Chapter 9
Content-Addressed Storage

KEY CONCEPTS

- Fixed Content and Archives
- Single-Instance Storage
- Object Storage and Retrieval
- Content Authenticity

In the life cycle of information, data is actively created, accessed, edited, and changed. As data ages, it becomes less likely to change and eventually becomes "fixed" but continues to be accessed by multiple applications and users. This data is called *fixed content*.

Traditionally, fixed content was not treated as a specialized form of data and was stored using a variety of storage media, ranging from optical disks to tapes to magnetic disks. While these traditional technologies store content, none of them provide all of the unique requirements for storing and accessing fixed content.

Accumulations of fixed content such as documents, e-mail messages, web pages, and digital media throughout an organization have resulted in an unprecedented growth in the amount of data. It has also introduced the challenge of managing fixed content. Furthermore, users demand assurance that stored content has not changed and require an immediate online access to fixed content. These requirements resulted in the development of Content-Addressed Storage (CAS).

CAS is an *object-based system* that has been purposely built for storing fixed content data. It is designed for secure online storage and retrieval of fixed content. Unlike file-level and block-level data access that use file names and the physical location of data for storage and retrieval, CAS stores user data and its attributes as separate objects. The stored object is assigned a globally unique address known as a *content address (CA)*. This address is derived from the object's binary representation. CAS provides an optimized and centrally managed storage solution that can support *single-instance storage (SiS)* to eliminate multiple copies of the same data.

This chapter describes fixed content and archives, traditional solutions deployed for archives and their limitations, the features and benefits of CAS, CAS architecture, storage and retrieval in a CAS environment, and examples of CAS solutions.

9.1 Fixed Content and Archives

Data is accessed and modified at varying frequencies between the time it is created and discarded. Some data frequently changes, for example, data accessed by an Online Transaction Processing (OLTP) application. Some data that does not typically change, but is allowed to change if required is, for example, bill of material and design documents.

Another category of data is fixed content, which defines data that cannot be changed. X-rays and pictures are examples of fixed content data. It is mandatary for all organizations to retain some data for an extended period of time due to government regulations and legal/contractual obligations. Fixed data, which is retained for future reference or business value, is referred to as fixed content asset. Some examples of fixed content asset include electronic documents, e-mail messages, Web pages, and digital media (see Figure 9-1).

Organizations make use of these digital assets to generate new revenues, improve service levels, and leverage historical value. This demands frequent and quick retrieval of the fixed contents at multiple locations. An *archive* is a repository where fixed content is placed. Online data availability in the archive can further increase the business value of the referenced information.

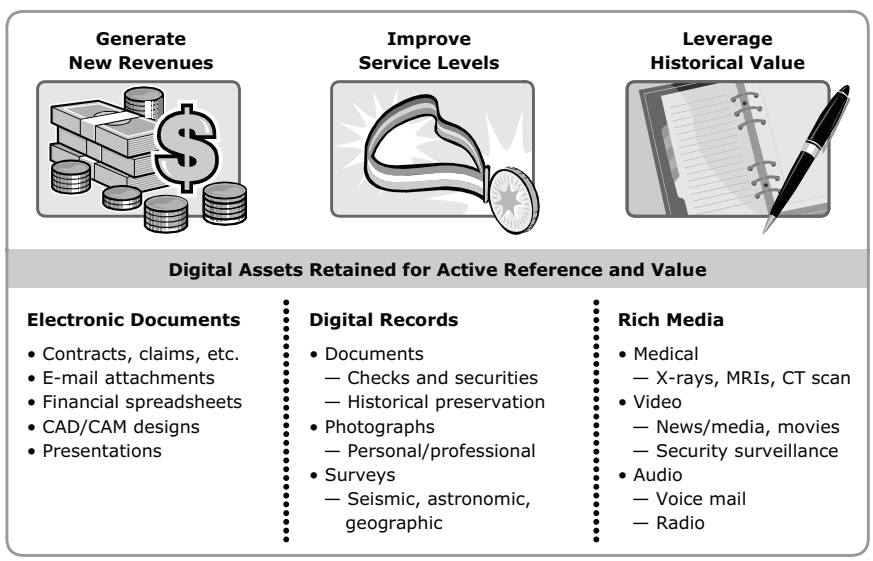

Figure 9-1: Examples of fixed content data

9.2 Types of Archives

An electronic data archive is a repository for data that has fewer access requirements. It can be implemented as online, nearline, or offline based on the means of access:

- **Online archive:** The storage device is directly connected to the host to make the data immediately available. This is best suited for active archives.
- **Nearline archive:** The storage device is connected to the host and information is local, but the device must be mounted or loaded to access the information.
- **Offline archive:** The storage device is not directly connected, mounted, or loaded. Manual intervention is required to provide this service before information can be accessed.

An archive is often stored on a *write once read many (WORM)* device, such as a CD-ROM. These devices protect the original file from being overwritten. Some tape devices also provide this functionality by implementing file-locking capabilities in the hardware or software. Although these devices are inexpensive, they involve operational, management, and maintenance overhead.

