Open Blueprint
Multimedia Resource Manager

Open Blueprint Multimedia Resource Manager

If you especially like or dislike anything about this paper, please tell us what you think. Feel free to comment on specific errors or omissions, accuracy, organization, subject matter, or completeness.

By FAX, use this number:

FAX (United States and Canada): 914+432-9405
FAX (Other Countries):
Your International Access Code +1+914+432-9405

If you prefer to send comments electronically, use this network ID:

IBMLink (United States customers only): KGNVMC(D58PUBS)
IBM Mail Exchange: USIB2NZL at IBMMAIL
Internet: d58pubs@vnet.ibm.com

© **Copyright International Business Machines Corporation 1995. All rights reserved.**
Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

Contents

About This Paper v
Who Should Read This Paper v
Open Blueprint Structure vi

Notices vii
Trademarks and Service Marks vii

Open Blueprint Multimedia Resource Manager 1
Introduction 1
The Management of Local Multimedia Resources 1
The Need to Dynamically Connect Multimedia Logical Devices 3
Supporting Distributed Multimedia 4
The Management of Multimedia Distributed Resources 6
Multimedia Resource Manager Components 9
The Collaboration Resource Manager Multimedia Logical Device 11
Open Distributed Multimedia Computing Migration Considerations 13
Summary 14

© Copyright IBM Corp. 1995

Figures

1. Open Blueprint Structure vi
2. The Multimedia Programming Model 2
3. The Multimedia Distributed File System 3
4. A Waveform and MIDI Audio Mixing Player 4
5. The Distributed Multimedia Programming Model 5
6. A Distributed Multimedia File Player 6
7. Multimedia Connections for Two-Party Collaboration 7
8. Multimedia (Distributed) Resource Manager in the Open Blueprint 8
9. Multimedia Resource Manager Components 10
10. A Multimedia Application Server 11
11. The Collaboration Resource Manager Multimedia Logical Device 12
12. The Internal Structure of Source and Destination CRM Multimedia Logical Devices 13
13. The Internal Structure of the CRM Multimedia Logical Devices to Achieve Interoperability 14

© Copyright IBM Corp. 1995

About This Paper

This paper describes the Multimedia Resource Manager component of the Open Blueprint and its relationships with other Open Blueprint components. The Open Blueprint expresses IBM's technical approach to distributed computing in a multivendor, heterogeneous environment.

The Open Blueprint (see Figure 1) structures the distributed systems environment and provides the base upon which to build, run, and manage distributed applications.

The Open Blueprint is based on a combination of existing and emerging industry standards. This standards base satisfies customer requirements for products and solutions which:

- Connect and work seamlessly
- Run on many industry hardware and software platforms
- Integrate with products and solutions from IBM and other industry providers

The Open Blueprint paves the way for a computing environment with a “single system image.” The network appears to the user as a single system with all functions integrated and accessible.

Open, distributed computing or client/server computing is the cornerstone of current information technology. Any strategy for this rapidly changing environment must be flexible and dynamic to accommodate technological advances, while at the same time protect significant current investments. So, while this document is a snapshot at a specific point in time, the Open Blueprint and the details behind it will evolve as new technologies take hold. A good example is object-oriented technology that is influencing all aspects of open, distributed computing. While work on the Open Blueprint began with the “procedural” model, the Open Blueprint has evolved to encompass object-oriented technologies as well.

The Open Blueprint serves several audiences:

- It helps *customers* develop their own architecture and organize products and applications in an open, distributed environment.
- It informs *customers*, *software vendors*, *consultants*, *system integrators*, and *service providers* about IBM's directions for products and solutions.
- It guides *developers* as they meet users' needs by supplying products and solutions that include appropriate functions and that can be integrated and can interoperate with other installed products.

Who Should Read This Paper

This paper is intended for audiences requiring technical detail about the Multimedia Resource Manager in the Open Blueprint. These include:

- Customers who are planning technology or architecture investments
- Software vendors who are developing products to interoperate with other products that support the Open Blueprint
- Consultants and service providers who offer integration services to customers

© Copyright IBM Corp. 1995

Open Blueprint Structure

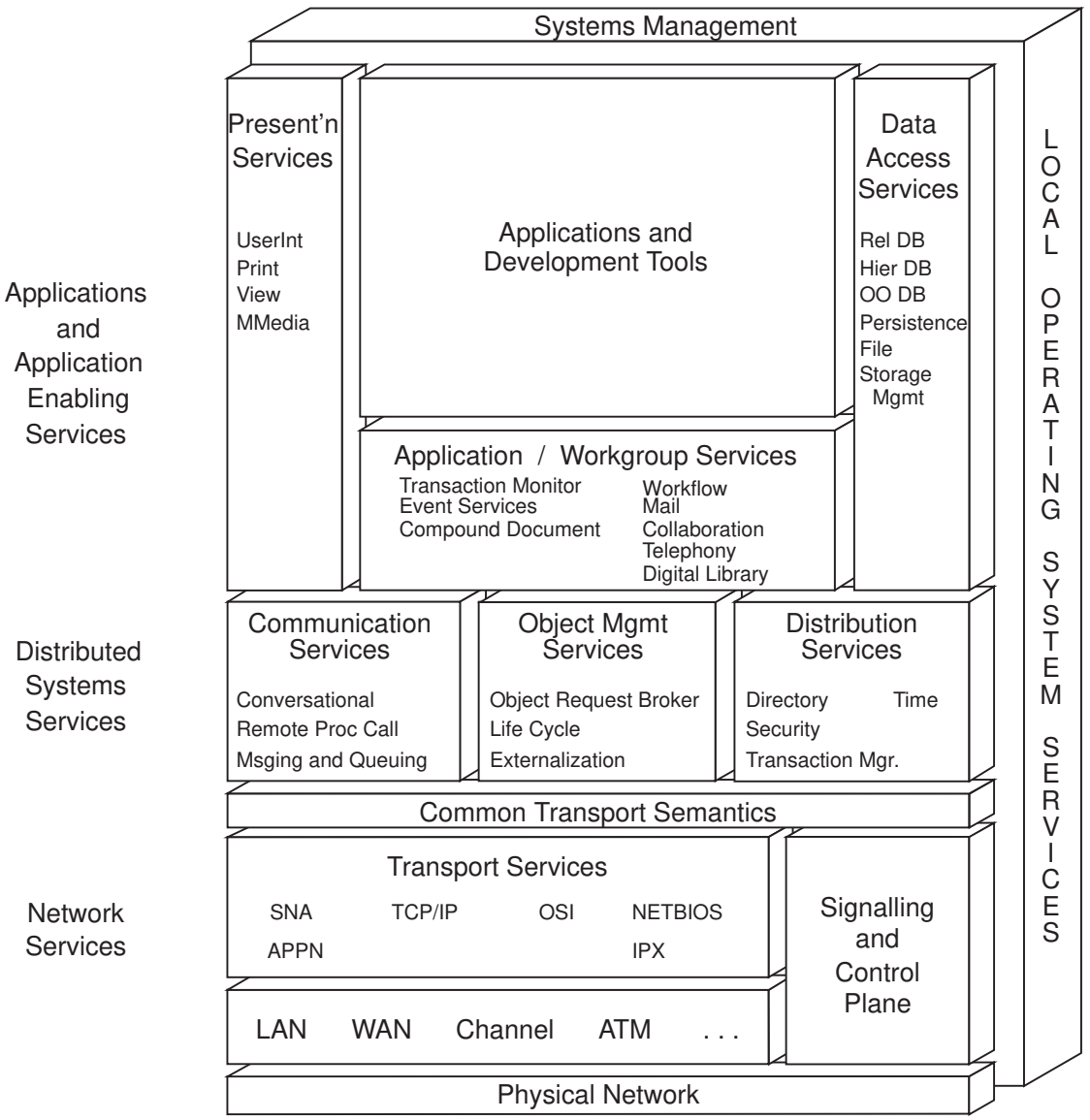

Figure 1. Open Blueprint Structure

This paper is one in a series of papers available in the *Open Blueprint Technical Reference Library* collection, SBOF-8702, (hardcopy) or SK2T-2478-00 (CD ROM), whose intent is to provide the most detailed level of information about each component. The authors of these papers are the developers and designers directly responsible for the components; therefore you may observe differences in style, scope, and format between this paper and others.

Readers less familiar with a particular component area may need to use some of the indicated reference materials to gain basic background knowledge not included in the papers. For a general technical overview of the Open Blueprint, see the *Open Blueprint Technical Overview*, GC23-3808.

Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM's product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any of IBM's intellectual property rights may be used instead of the IBM product, program, or service. Evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, is the user's responsibility.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
USA

Trademarks and Service Marks

The following terms are trademarks or service marks of the IBM Corporation in the United States and/or other countries:

AIX/6000
IBM
IBMLink
OS/2

Open Blueprint is a trademark or registered trademark of the IBM Corporation.

