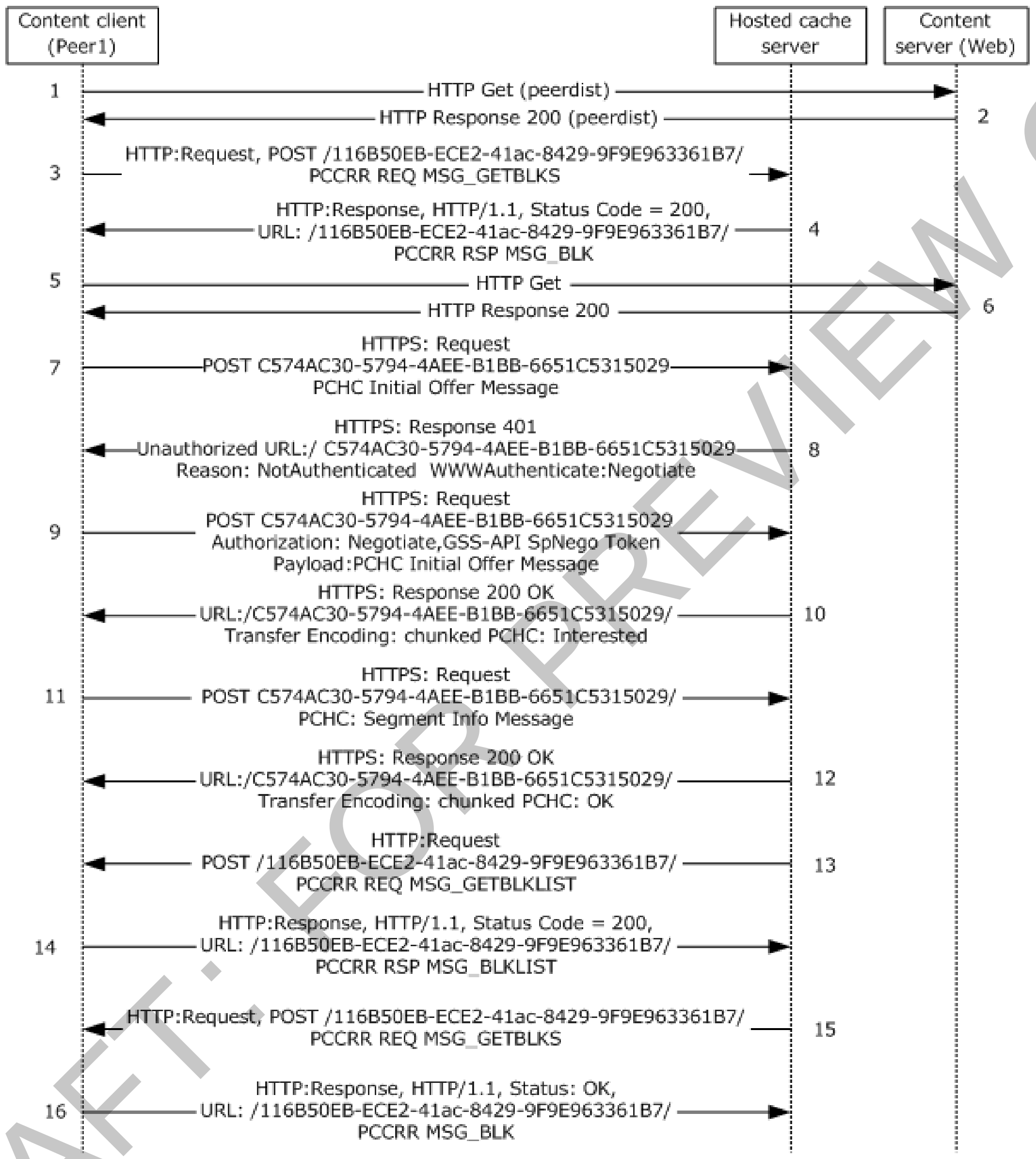

Content client (Peer1)
Hosted cache server
Content server (Web)
1
HTTP Get (peerdist)
HTTP Response 200 (peerdist)
2
3
HTTP:Request, POST /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR REQ MSG_GETBLKS
HTTP:Response, HTTP/1.1, Status Code = 200, URL: /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR RSP MSG_BLK
4
5
HTTP Get
HTTP Response 200
6
7
HTTPS: Request POST C574AC30-5794-4AEE-B1BB-6651C5315029 PCHC Initial Offer Message
HTTPS: Response 401 Unauthorized URL:/ C574AC30-5794-4AEE-B1BB-6651C5315029 Reason: NotAuthenticated WWWAuthenticate:Negotiate
8
9
HTTPS: Request POST C574AC30-5794-4AEE-B1BB-6651C5315029 Authorization: Negotiate,GSS-API SpNego Token Payload:PCHC Initial Offer Message
HTTPS: Response 200 OK URL:/C574AC30-5794-4AEE-B1BB-6651C5315029/ Transfer Encoding: chunked PCHC: Interested
10
11
HTTPS: Request POST C574AC30-5794-4AEE-B1BB-6651C5315029/ PCHC: Segment Info Message
HTTPS: Response 200 OK URL:/C574AC30-5794-4AEE-B1BB-6651C5315029/ Transfer Encoding: chunked PCHC: OK
12
HTTP:Request POST /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR REQ MSG_GETBLKLIST
13
14
HTTP:Response, HTTP/1.1, Status Code = 200, URL: /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR RSP MSG_BLKLIST
HTTP:Request, POST /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR REQ MSG_GETBLKS
15
16
HTTP:Response, HTTP/1.1, Status: OK, URL: /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR MSG_BLK

Figure 14: Content retrieval and offering (Peer1)

A. HTTP metadata Retrieval

DRAFT: FOR PREVIEW ONLY

1. The content client (Peer1) sends an HTTP GET request to the content server. The content client indicates in the HTTP request header that it is PeerDist-enabled by listing PeerDist as an accepted encoding. It also declares the version of PeerDist encoding, for example, X-P2P-PeerDist: Version=1.0 as described in [MS-PCCRTP] section 3.1.4.