Requirements to retain archives have caused corporate archives to grow at a rate of 50 percent or more per year. At the same time, organizations must reduce costs while maintaining required service-level agreements (SLAs). Therefore, it is essential to find a solution that minimizes the fixed costs of the archive's operations and management.

Archives implemented using tape devices and optical disks involve many hidden costs. The traditional archival process using optical disks and tapes is not optimized to recognize the content, so the same content could be archived several times. Additional costs are involved in offsite storage of media and media management. Tapes and optical media are also susceptible to wear and tear. Frequent changes in these device technologies lead to the overhead of converting the media into new formats to enable access and retrieval.

Government agencies and industry regulators are establishing new laws and regulations to enforce the protection of archives from unauthorized destruction and modification. These regulations and standards affect all businesses and have established new requirements for preserving the integrity of information in the archives. These requirements have exposed the hidden costs and shortcomings of the traditional tape and optical media archive solutions.

COMPLIANCE REQUIREMENTS

Businesses such as banking, finance, and health care have to comply with standards enforced by regulators for archived data. These rules detail the regulatory requirements for maintaining the authenticity, integrity, and availability of all business records, contracts, legal documents, and business communications in electronic data formats. These regulations may also state that all businesses are required to inform their customers when their electronic data is compromised.

SEC Rule 17a-3 and 17a-4 of the Securities Exchange Act of 1934, the Sarbanes-Oxley Act, and the Health Insurance Portability and Accountability Act (HIPAA) are some examples of these regulations.

9.3 Features and Benefits of CAS

CAS has emerged as an alternative to tape and optical solutions because it overcomes many of their obvious deficiencies. CAS also meets the demand to improve data accessibility and to properly protect, dispose of, and ensure service-level agreements for archived data. The features and benefits of CAS include the following:

- **Content authenticity:** It assures the genuineness of stored content. This is achieved by generating a unique content address and automating the process of continuously checking and recalculating the content address for stored objects. Content authenticity is assured because the address assigned to each piece of fixed content is as unique as a fingerprint. Every time an object is read, CAS uses a hashing algorithm to recalculate the object's content address as a validation step and compares the result to its original content address. If the object fails validation, it is rebuilt from its mirrored copy.
- **Content integrity:** Refers to the assurance that the stored content has not been altered. Use of hashing algorithm for content authenticity also ensures content integrity in CAS. If the fixed content is altered, CAS assigns a new address to the altered content, rather than overwrite the original fixed content, providing an audit trail and maintaining the fixed content in its original state. As an integral part of maintaining data integrity and audit trail capabilities, CAS supports parity RAID protection in addition to mirroring. Every object in a CAS system is systematically checked in the background. Over time, every object is tested, guaranteeing content integrity even in the case of hardware failure, random error, or attempts to alter the content with malicious intent.
- **Location independence:** CAS uses a unique identifier that applications can leverage to retrieve data rather than a centralized directory, path

names, or URLs. Using a content address to access fixed content makes the physical location of the data irrelevant to the application requesting the data. Therefore the location from which the data is accessed is transparent to the application. This yields complete content mobility to applications across locations.

- **Single-instance storage (SiS):** The unique signature is used to guarantee the storage of only a single instance of an object. This signature is derived from the binary representation of the object. At write time, the CAS system is polled to see if it already has an object with the same signature. If the object is already on the system, it is not stored, rather only a pointer to that object is created. SiS simplifies storage resource management tasks, especially when handling hundreds of terabytes of fixed content.
- **Retention enforcement:** Protecting and retaining data objects is a core requirement of an archive system. CAS creates two immutable components: a data object and a meta-object for every object stored. The meta-object stores object's attributes and data handling policies. For systems that support object-retention capabilities, the retention policies are enforced until the policies expire.
- **Record-level protection and disposition:** All fixed content is stored in CAS once and is backed up with a protection scheme. The array is composed of one or more storage clusters. Some CAS architectures provide an extra level of protection by replicating the content onto arrays located at a different location. The disposition of records also follows the stringent guidelines established by regulators for shredding and disposing of data in electronic formats.
- **Technology independence:** The CAS system interface is impervious to technology changes. As long as the application server is able to map the original content address the data remains accessible. Although hardware changes are inevitable, the goal of CAS hardware vendors is to ensure compatibility across platforms.
- **Fast record retrieval:** CAS maintains all content on disks that provide subsecond "time to first byte" (200 ms–400 ms) in a single cluster. Random disk access in CAS enables fast record retrieval.

9.4 CAS Architecture

The CAS architecture is shown in Figure 9-2. A client accesses the CAS-Based storage over a LAN through the server that runs the CAS API (application programming interface). The CAS API is responsible for performing functions that enable an application to store and retrieve the data.