IBM encourages companies to use the Open Blueprint as a model for the development of their information technology architectures. The content and concepts within the Open Blueprint also can be applied to many client/server or distributed systems related information technology activities.

The following terms are trademarks or service marks of other companies:

Windows	Microsoft Corporation

© Copyright IBM Corp. 1995

Open Blueprint Multimedia Resource Manager

This description of the Multimedia resource manager represents a strategic architecture for the provision of distributed multimedia services. It is based on the integration of the Multimedia resource manager with the native operating systems Multimedia System Services and the Collaboration resource manager to provide n-ways connections between multimedia information sources and target logical devices. Implementation of the architecture is a long-term objective.

Introduction

The term multimedia has become ambiguous because it is used interchangeably with interactive video, digital video, digital media, new media and so forth. We use the term multimedia to mean two or more digital media, at least one of which is a time-based digital medium: audio, animation, or video. The focus on digital media is a consequence such as to extend the capabilities of existing information systems by treating multimedia “as just another data type.” In truth, it is not that simple. A time-based digital medium when stored can occupy considerable storage space and when recorded or played appears as a continuous stream of data packets, or **data stream**, which must be delivered from source to destination within stringent performance constraints; for example, from the disk drive attached to the multimedia file server across a network right to and including the user interface support on the client system. Insuring these constraints are met in a distributed system's environment is known as **guaranteeing the quality of service (QoS)**. Enabling a PC or workstation to support multimedia greatly expands its communications and information management roles in commercial, entertainment, education, and government applications. To this end, the Multimedia resource manager has been incorporated into the Open Blueprint. The **Multimedia resource manager** employs the services of the **Collaboration resource manager** to interconnect multimedia logical devices. These devices are surrogates for the multimedia resources managed by the native operating systems' **Multimedia System Services**. The result is the enablement of distributed multimedia computing.

The Management of Local Multimedia Resources

The abstract model of multimedia programming employed by the Multimedia System Services component typically found in a desktop operating system represents a multimedia application as using a programming interface to activate, control, and deactivate the resources involved in delivering a data stream from the **Source Multimedia Logical Device** to a **Destination Multimedia Logical Device** in a local environment (see Figure 2 on page 2). A buffer pool in shared memory is used to exchange the data stream from source to destination. In addition, the multimedia programming interface is used by the application to control and synchronize data streams and to notify an application when an event of interest has occurred. The Multimedia System Services is responsible for taking application requests and forwarding them to the appropriate multimedia logical device to control the device. It also sends back replies to service requests and event notifications to the application. The model is an example of what is required to achieve computer control of simple consumer electronics equipment. The application does not handle time-based digital media; it controls multimedia devices, data stream flow, and synchronization. A multimedia logical device is assumed to be **local** and is a generic model of a particular class of multimedia devices: for example, VCR. It also interfaces with the physical device driver of the actual multimedia device installed of that class. In this way an application is shielded from the unique control characteristics of a vendors' multimedia devices. Multimedia System Services is designed to be extensible, and this capability has been used to add support for new classes of multimedia logical devices in successive operating system releases.

© Copyright IBM Corp. 1995

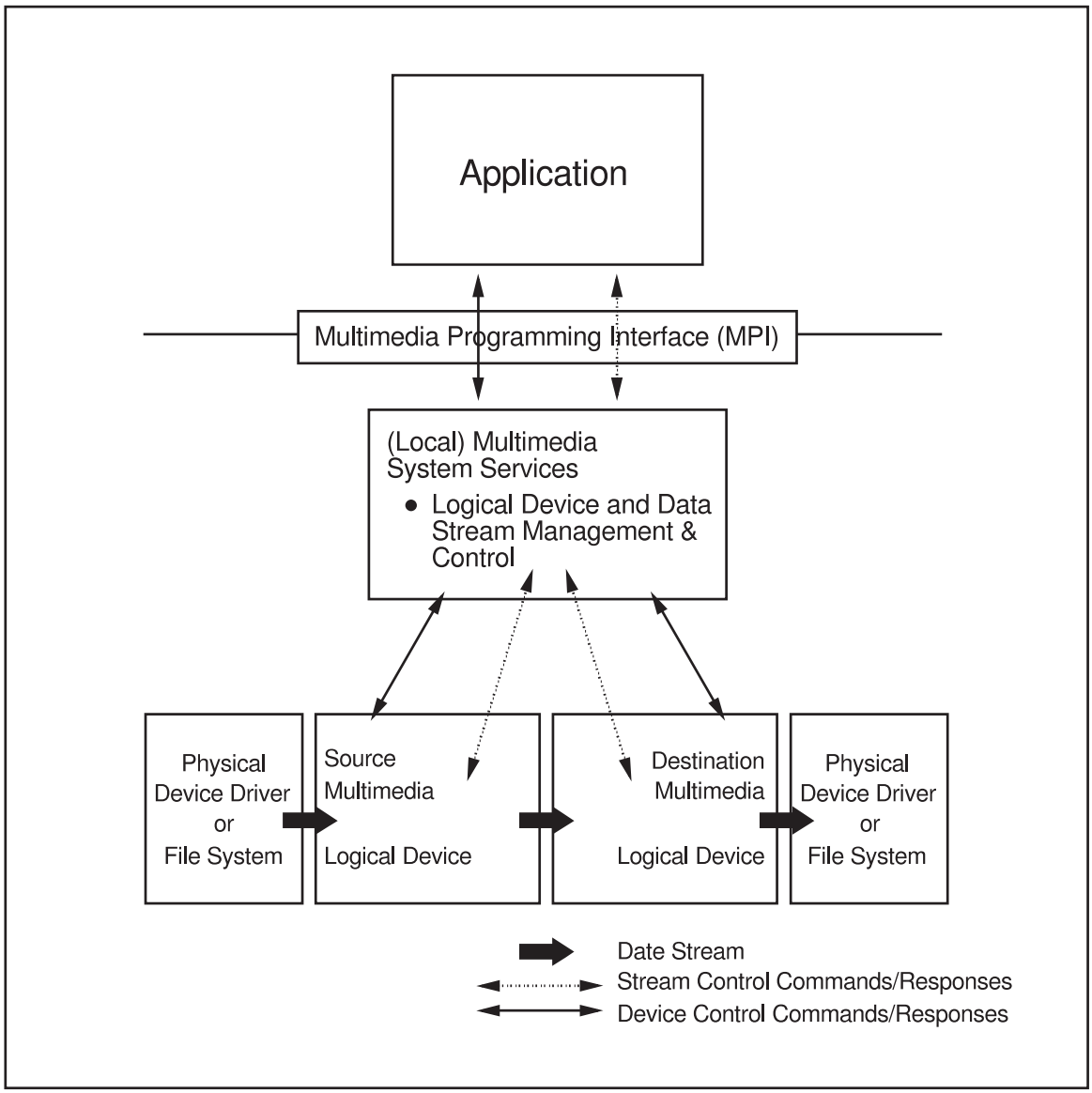

Figure 2. The Multimedia Programming Model

Although the multimedia programming model is targeted for the control of local multimedia resources, it can be utilized to support multimedia applications that request playing of a multimedia file stored either locally or remotely on a multimedia file server. In the remote file server case, as portrayed in Figure 3 on page 3, we see the Source Multimedia Logical Device to be a **proxy** for the actual multimedia data source, the file server. The proxy is interfaced to the local file system. A request to play a file remotely-stored causes the proxy to request the local file system to open the file and retrieve multimedia data records afterwards. The local file system finds that the requested file is on a remote file server and thus redirects all file requests to the file server. To ensure acceptable performance measured in terms of audio/video quality as perceived by the end-user requires the distributed file system infrastructure to support guaranteed QoS. In summary, the desktop operating system's Multimedia System Services depicted in Figure 2 is targeted for the stand-alone PC environment, with the exception that it will utilize an enhanced distributed file system to access remote multimedia files that are delivered in real-time with guaranteed QoS.