2. In the HTTP 200 response, the content server indicates that the content is encoded using the PeerDist content encoding. It also includes the content length and indicates the length of the metadata, with the metadata constructed as described in [MS-PCCRTP] section 3.2.5.1.

B. Content discovery in hosted cache mode (content unavailable)

3. The content client (Peer1) initiates the transport with the hosted cache server by sending an HTTP POST request to the root path of {116B50EB-ECE2-41ac-8429-9F9E963361B7/} of the server. That request also carries a REQUEST-MESSAGE (MSG_GETBLKS) message, as specified in [MS-PCCRR] section 2.2.4.3 (carried as an entity body of the HTTP POST) to the hosted cache server requesting content of block index Zero within the target segment that the content client is interested in.

4. The hosted cache server responds with an HTTP status code of 200 (OK) for the URL /116B50EB-ECE2-41ac-8429-9F9E963361B7/ and RESPONSE-MESSAGE (MSG_BLK), as described in [MS-PCCRR] section 2.2.5.3, which indicates that content is not available for download from the hosted cache server.

C. Return trip to the content server for content download

The content client (Peer1) downloads the desired content from the content server using HTTP protocol.

5. The content client (Peer1) sends an HTTP GET request to the content server to download the content.

6. The content server responds with an HTTP status code of 200 (OK) and also the desired content.

D. Content client offers the content to the hosted cache server

The initial SSL handshake is not discussed; only the traffic shown is pertinent to Content Caching and Retrieval. The URL on which the hosted cache server listens is normally https://:<port number>/C574AC30-5794-4AEE-B1BB-6651C5315029/. The port number is configurable but would normally be 443 for HTTPS. If the hosted cache server is configured to something other than port 443, content clients also need to be configured to use that port.