CAS architecture is a *Redundant Array of Independent Nodes (RAIN)*. It contains storage nodes and access nodes networked as a cluster by using a private LAN that is internal to it. The internal LAN can be reconfigured automatically to detect the configuration changes such as the addition of storage or access nodes. Clients access the CAS on a separate LAN, which is used for interconnecting clients and servers to the CAS.

The nodes are configured with low-cost, high-capacity ATA HDDs. These nodes run an operating system with special software that implements the features and functionality required in a CAS system.

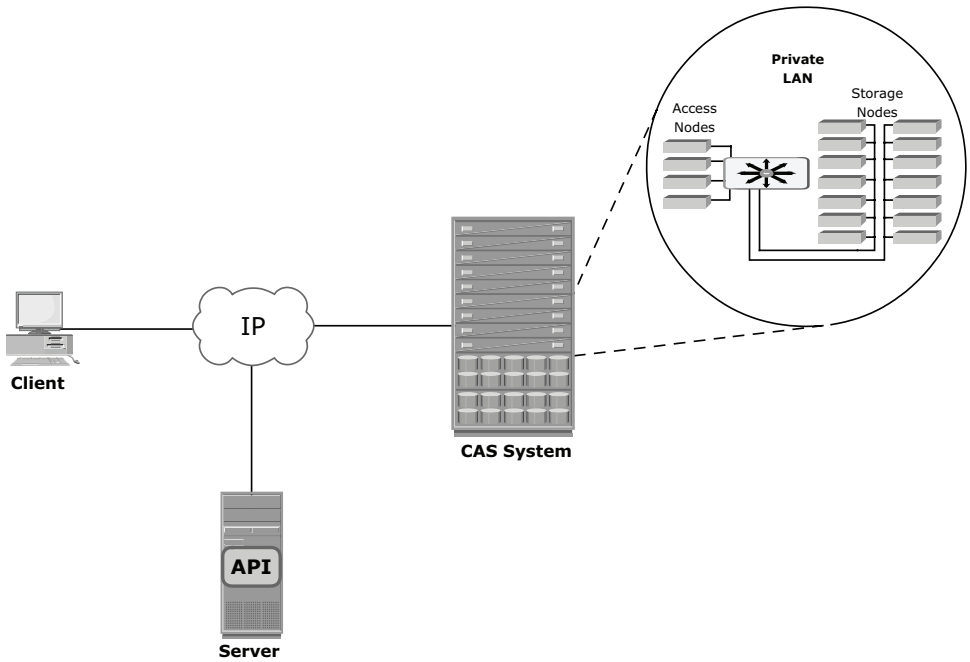

Figure 9-2: CAS Architecture

When the cluster is installed, the nodes are configured with a "role" defining the functionality they provide to the cluster. A node can be configured as a storage node, an access node, or a dual-role node. *Storage nodes* store and protect data objects. They are sometimes referred to as *back-end nodes*. *Access nodes* provide connectivity to application servers through the customer's LAN. They establish connectivity through a private LAN to the storage nodes in the cluster. The number of access nodes is determined by the amount of user required throughput from the cluster. If a node is configured solely as an "access node," its disk space cannot be used to store data objects. This configuration is generally found in older installations of CAS. Storage and retrieval requests are sent to the access node via the customer's LAN. *Dual-role nodes* provide both storage

and access node capabilities. This node configuration is more typical than a pure access node configuration.

Almost all CAS products have the same features and options. Some may be implemented differently, but the following features are an essential part of any CAS solution:

- **Integrity checking:** It ensures that the content of the file matches the digital signature (hashed output or CA). The integrity checks can be done on every read or by using a background process. If problems are identified in any of the objects the nodes automatically repair or regenerate the object.
- **Data protection and node resilience:** This ensures that the content stored on the CAS system is available in the event of disk or node failure. Some CAS systems provide local replication or mirrors that copy a data object to another node in the same cluster. This decreases the total available capacity by 50 percent. Parity protection is another way to protect CAS data. It uses less capacity to store data, but takes longer to regenerate the data if corrupted. Remote replication copies data objects to a secondary storage device in a remote location. Remote replication is used as a disaster-recovery solution or for backup. Replication technologies are detailed in Chapters 13 and 14.
- **Load balancing:** Distributes data objects on multiple nodes to provide maximum throughput, availability, and capacity utilization.
- **Scalability:** Adding more nodes to the cluster without any interruption to data access and with minimum administrative overhead.
- **Self diagnosis and repair:** Automatically detects and repairs corrupted objects and alert the administrator of any potential problem. These failures can be at an object level or a node level. They are transparent to the users who access the archive. CAS systems can be configured to alert remote support teams who diagnose and make repairs remotely.
- **Report generation and event notification:** Provides on-demand reporting and event notification. A command-line interface (CLI) or a graphical user interface (GUI) can produce various types of reports. Any event notification can be communicated to the administrator through syslog, SNMP, SMTP, or e-mail.
- **Fault tolerance:** Ensures data availability if a component of the CAS system fails, through the use of redundant components and data protection schemes. If remote replication of CAS is implemented, failover to the remote CAS system occurs when the primary CAS system is unavailable.
- **Audit trails:** Enable documentation of management activity and any access and disposition of data. Audit trails are mandated by compliance requirements.