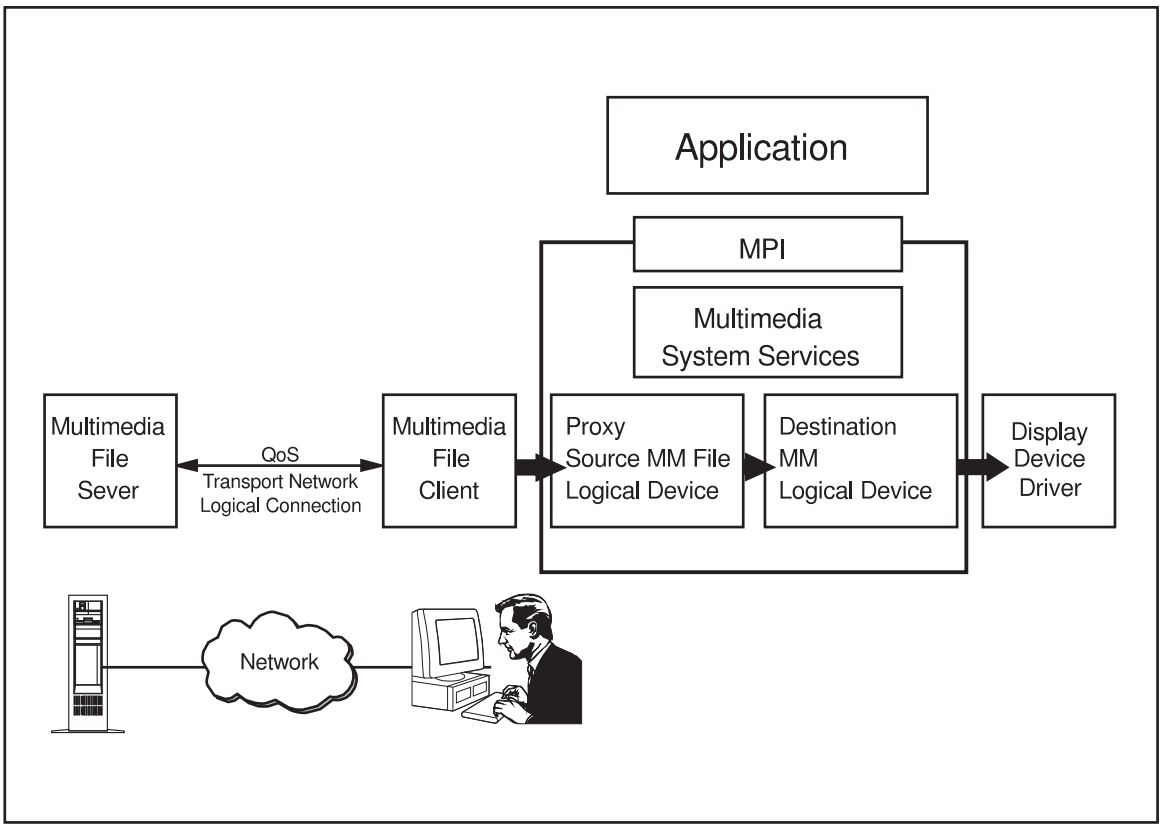

Figure 3. The Multimedia Distributed File System

The Need to Dynamically Connect Multimedia Logical Devices

One premise underlying the original multimedia programming model was that it was sufficient for the multimedia devices installed on a PC system to be preconfigured through static interconnections to form a very limited number of simple configurations. However, due to the growing sophistication of the audio/video equipment attached to a PC, the variety of alternative compression and decompression algorithms, and the desire to mix various audio sources, this premise is no longer true. Figure 4 on page 4 illustrates the situation where an application needs a “network” of multimedia logical devices in order to play the audio data stream that results from mixing a waveform audio data stream with the waveform audio data stream that results after converting a MIDI audio data stream. This is just one of several audio playback “networks” that may be needed by various applications that run on the PC. Thus, in the stand-alone PC or local environment a requirement to dynamically interconnect local multimedia logical devices exists.

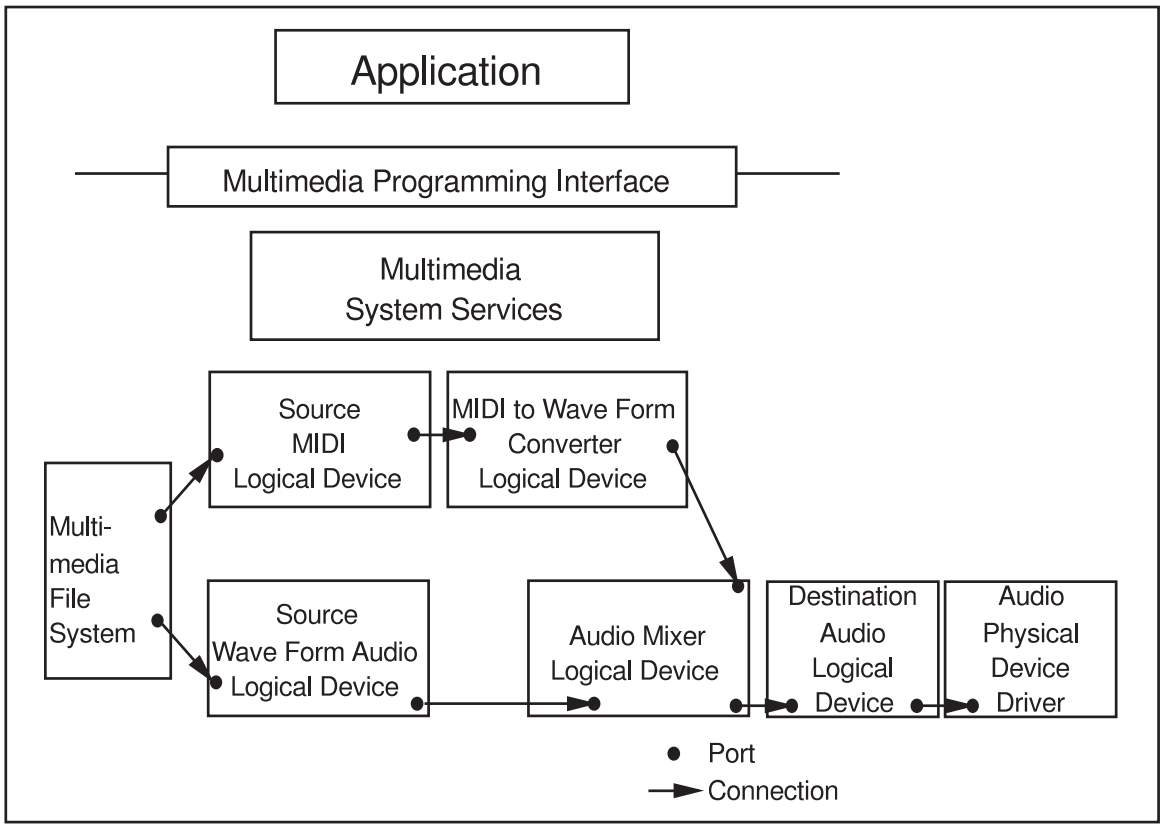

Figure 4. A Waveform and MIDI Audio Mixing Player

Supporting Distributed Multimedia

The emergence of distributed multimedia applications, such as real-time collaboration, digital libraries, and interactive TV have also resulted in the requirement for the ability to dynamically interconnect multimedia logical devices to form a multimedia logical-device "network." As a result, the multimedia programming model has been enhanced to include the capability to set up and take down **logical connections** between logical devices on request. The logical connection supports a single data stream and the associated two-way control flow. It masks whether the data stream flows through memory, across a local bus, or over a transport network. Each end of a connection terminates at a multimedia logical device's port. The capability to dynamically connect multimedia logical devices across a network transforms the Multimedia System Services of a desktop operating system used to control local multimedia resources into a **Multimedia resource manager** that provides **distributed multimedia system services**. Figure 5 on page 5 illustrates the simplest multimedia logical-device configuration consisting of a source multimedia logical device and destination multimedia logical device connected through a logical connection. A device may have several ports which allow it to handle a corresponding number of data streams. Once we introduce the concept that the multimedia programming model needs the capability to dynamically set up a configuration of multimedia logical devices, the multimedia programming interface needs to be correspondingly extended to effect the set up and take down of logical connections having a specified quality of service. To this end, the Multimedia resource manager is responsible for the management and control of multimedia logical device configurations consisting of multimedia logical devices, logical connections, associated ports, and assigned data streams. The objective is to have an application request the Multimedia resource manager to activate a named configuration and handle all the details about activating and interconnecting multimedia logical devices. After the configuration of multimedia resources is activated the application can request the Multimedia resource manager initiate the data stream flow from multimedia source to destination.