7. The content client (Peer1) sends a request message as the payload of an HTTP POST request to the hosted cache server. The message is a PCHC initial offer of content to the Hosted Cache; see message 7 (as specified in [MS-PCHC] section 2.2.1.3) in the figure in this section.

8. The Hosted Cache Server responds with an HTTP 401, unauthorized message. The HTTP response is as follows.

```
Http: Response, HTTP/1.1, Status: Unauthorized, URL: /C574AC30-5794-4AEE-B1BB-
6651C5315029/, Using Negotiate
Date: Authentication
```

DRAFT - FOR PREVIEW ONLY

```
    ProtocolVersion: HTTP/1.1
    StatusCode: 401, Unauthorized
    Reason: NotAuthenticated
  + ContentType:  text/html
    Server:  Microsoft-HTTPAPI/2.0
  - WWWAuthenticate: Negotiate
```

The content client (Peer1) and hosted cache server use the mechanisms described in [RFC4559], which are based on GSS-API [RFC2743]. The exact number of packets exchanged during client authentication is dependent on the authentication mechanism<15> and whether the authentication is successful. In any case, if authentication fails for any reason, the content client (Peer1) will not be able to offer content to the hosted cache server. In this example, a successful case of client authentication with Kerberos [MS-KILE] as the authentication mechanism is covered.

9. The content client (Peer1), on receipt of the Unauthorized HTTP response, is expected to repeat the previous POST message, passing an HTTP Authorization header line. In this example, the content client passes the GSS-API SPNEGO token that contains the Kerberos token as the optimistic token in the Authorization header.

10. The hosted cache server verifies the Authorization header, which is received in the previous step, and authenticates the content client (Peer1). After authentication, message processing proceeds as the PCHC initial offer of content is made to the hosted cache server, which then responds with a response code of 1 (see [MS-PCHC] section 2.2.2.2) indicating that it is interested in getting block hashes.

11. The content client responds by sending a SEGMENT_INFO_MESSAGE ([MS-PCHC] section 2.2.1.4) message. This message contains the segment hash of data (HoD) for the previously offered segment, as well as the range of block hashes in the segment.

12. After the hosted cache server obtains the SEGMENT_INFO_MESSAGE message, it responds with an HTTP 200 OK response.

E. Hosted Cache Server downloads the content from Content Client

13. The hosted cache server initiates the transport with the content client (Peer1) by sending an HTTP POST request to the root path of {116B50EB-ECE2-41ac-8429-9F9E963361B7/} of the Server. That request also carries a REQUEST-MESSAGE (MSG_GETBLKLIST), as specified in [MS-PCCRR] section 2.2.4.2, (carried as an entity body of the HTTP POST) to the content client requesting the block IDs of the blocks within the target segment that the hosted cache server is interested in.

14. The content client (Peer1) responds with an HTTP status code of 200 (OK) for the URL /116B50EB-ECE2-41ac-8429-9F9E963361B7/ and a RESPONSE-MESSAGE (MSG_BLKLIST), as specified in [MS-PCCRR] section 2.2.5.2, indicating the blocks currently available for download from the content client (Peer1).

15. The hosted cache server sends a REQUEST-MESSAGE (MSG_GETBLKS) as specified in [MS-PCCRR] section 2.2.4.3 to the content client (Peer1).

16. The content client (Peer1) responds with RESPONSE-MESSAGE (MSG_BLK) as specified in [MS-PCCRR] section 2.2.5.3.

Steps 13 and 14 followed by 15 and 16 are repeated until the required content is transferred.