9.5 Object Storage and Retrieval in CAS

The process of storing and retrieving objects in CAS is explained in Figures 9-3 and 9-4, respectively. This process requires an understanding of the following CAS terminologies:

- **Application programming interface (API):** A high-level implementation of an interface that specifies the details of how clients can make service requests. The CAS API resides on the application server and is responsible for storing and retrieving the objects in a CAS system.
- **Access profile:** Used by access applications to authenticate to a CAS cluster and by CAS clusters to authenticate themselves to each other for replication.
- **Virtual pools:** Enable a single logical cluster to be broken up into multiple logical groupings of data.
- **Binary large object (BLOB):** The actual data without the descriptive information (metadata). The distinct bit sequence of user data represents the actual content of a file and is independent of the name and physical location.
- **Content address (CA):** An object's address, which is created by a hash algorithm run across the binary representation of the object. While generating a CA, the hash algorithm considers all aspects of the content, returning a unique content address to the user's application.

  A unique number is calculated from the sequence of bits that constitutes file content. If even a single character changes in the file, the resulting CA is different. A *hash output,* also called a *digest*, is a type of fingerprint for a variable-length data file. This output represents the file contents and is used to locate the file in a CAS system. The digest can be used to verify whether the data is authentic or has changed because of equipment failure or human intervention. When a user tries to retrieve or open a file, the server sends the CA to the CAS system with the appropriate function to read the file. The CAS system uses the CA to locate the file and passes it back to the application server.
- **C-Clip:** A virtual package that contains data (BLOB) and its associated CDF. The *C-Clip ID* is the CA that the system returns to the client application. It is also referred as a *C-Clip handle* or *C-Clip reference*.
- **C-Clip Descriptor File (CDF):** An XML file that the system creates while making a C-Clip. This file includes CAs for all referenced BLOBs and associated metadata. Metadata includes characteristics of CAS objects such as size, format, and expiration date.

Referring to Figure 9-3, the data object storage process in a CAS system is as follows:

1. End users present the data to be archived to the CAS API via an application. The application server may also interact directly with the source (e.g., an X-ray machine) that generated this fixed content.
2. The API separates the actual data (BLOB) from the metadata and the CA is calculated from the object's binary representation.
3. The content address and metadata of the object are then inserted into the C-Clip Descriptor File (CDF). The C-clip is then transferred to and stored on the CAS system.
4. The CAS system recalculates the object's CA as a validation step and stores the object. This is to ensure that the content of the object has not changed.
5. An acknowledgment is sent to the API after a mirrored copy of the CDF and a protected copy of the BLOB have been safely stored in the CAS system. After a data object is stored in the CAS system, the API is given a C-Clip ID and C-Clip ID is stored local to the application server.
6. Using the C-Clip ID, the application can read the data back from the CAS system.

   Once an object is stored successfully, it is made available to end users for retrieval and use. The content address is usually hidden from the user. A user accesses the file stored on CAS by the same file name. It is the application server that references the CA to retrieve the stored content; this process is transparent to the user. No modification is needed to the application's user interface to accommodate the CAS storage and retrieval process.

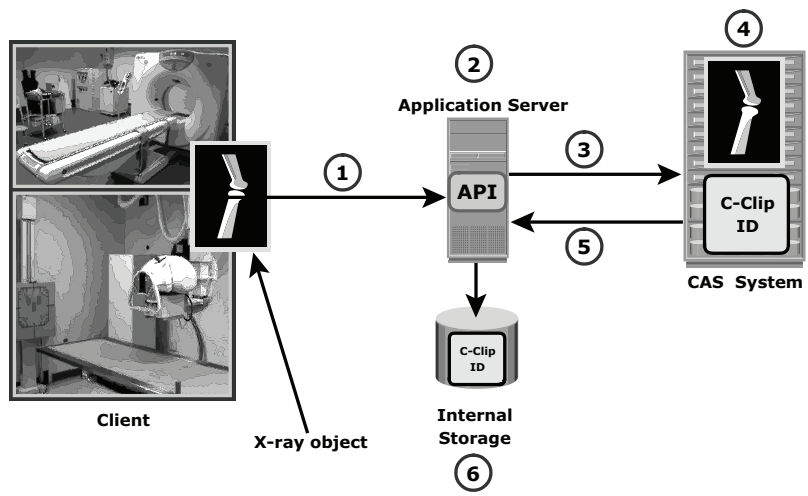

Figure 9-3: Storing data objects on CAS

The process of data retrieval from CAS follows these steps:

1. The end user or an application requests an object.
2. The application queries the local table of C-Clip IDs stored in the local storage and locates the C-Clip ID for the requested object.
3. Using the API, a retrieval request is sent along with the C-Clip ID to the CAS system.
4. The CAS system delivers the requested information to the application, which in turn delivers it to the end user.