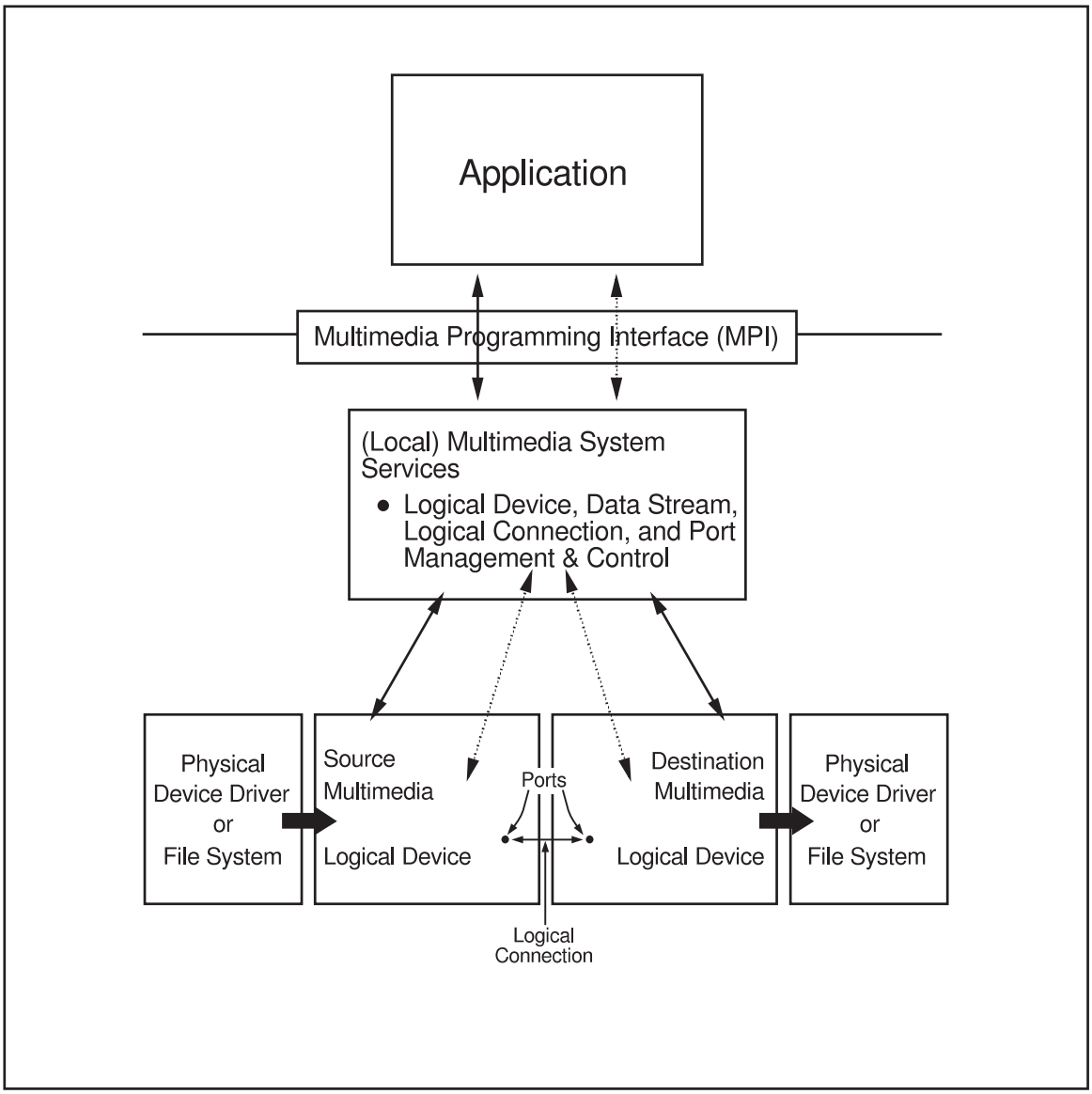

Figure 5. The Distributed Multimedia Programming Model

The introduction of the concept of using ports and logical connections to interconnect multimedia logical devices into the multimedia programming model greatly unifies supporting multimedia in a distributed environment. This point is illustrated in Figure 6 on page 6 where the same configuration of interconnected multimedia logical devices is required to play a multimedia file, whether it is stored locally or remotely. In the local case, the logical connection between the Source Multimedia File Logical Device and the Demultiplexor Logical Device is through shared memory and in the remote case it is through a transport network connection.

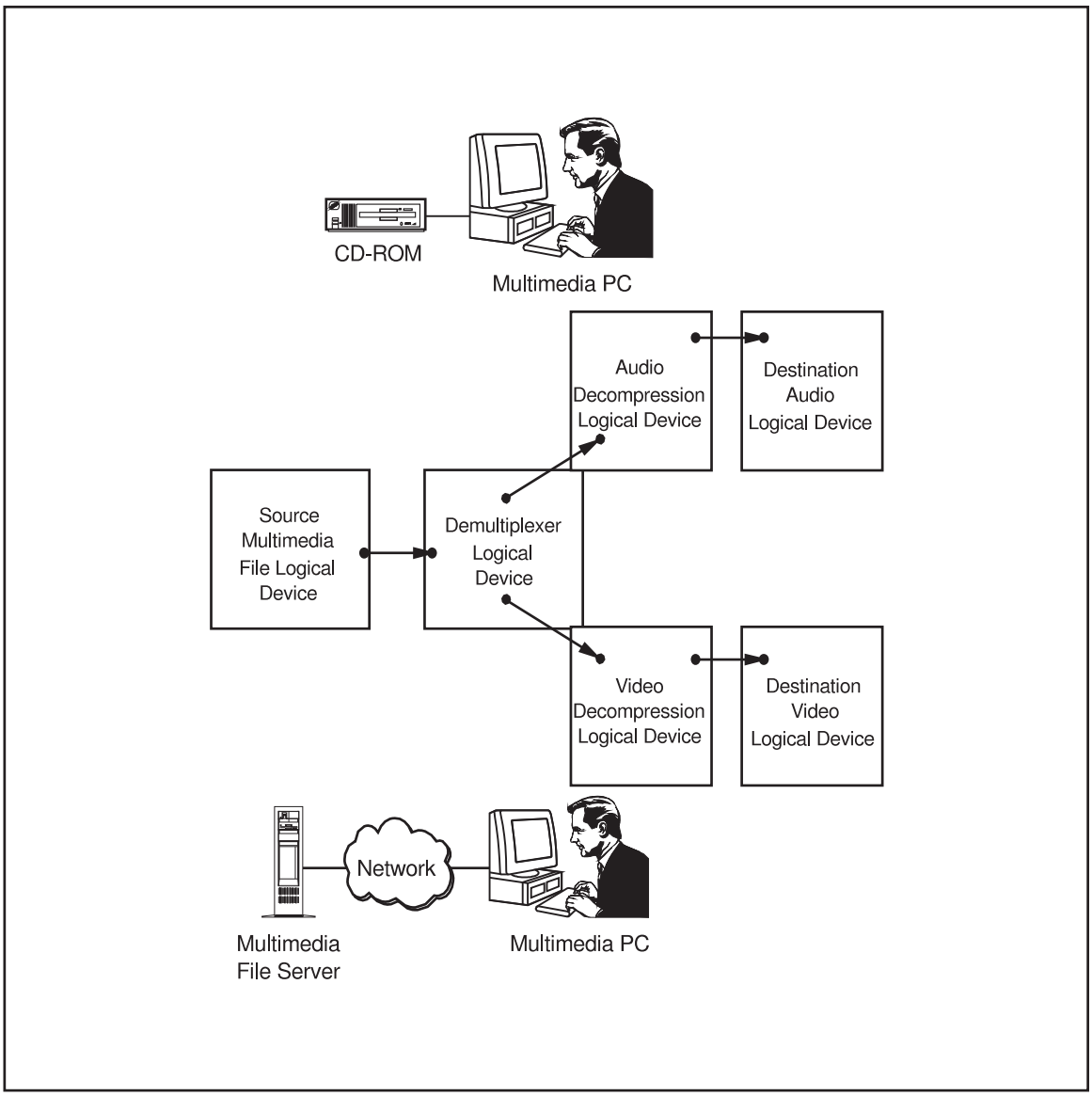

Figure 6. A Distributed Multimedia File Player

The Management of Multimedia Distributed Resources

Real-time collaboration illustrates how important it is to have the Multimedia resource manager support the dynamic configuration of multimedia logical devices. In two-party collaboration, the applications need to be interconnected as do the audio/video equipment attached to each PC. For example, the speaker on each PC needs to be logically connected to the microphone on the other PC, and similarly the video camera attached to each PC needs to be logically connected to the video display window on the other PC. Figure 7 on page 7 shows the logical connections required for a two-party, real-time collaboration. A Mux/Demux Logical Device was included to represent the situation where the objective is to reduce transmission costs by multiplexing the audio and video data streams together over the logical connection that is routed over a network. It is easy to imagine how complex the logical connections become to support multi-party collaborations. In fact, the logical connection types employed by the Multimedia resource manager must support **two-party** and **multi-party collaborations**.