Peer2:

DRAFT: FOR PREVIEW ONLY

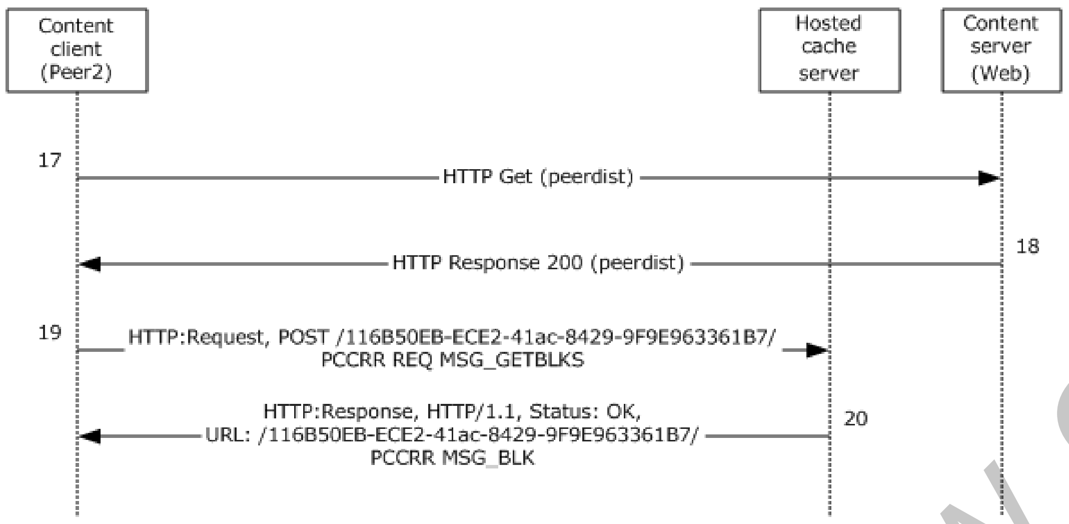

Content client (Peer2)
Hosted cache server
Content server (Web)
17
HTTP Get (peerdist)
18
HTTP Response 200 (peerdist)
19
HTTP:Request, POST /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR REQ MSG_GETBLKS
20
HTTP:Response, HTTP/1.1, Status: OK, URL: /116B50EB-ECE2-41ac-8429-9F9E963361B7/ PCCRR MSG_BLK

Figure 15: Content Retrieval (Peer2)

F. Metadata retrieval over HTTP

17. The content client (Peer2) sends an HTTP GET request to the content server. The content client indicates in the HTTP request header that it is PeerDist-enabled by listing PeerDist as an accepted encoding, as described in [MS-PCCRTP] section 3.1.4.

18. In the HTTP 200 response, the content server indicates that the content is encoded using the PeerDist content encoding. It also includes the content length and indicates the length of the metadata, with the metadata constructed as described in [MS-PCCRTP] section 3.2.5.1.

G. Content discovery and download from hosted cache mode (content available)

19. The content client (Peer2) initiates the transport with the hosted cache server by sending an HTTP POST request to the root path of {116B50EB-ECE2-41ac-8429-9F9E963361B7/} of the server. That request also carries a REQUEST-MESSAGE (MSG_GETBLKS) as specified in [MS-PCCRR] section 2.2.4.3 (carried as an entity body of the HTTP POST) to the hosted cache server requesting content for an interested block (starting from index 0) within the target segment that the content client is interested in.

20. The hosted cache server responds with an HTTP status code of 200 (OK) for the URL /116B50EB-ECE2-41ac-8429-9F9E963361B7/ and a RESPONSE-MESSAGE (MSG_BLK) as specified in [MS-PCCRR] section 2.2.5.3, with the content of the requested block index.

Steps 19 and 20 are repeated until the required content is transferred.

Final System State

- Content clients (for example, Peer1 and Peer2) retrieved the desired content.
- The cached content is available on the hosted cache server.

DRAFT: FOR PREVIEW ONLY

4 Microsoft Implementations

There are no variations in the behavior of the Content Caching and Retrieval system in different versions of Microsoft Windows® beyond those described in the specifications of the protocols supported by the system, as listed in section 2.1.4.