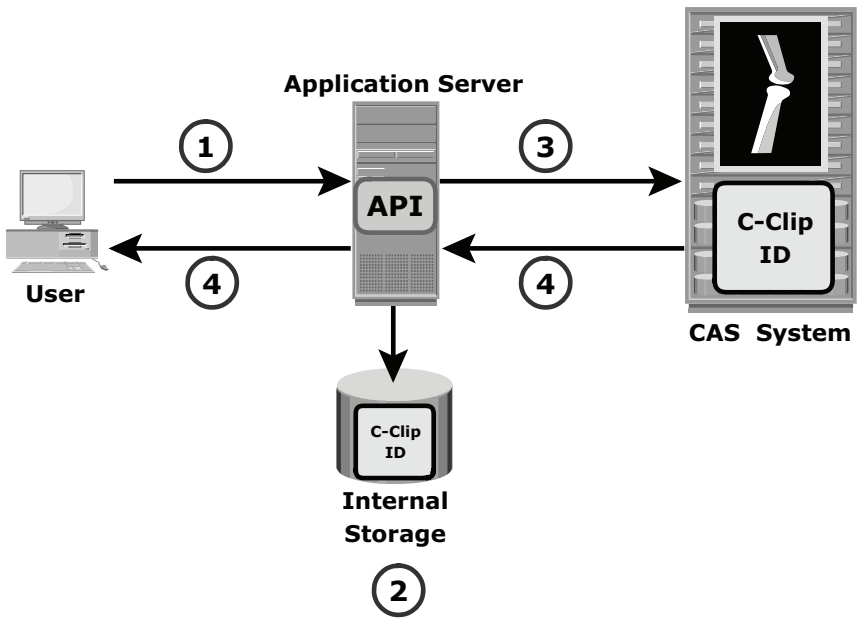

Figure 9-4: Data object retrieval from CAS system

9.6 CAS Examples

Organizations have deployed CAS solutions to solve several business problems. Two solutions are described in detail in the following sections.

9.6.1 Health Care Solution: Storing Patient Studies

A large health care center examines hundreds of patients every day and generates large volumes of medical records. Each record may be composed of one or more images that range in size from about 15 MB for standard digital X-ray images to

over 1 GB for oncology studies. The patient records are stored online for a period of 60–90 days for immediate use by attending physicians. Even if a patient's record is no longer needed for any reason, HIPAA requirements stipulate that the records should be kept in the original formats for at least seven years.

Beyond 90 days, hospitals may backup images to tape or send them to an offsite archive service for long-term retention. The cost of restoring or retrieving an image in long-term storage may be five to ten times more than leaving the image online. Long-term storage may also involve extended recovery time, ranging from hours to days. Medical image solution providers offer hospitals the capability to view medical records, such as online X-ray images, with sufficient response times and resolution to enable rapid assessments of patients. Figure 9-5 illustrates the use of CAS in this scenario. The patient records are moved to the CAS system after 60-90 days. This facilitates long-term storage and at the same time when immediate access is needed, the records are available by accessing the CAS system.

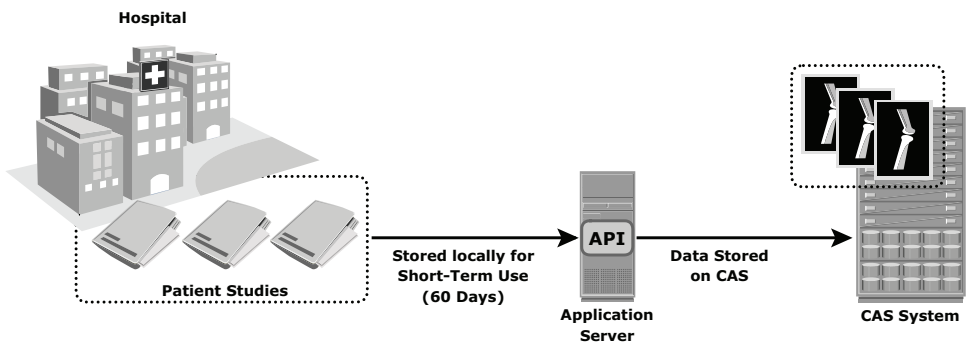

Figure 9-5: Storing patient studies on CAS system

9.6.2 Finance Solution: Storing Financial Records

In a typical banking scenario, images of checks, each about 25 KB in size, are created and sent to archive services over an IP network. A check imaging service provider may process 50–90 million check images per month. Typically, check images are actively processed in transactional systems for about five days.

For the next 60 days, check images may be requested by banks or individual consumers for verification purposes at the rate of about 0.5 percent of the total check pool, or about 250,000–450,000 requests. Beyond 60 days, access requirements drop drastically, to as few as one for every 10,000 checks. Figure 9-6 illustrates the use of CAS in this scenario. The check images are stored on a CAS system, starting at day 60 and can be held there indefinitely.