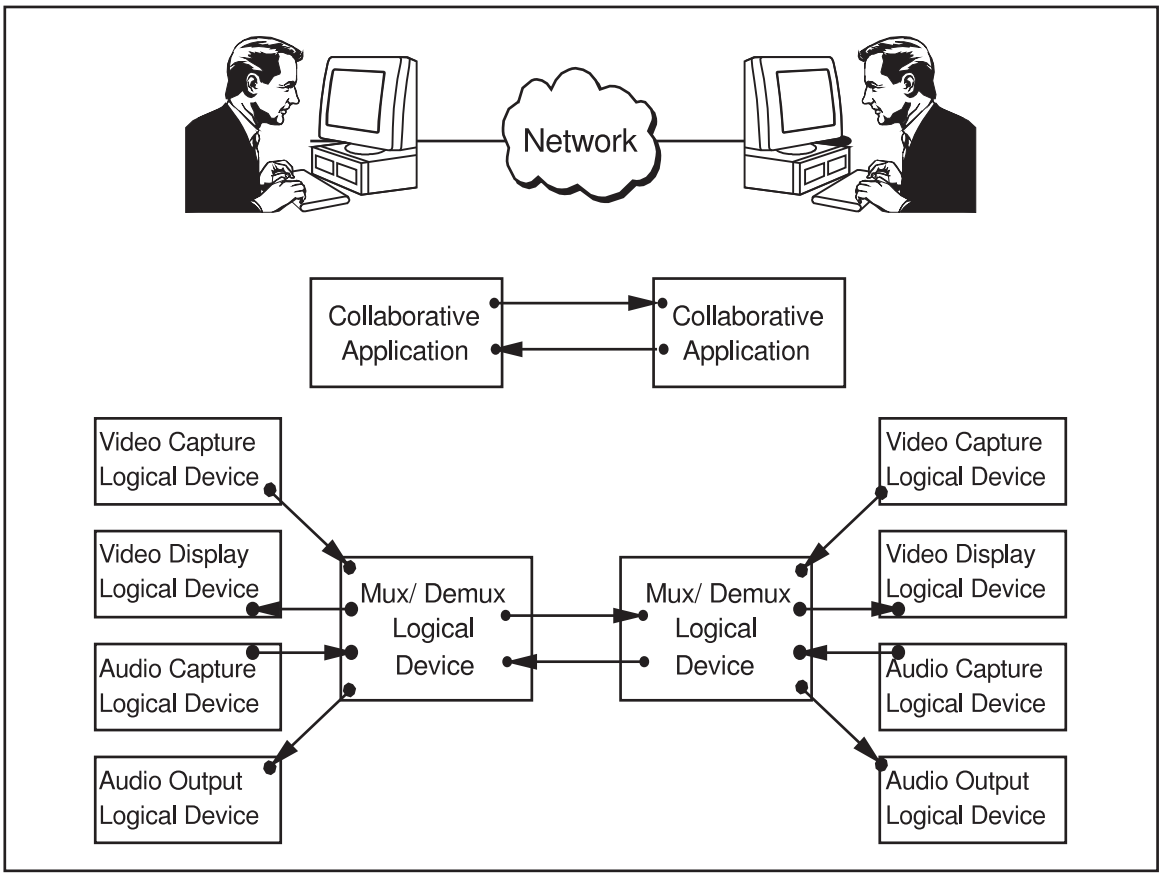

Figure 7. Multimedia Connections for Two-Party Collaboration

In the Open Blueprint, the Multimedia resource manager depends on the **Collaboration resource manager** to provide a comprehensive set of services that support communications that require guaranteed QoS among collaborating applications and/or logical devices. This results from the Collaboration resource manager's need to satisfy audio/video conferencing requirements that arise in real-time collaboration (see Figure 8 on page 8). Thus, to accomplish distributed multimedia computing, the Multimedia resource manager synergistically exploits the capabilities of the Collaboration resource manager to manage the following:

- Logical connections that provide guaranteed QoS (for example, channels)
- Logical devices representing local multimedia resources
- Ports assigned to logical devices and channels
- Events associated with controlling logical devices and data transmissions between devices.

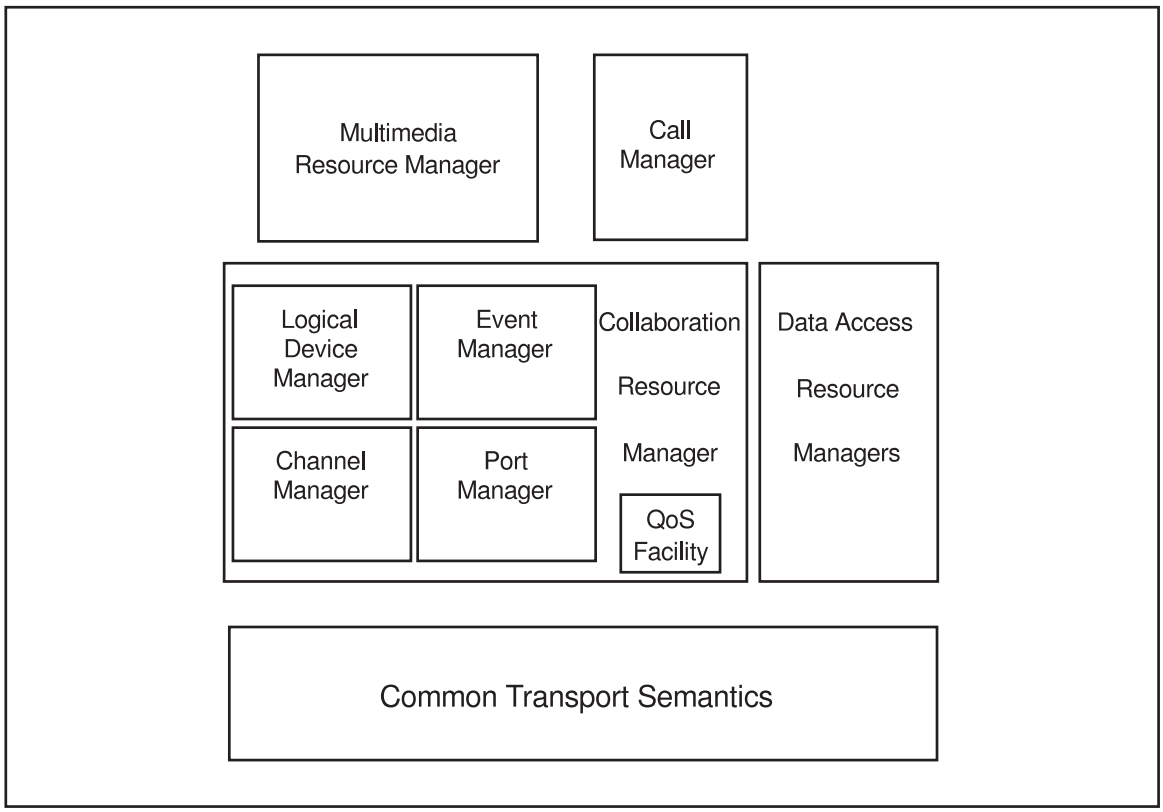

Figure 8. Multimedia (Distributed) Resource Manager in the Open Blueprint

Like any resource manager in IBM's Open Blueprint, both the Multimedia resource manager and the Collaboration resource manager have comprehensive application programming interfaces (APIs). From a multimedia application's perspective, the former's API is used to request the activation of a configuration of multimedia logical devices and to synchronize and control the flow of data streams. In turn, the latter API is employed by the Multimedia resource manager to effect true distributed multimedia services. Although, we've focused on data stream communications among multimedia logical devices, the Collaboration resource manager also permits data streams to flow from logical device to application and the reverse. There are multimedia applications in the area of interactive TV where the capability to dynamically augment a data stream by an application is important. The Collaboration resource manager employs directory services to locate logical devices and determine their characteristics and a call manager to setup connections.

Thus, we have multiple levels of abstraction when it comes to managing multimedia resources in an open distributed system environment. At the highest level, the Multimedia resource manager is targeted to address the portability and interoperability requirements. It is the enabler for distributed multimedia computing by providing an API tailored to manage a configuration of distributed multimedia resources and to synchronize and control multimedia data stream flows. To this end, it employs the services of the Collaboration resource manager to activate a configuration of interconnected multimedia logical devices. These logical devices encapsulate the local multimedia resources actually managed and controlled by the native operating system's Multimedia System Services which may include microphone, speakers, multimedia data files, video camera, video window, and so forth. At the lowest level of abstraction, the Multimedia System Services satisfies the requirement that an application should control generic representations of the various classes of local multimedia devices and not have to deal with vendor-specific device characteristics. The Multimedia System Services available on each operating system (for example, OS/2 WARP, AIX/6000, System 7, and Windows 3.1) differ in terms of multimedia programming interfaces and some of the multimedia logical devices supported. However, each one is extensible. There is a well-specified procedure for adding support for new types of multimedia devices.

Thus, in fact a “network connected” multimedia logical device can be added to each operating system's Multimedia System Services to begin addressing distributed multimedia computing requirements. This technique was recently employed to extend System 7's Multimedia System Services (that is, QuickTime) to support video conferencing.