4.1 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:

- 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 2.1.1: Windows Server 2008 R2 is the first server to support the server feature BranchCache retrieval - Hosted Cache Mode. The Hyper-V Core and Home Server SKU are not branch cache capable. All other Windows Server 2008 R2 SKUs can act as branch cache clients and/or content servers.

<2> Section 2.1.1: Windows Server 2008 R2 and Windows 7 support the client feature BranchCache retrieval - Hosted Cache Mode and Distributed Mode. BranchCache retrieval is available with Windows 7 Enterprise and Ultimate.

<3> Section 2.1.2.2.2: In Windows Vista and Windows Server 2008, support for the client-side elements of Content Caching and Retrieval is available only via the optional installation of the Background Intelligent Transfer Service (BITS) via the Windows Management Framework. Support for the server-side elements of this protocol is not available for Windows Server 2008. When the Windows Management Framework is installed, the BITS service use of [MS-BPDP] is replaced by [MS-PCCRD] for discovery and [MS-BPCR] is replaced by [MS-PCCRR] for content retrieval.

<4> Section 2.4.1: In Windows, the Object Store is provided by a local file system, usually NTFS.

<5> Section 2.5: SMB Version 2.1 offers functionality relevant to the Peer Content Caching and Retrieval System. Content Caching was first introduced in the Windows 7 platform and can be made available on Windows Vista and Windows Server 2008 systems by the addition of a Windows Management Framework. This makes the Peer Content Caching and Retrieval System available only to BITS clients.

<6> Section 2.7.1.2.2: In the Windows implementation, the timeout value is set by the higher-layer applications. The recommended value is 5 seconds for each segment retrieval session. The range of permitted timeout values is from 0 to 4,294,967,294 milliseconds.

DRAFT: FOR PREVIEW ONLY

<7> Section 2.7.1.2.2: Windows uses a 15-second lifetime for an entry in the discovery cache with no associated address.

<8> Section 2.7.1.2.2: Windows uses 1 minute as the lifetime for each address that is added to a discovery cache entry.

<9> Section 2.7.2.5.1: The Windows implementation uses 16 as the default maximum number of peers used per download. The number is configurable from 1 to 16,384, inclusive.

<10> Section 2.7.2.5.1: The Windows implementation carries out a simple download each time a download involves less than four consecutive blocks in a single block range, which implies that the blocks are also adjacent (consecutive) in the segment.

<11> Section 2.7.2.5.6: Windows uses a 2-second timeout for each request message. The timeout is configurable between 1 millisecond and 1 minute.

<12> Section 2.7.2.5.6: The client-role peer Windows implementation allows a serving-role peer to time out up to a configurable three times, before excluding the peer from the current download. This value is configurable from 1 to 100, inclusive.

<13> Section 2.9: In Windows 7 and Windows Server 2008 R2, there are two ways for an application to use branch cache retrieval: directly by calling the branch cache platform APIs, or indirectly by calling WinINET/WinHTTP or SMBv2 APIs, which are instrumented to use branch cache, but transparent to the applications.

<14> Section 3.1: Windows 7 clients will offer MS KRB 5, KRB5 SNMPV2, and NTLMSSP if they are configured with the FQDN of the hosted cache server. If a NetBIOS address or IP address is given for the hosted cache server, then only NTLMSSP is offered.

<15> Section 3.2: Windows 7 clients will offer MS KRB 5, KRB5 SNMPV2, and NTLMSSP if they are configured with the FQDN of the hosted cache server. If a NetBIOS address or IP address is given for the hosted cache server, then only NTLMSSP is offered.

DRAFT: FOR PREVIEW ONLY

Copyright © 2011 Microsoft Corporation.

Release: Sunday, October 16, 2011

5 Change Tracking

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

DRAFT: FOR PREVIEW ONLY

6 Index

DRAFT: FOR PREVIEW ONLY

DRAFT: FOR PREVIEW ONLY

Copyright © 2011 Microsoft Corporation.

Release: Sunday, October 16, 2011