A typical check image archive can approach a size of 100 TB. Check imaging is one example of a financial service application that is best serviced with CAS. Customer transactions initiated by e-mail, contracts, and security transaction records may need to be kept online for 30 years; CAS is the preferred storage solution in such cases.

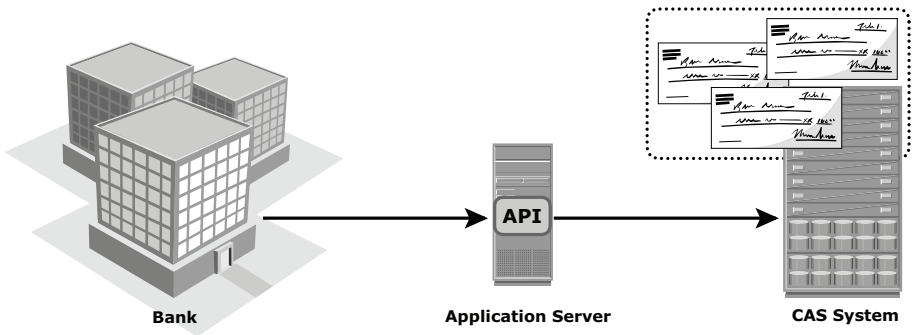

Figure 9-6: Storing finiancial records on CAS system

HIERARCHICAL STORAGE MANAGEMENT (HSM)

Business organizations need to move data between storage tiers for many reasons, including cost, protection, and compliance, depending on its value over time. HSM is a policy-based management system of file backup and archiving. The policies are established to store data on different tiers of storage based on the relevance and importance of the information. An example of a policy is if a file on a high-performance storage tier is not accessed for 120 days, it should be migrated to a low performance storage tier or archived on the CAS system. At the same time, when users need that migrated file, access to it must be provided seamlessly. HSM implements such policies and automates the process of migration and recall of data from different tiers of storage. The process of migration is completely transparent to the user. After the migration, the original file on the high-performance storage is replaced with a *stub file*, which acts as a placeholder and looks like the original file to the user.

9.7 Concepts in Practice: EMC Centera

EMC Centera is a simple, affordable, and secure repository for information archiving. EMC Centera is the first platform designed and optimized specifically to deal with the storage and retrieval of fixed content, meeting performance,

compliance, and regulatory requirements. Compared to traditional archive solutions, EMC Centera provides faster record retrieval, SiS, guaranteed content authenticity, self-healing, and support for numerous industry regulatory standards. Visit `http://education.EMC.com/ismbook` for the latest information.

9.7.1 EMC Centera Models

EMC Centera is offered in three different models to meet different types of user requirements—EMC Centera Basic, EMC Centera Governance Edition, and EMC Centera Compliance Edition Plus (CE+):

- **EMC Centera Basic:** Provides all functionality without the enforcement of retention periods.
- **EMC Centera Governance Edition:** Provides the retention capabilities required by organizations to responsibly manage electronic records in addition to the features provided by EMC Centera Basic. Deploying Governance Edition enforces organizational and application policies for information retention and disposition.
- **CE+:** Provides extensive compliance capabilities. CE+ is designed to meet the requirements of the most stringent of regulated business environments for electronic storage media as established by regulations from the Securities and Exchange Commission (SEC), or other national and international regulatory groups.

9.7.2 EMC Centera Architecture

The architecture of the RAIN-based EMC Centera system is designed to be highly scalable and to store petabytes of content. While the EMC Centera cabinet can accommodate 32 nodes, the entry level configuration starts with as few as 4 nodes and 2 internal switches and is expandable in increments of 4 nodes. The nodes run a Linux operating system and CentraStar software to implement all the CAS functionality. A collection of nodes and internal switches is referred to as a cube. A cube contains two internal switches and a maximum of 16 nodes. A fully configured cabinet is comprised of two cubes, as shown in Figure 9-7.

The EMC Centera node contains four SATA HDDs and a dual-source power supply that enables EMC Centera nodes to connect to two power sources. Each power outlet on a node connects to a separate power rail.

Figure 9-8 illustrates the EMC Centera architecture. Each node contains more than 1 TB of usable capacity and can be configured as access and/or storage nodes. EMC Centera has two 24-port 2 gigabit internal switches that provide communications for up to 16 nodes within the private LAN. Several cabinets

of these nodes and switches can be connected to form an EMC Centera cluster. An EMC Centera domain consists of one or more clusters. Applications may support multiple clusters within a EMC Centera domain. Each cabinet can host up to 23 TB. The usable protected capacity varies depending on whether it uses *content protection parity (CPP)* or *content protection mirrored (CPM)*.

In CPP, the data is fragmented into segments, with an additional parity segment. Each segment is on a different node, similar to a file-type RAID. If a node or a disk fails, the other nodes regenerate the missing segment on a different node.