Figure 8 on page 8 illustrates the primary resource managers that are directly involved in providing distributed multimedia services and therefore require the capability to guarantee quality of service. Figure 8 implies that if Common Transport Semantics guarantees quality of service (and multi-party communications) then the Transport Network and Subnetwork resource managers employed to provide multimedia networking services must do the same. In a distributed systems environment, playing a remote multimedia data object requires that all resource managers involved from end-to-end (from source to destination) must employ an admission policy, reserve resources, schedule their work to meet deadlines, and monitor and report quality of service actually being delivered.

Multimedia Resource Manager Components

The Multimedia resource manager consists of two components, the **Multimedia Configuration Control Point** and **Multimedia Port Services Interface** (see Figure 9 on page 10). The Multimedia Configuration Control Point is a collaboration-aware application. This means it is registered with the Collaboration resource manager and therefore can use its API to request CRM multimedia logical devices to be activated and interconnected through channels (see *Open Blueprint Collaboration Resource Manager* for more details). The prefix CRM distinguishes between logical devices managed by the Collaboration resource manager and those controlled by the operating system's Multimedia System Services (which are in effect encapsulated within the CRM multimedia logical device). A Multimedia Configuration Control Point and a Collaboration resource manager must be installed on systems that are either a multimedia (data) source or destination. The Multimedia Configuration Control Points at the distributed system nodes corresponding to the source and destination multimedia logical devices cooperate as peers along with their respective Collaboration resource managers to activate and deactivate multimedia resources, such as CRM multimedia logical devices and logical connections. In addition, the Collaboration resource manager utilizes its QoS facility to ensure that all logical connections provide the required guaranteed QoS. The Multimedia Port Services Interface is a component that customizes the Collaboration resource manager API for use by applications to control CRM multimedia logical devices, logical connections, data stream flow and synchronization, and information exchange. Its objective is to provide a common multimedia programming interface to achieve application portability in an open distributed systems environment. The communication between applications and logical devices managed by the Collaboration resource manager consists of commands, responses, event notifications and data that are transmitted through logical device ports. The Multimedia Port Services Interface allows an application to send a device control message, data stream control message, or a signal to a port associated with a CRM multimedia logical device.

Once the Multimedia Configuration Control Point has requested the Collaboration resource manager activate a CRM multimedia logical device, the capabilities of the device can be viewed as extending the services of the Collaboration resource manager. Thus, the activation of a logical device representing a local CD-ROM or video camera results in the device being available to all applications participating in a related real-time collaboration.

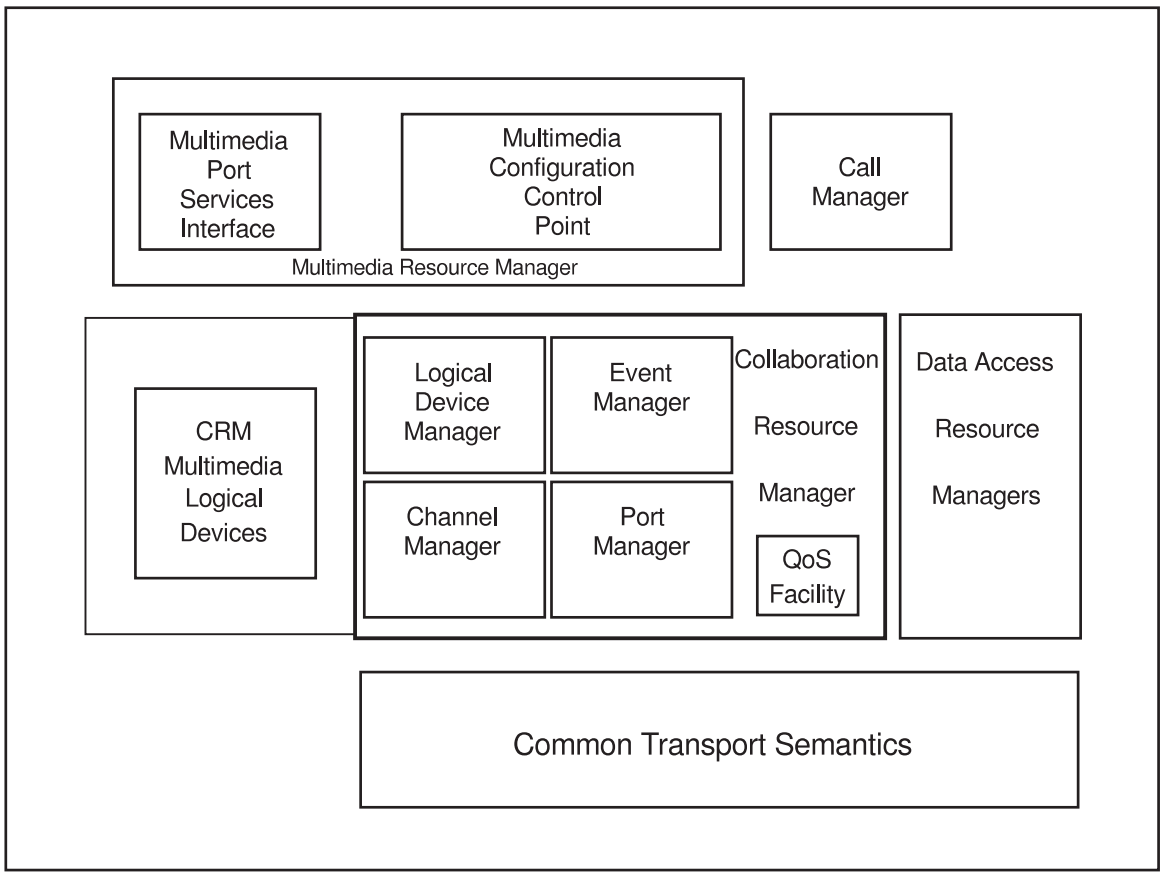

Figure 9. Multimedia Resource Manager Components

There are client/server multimedia applications, for example in interactive TV or multimedia information services, that have requirements that lead to a distribution of function that results in the application software running on a server. An example of this is an application server. This implies that all requests to "play" multimedia content occur on the server and the resulting data stream is "pushed" from the server to the client over a logical connection set up between the CRM source multimedia logical device (on the server) and the CRM destination multimedia logical device (on the client). Figure 10 on page 11 is a simplified portrayal of a client/server multimedia a involving a multimedia application server. In response to a request from the end-user at the client PC, the application on the server requests its Multimedia Configuration Control Point activate the co-located CRM Source MM File Logical Device and in collaboration with the Multimedia Configuration Control Point in the client PC the CRM Destination MM Logical Device. The actual logical device activations are performed by the corresponding Collaboration resource managers. Next, the Multimedia Configuration Control Point in the application server asks its Collaboration resource manager to negotiate the setup of a channel capable of a pre-specified QoS from the source to the destination multimedia logical device. The Call Managers and QoS Facilities in the server and client are involved in connection setup and resource reservation to ensure the required guaranteed QoS through the transport network is achievable. The Multimedia Configuration Control Points then complete the setup by requesting the Collaboration resource managers to connect the CRM multimedia logical devices to the channel. This connection is made by connecting the proper device port to the proper channel port illustrating the coordination needed to actuate data stream communication.

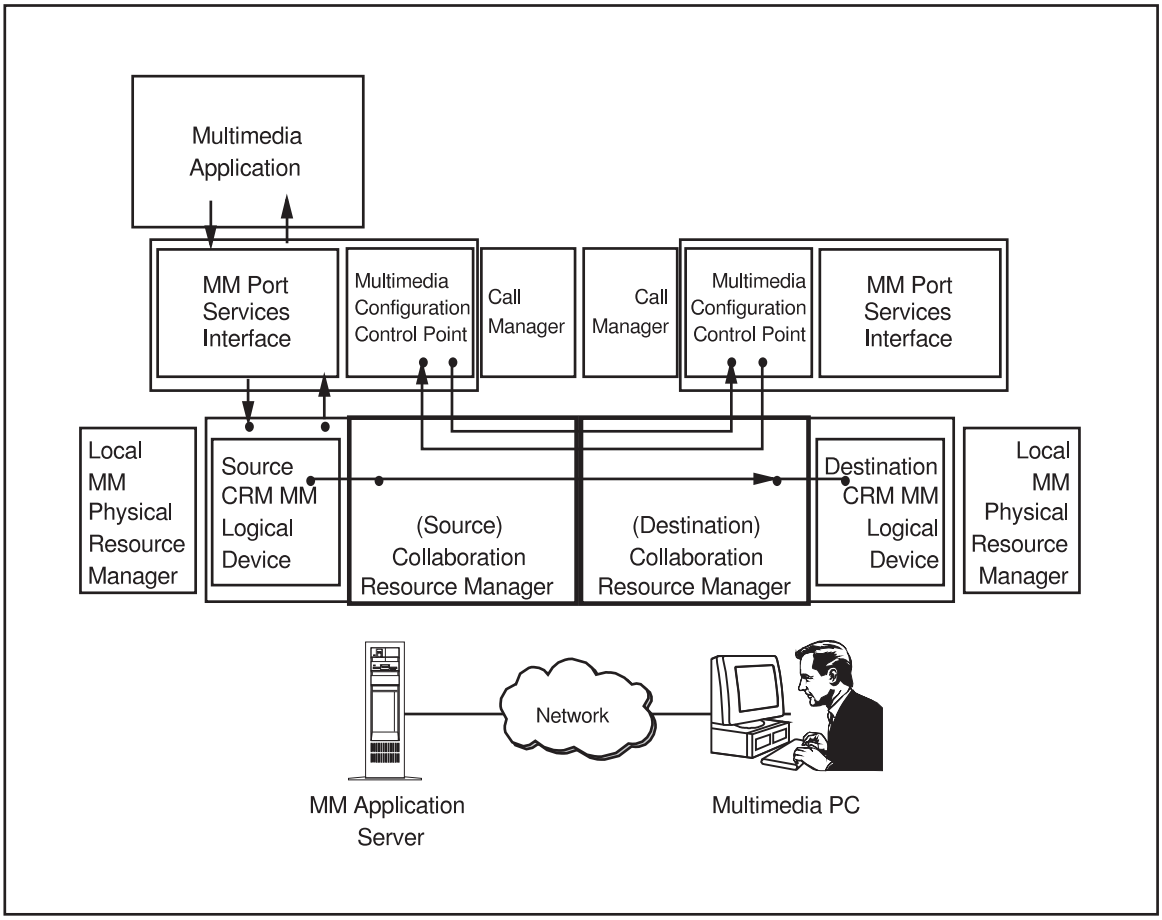

Figure 10. A Multimedia Application Server

The Collaboration Resource Manager Multimedia Logical Device

The Collaboration resource manager models the communication between an application and a logical device as employing device ports to exchange messages representing commands, data, and event signals or notifications. Therefore, the same holds true for multimedia applications which communicate with CRM multimedia logical devices (see Figure 11 on page 12). A CRM multimedia logical device derives the following capabilities by employing both the underlying native operating system's Multimedia System Services and communication services of the Collaboration resource manager:

- Multimedia device control
- Data stream reception and transmission
- Logical timer services
- Data stream flow control and synchronization
- Event detection and notification
- Device and data stream specific support for QoS management
- Logical device systems management support.

As noted before, a major benefit of employing the Collaboration resource manager in implementing distributed multimedia computing, is that a CRM multimedia logical device can employ the collaboration API. This makes it possible, for example, to put an application program in the path from a source device to a destination logical device to dynamically augment the data stream in real-time based on end-user requirements. Home shopping applications delivered through interactive TV might employ this capability to customize the multimedia information that is being presented “on the fly.” The playing of multi-party

electronic games or participation in real-time collaborative training programs over a broadband network will exploit this capability.

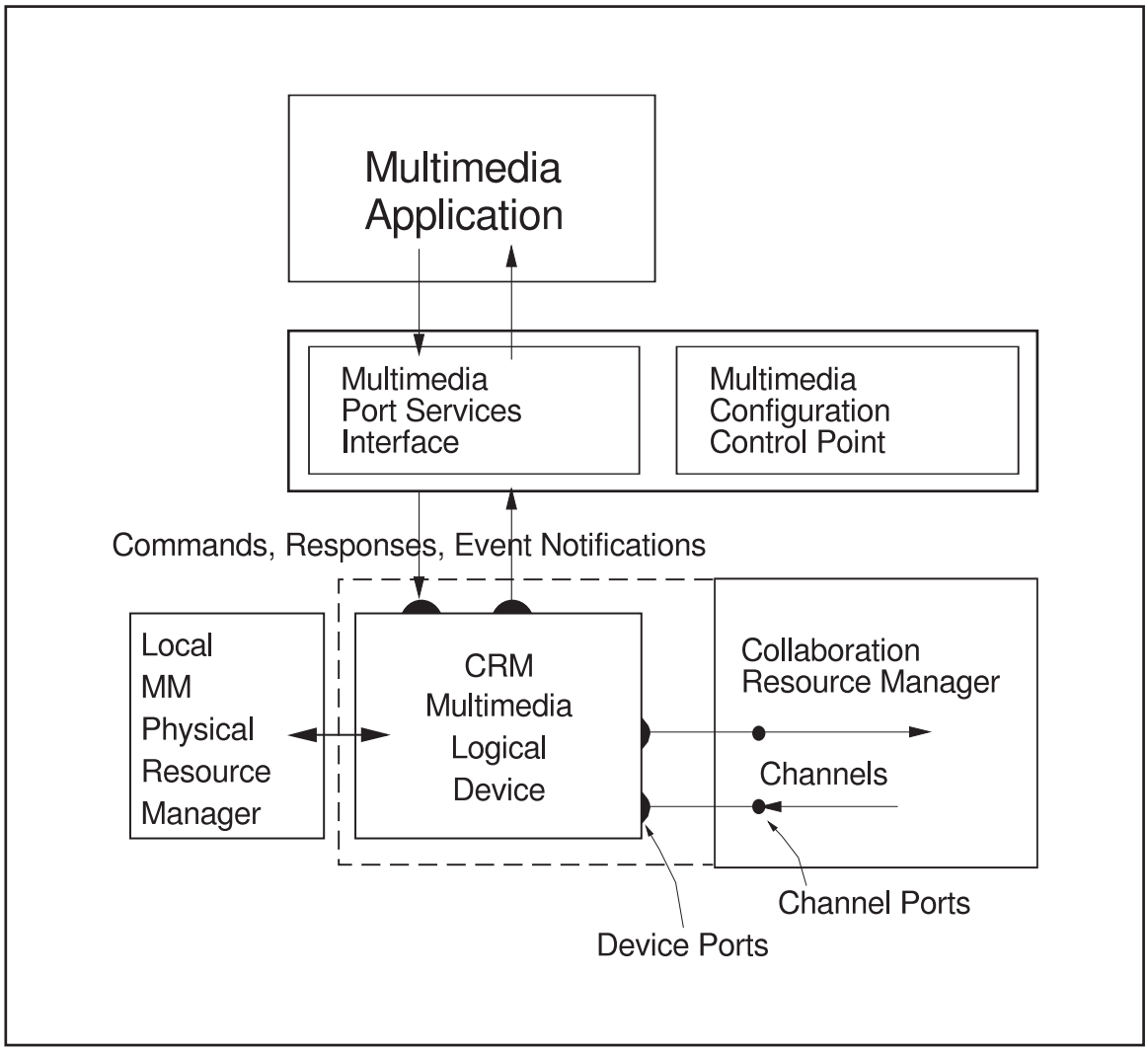

Figure 11. The Collaboration Resource Manager Multimedia Logical Device

The Multimedia Configuration Control Point is responsible for determining the configuration of interconnected CRM multimedia logical devices that need to be activated on behalf of a multimedia application's request(s). In turn, the Multimedia Configuration Control Point employs the Collaboration resource manager to perform the actual logical device activations and interconnect them. The internal structure of the source and destination CRM multimedia logical devices is represented in Figure 12 on page 13. A CRM multimedia logical device masks the details of how the native operating system's Multimedia System Services must be used to control local multimedia devices (resources). A software component is implemented within the CRM multimedia logical device that performs the equivalent of a device driver, albeit a Multimedia Logical Device Driver. The **Multimedia Logical Device Driver** is responsible for communication with the application using the corresponding Multimedia Port Services Interface. This includes a proxy error handler that forwards event notifications arising from the native operating system's Multimedia System Services. The Multimedia Logical Device Driver uses the native Operating System's multimedia programming interface to activate local multimedia resources and control the local multimedia devices, data stream flow, and synchronization. As mentioned previously, the Multimedia System Services assumes a multimedia service involves a pair of native (local) multimedia logical devices, a source and destination, that are statically interconnected through shared buffer memory. We employ this structure and substitute customized multimedia logical devices, the **Proxy Destination**

Multimedia Logical Device and **Proxy Source Multimedia Logical Device**. A proxy logical device, such as the Proxy Destination Multimedia Logical Device, is the local counterpart to the corresponding remote Destination Multimedia Logical Device. A proxy logical device does not perform any multimedia device functions. It simply uses the port communication services available through the Collaboration resource manager API to transmit or receive a data stream over a logical (transport network) connection; that is, it is a data stream transmitter or receiver logical device. The Collaboration resource manager supports the use of passing buffer pointers to exchange data, thereby eliminating the need to move data which is so important in handling streams of multimedia data. The technique of encapsulating the details of the native operating system's Multimedia System Services within a CRM Multimedia Logical Device is a typical practice to provide a common set of distributed services and common API across heterogeneous system platforms.