In CPM, each data object is mirrored and each mirror resides on a different node (refer to Figure 9-8). If a node or a disk fails, the EMC Centera software automatically broadcasts to the node with the mirrored copy to regenerate another copy to a different node so that two copies are always available.

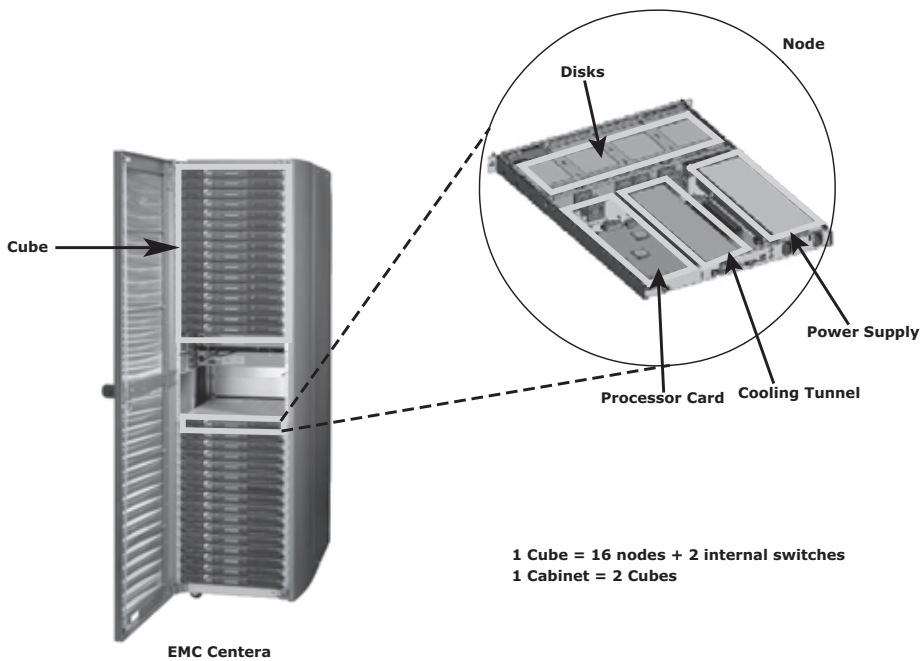

Figure 9-7: EMC Centera configuration

Both CPP and CPM provide total protection against failure using EMC Centera's unique *self-healing* functions. With the self-healing feature, if any component in the node or the entire node fails, data is regenerated to a different part of the cluster, ensuring that data is always protected. In addition to this "organic regeneration" process, there are other processes that run continuously in the background, verifying objects by scrubbing and ensuring that objects are not corrupted. The self-managing and configuring functionality enable rapid installation and implementation of EMC Centera.

EMC Centera protects users from technology changes by allowing various generations to coexist in a single CAS cluster.

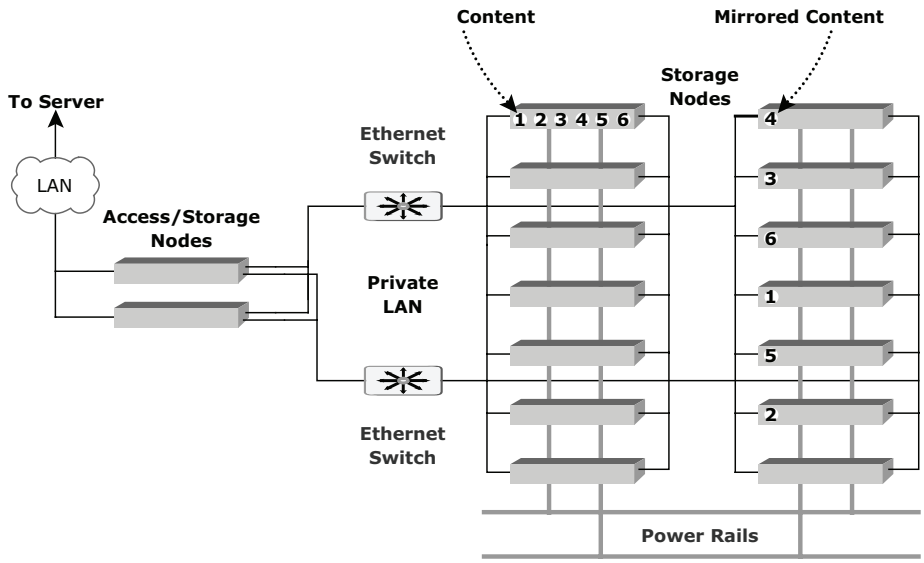

Figure 9-8: EMC Centera architecture

9.7.3 Centera Tools

A group of tools are available to users and service personnel to manage the functions of EMC Centera. They include EMC Centera Viewer, EMC Centera Monitor, EMC Centera Console, and EMC Centera Health Reporting:

- **EMC Centera Viewer:** It is a GUI that is loaded to a client that has network access to EMC Centera. The tool provides a simple means of displaying EMC Centera's capacity utilization and operational performance. It also enables the system administrator to change any site-specific information, such as the public network information and end-user contact information. EMC Centera Viewer is commonly used by service personnel to perform maintenance and to upgrade the CentraStar code.
- **EMC Centera Monitor:** It is a tool that enables users to monitor a single cube in EMC Centera, by displaying system properties such as configuration, capacity, and software version.