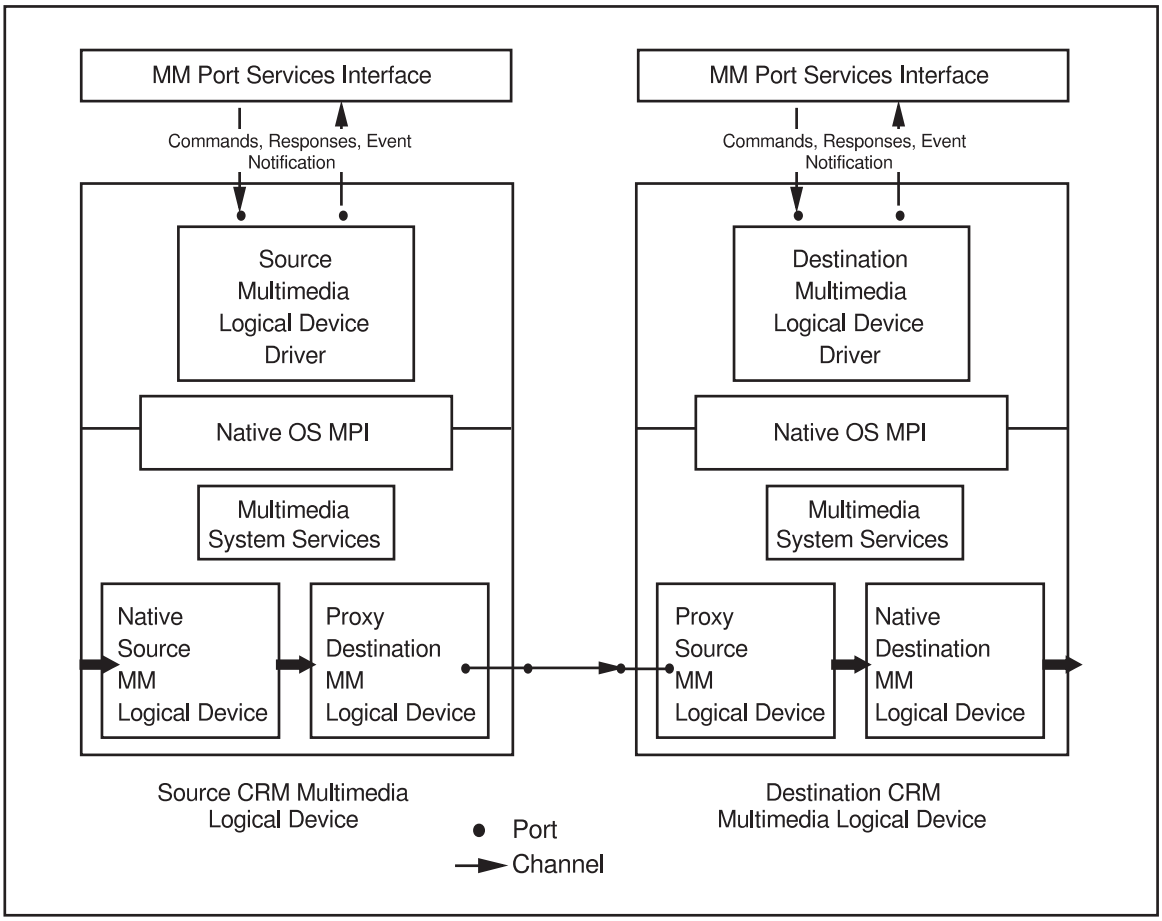

Figure 12. The Internal Structure of Source and Destination CRM Multimedia Logical Devices

Open Distributed Multimedia Computing Migration Considerations

Although it is very desirable to provide a common multimedia programming interface across heterogeneous system platforms, the existing multimedia tools and applications are programmed to use the native operating systems' multimedia programming interfaces. These recent programming investments need to be protected. Thus, to migrate to open distributed multimedia computing the initial support must focus on achieving interoperability and not necessarily demand a common multimedia programming interface. Figure 13 on page 14 illustrates how this can be done. The Multimedia resource manager would consist of the Multimedia Configuration Control Point and an API that provides access to the services of both the Multimedia Configuration Control Point and those of the native operating system's Multimedia System Services to control local multimedia logical devices and data streams. As a result, the

Source/Destination Multimedia Logical Device Driver is unnecessary as a CRM Multimedia Logical Device component. The Multimedia Configuration Control Point is required because it uses the services of the Collaboration resource manager to activate and inter-connect CRM multimedia logical devices. In addition, it provides the interoperability mechanism through peer-to-peer protocols needed to coordinate the management and control of multimedia resources located at the source and destination distributed system nodes. The same technique of employing proxy source/destination multimedia logical devices is used.

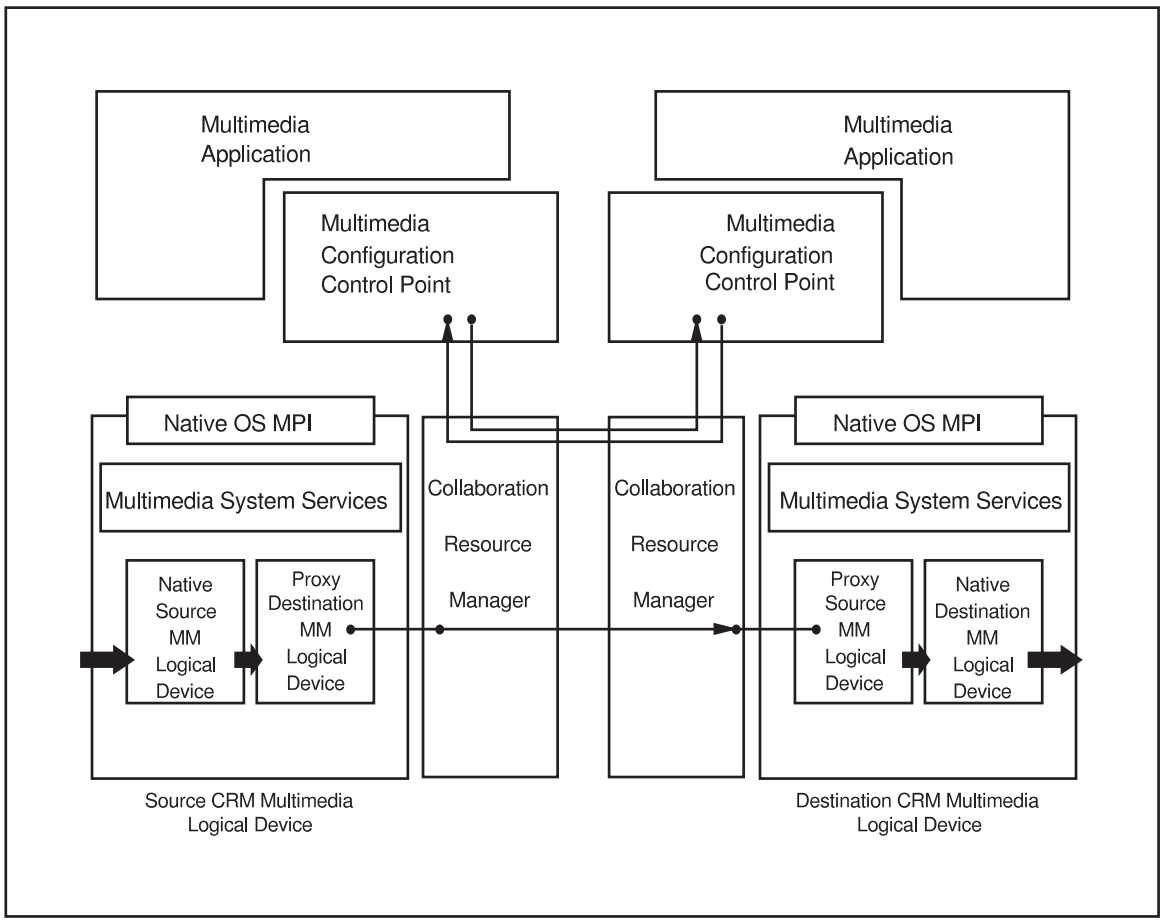

Figure 13. The Internal Structure of the CRM Multimedia Logical Devices to Achieve Interoperability

Summary

Interactive TV applications, digital libraries, multimedia information services and real-time workgroup collaboration have identified the need for distributed multimedia services in an open systems environment. As shown in Figure 7 on page 7, it is rather surprising to see even for only two people collaborating in real-time the complexity of the logical connections. Much more complexity in connectivity requirements materializes when multi-party, interactive multimedia applications are used. This leads to the requirement that the Multimedia resource manager employs both the native operating system's Multimedia System Services to control local multimedia resources, data streams, and synchronization and the Collaboration resource manager to establish and manage the connections between multimedia resources. It is the combination of the Multimedia and Collaboration resource managers that provide the distributed multimedia computing foundation that is required to allow multimedia-enabled systems to play a greater role in communications and information (digital media) management and delivery.

This page is intentionally left blank.

This page is intentionally left blank.