- **EMC Centera Console:** It is a web-based management tool that enables system operators to view detailed information about alerts, configurations, performance, and relationships between multiple EMC Centera clusters.
- **EMC Centera Health Reporting:** It is done through an automatic e-mail message that an EMC Centera cluster periodically sends to the EMC Customer Support Center or a list of predefined recipients. The message reports the current status of the EMC Centera cluster. This enables remote monitoring, diagnosis, and support for EMC Centera hardware and software.

9.7.4 EMC Centera Universal Access

An important feature of EMC Centera, when compared to traditional archive solutions, is that the EMC Centera archive is online. In addition, the EMC Centera archive is accessible from any application or platform. EMC Centera Universal Access acts as a high-performance store and forward protocol translator. It communicates with application servers using network file protocols (NFS, CIFS, HTTP) and with an EMC Centera cluster through Centera API (see Figure 9-9).

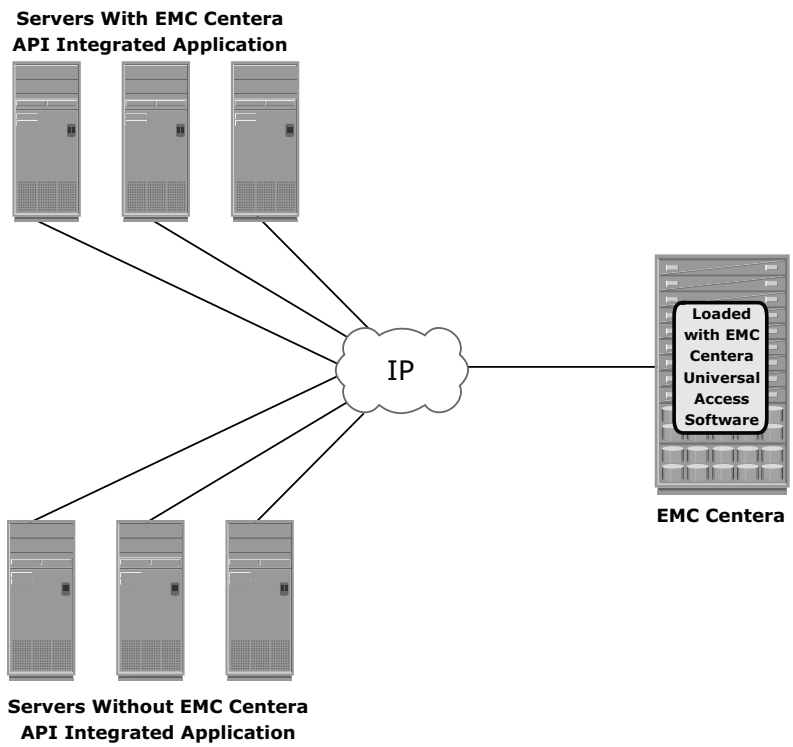

Figure 9-9: EMC Centera Universal Access

With EMC Centera Universal Access, any enterprise application that can mount a network drive or use FTP and HTTP can take advantage of EMC Centera's benefits. From home-grown applications to nonintegrated versions of applications, EMC Centera Universal Access makes it possible to utilize EMC Centera in customer environments with no change to existing applications. This greatly simplifies and accelerates deployment.

Summary

Understanding the information lifecycle enables storage designers to adopt different storage technology to meet data archival requirements. CAS offers a centrally managed networked storage solution for fixed content. CAS has enabled IT organizations to realize significant cost savings and improve operational efficiency and data protection. CAS meets stringent regulatory requirements that have helped organizations avoid penalties and issues associated with regulatory noncompliance. CAS eliminates data duplication with SiS, which reduces IT maintenance costs and increases long-term return on investment for storage systems.

This chapter outlined the challenges involved in managing fixed content, the CAS architecture and its benefits, and how CAS is deployed and managed, using EMC Centera as an example.

Storage networking technologies have provided the primary architecture that has evolved and matured to meet the business's information storage demands. As this evolution continues, organizations have started using multi-vendor and multi-generation technologies that must work together without compromising business requirements. The virtualization technologies described in the next chapter provides more flexibility and scalability to meet next-generation storage needs.

EXERCISES

1. **Explain how a CAS solution fits into the ILM strategy.**
2. **To access data in a SAN, a host uses a physical address known as a logical block address (LBA). A host using a CAS device does not use (or need) a physical address. Why?**
3. **The IT department of a department store uses tape to archive data. Explain 4–5 major points you could provide to persuade the IT department to move to a CAS solution. How would your suggestions impact the IT department?**