

Media Banks: Entertainment and the Internet

by A. Lippman
R. Kermode

There are two emerging models for delivering high-density, synchronized audiovisual presentations: video-on-demand and the Internet. The first is based on long-lived connections and guaranteed timeliness; the second assumes short spurts of low-bandwidth data on demand. We present the design for a Media Bank that intermediates between them. It provides on-demand access to media elements that are assembled on-the-fly by the recipient to reproduce synchronized audiovisual presentations. The Media Bank uses a fully distributed architecture that assumes a community of viewers. Any member citizen can request or deliver sound, picture, descriptive annotations, or programs to control assembly. Data are redundantly stored in small segments and are cataloged by content and format to facilitate personalized and interactive retrieval.

The notion of a Media Bank first arose at MIT in 1991 during discussions about building a virtual communications research program from diverse activities already in progress throughout the campus. An under-researched area was a “bit-bucket”—a simple repository for all sorts of data where large-scale, synchronous content could be easily deposited and later withdrawn. Video servers were not widely discussed in 1991; however, constructing such a bank would yield potentially valuable insights into new uses for digital video and permit diverse groups across the campus to cooperate.

It was a small step to distribute the functions through a large set of machines and invert the notion of a concentrated video server from a single (or small) set of large machines to a large set of very small ones. But this small step has ramifications. Video is a “dino-

saur” of a data type. Not only is it big and unwieldy, it is notoriously slow to evolve because of the engineering lag and a large installed base. Solutions for digital video processing and storage require new technology and new investment. Economies of scale and construction have given it a history like that of broadcasting, where a few sources synchronously drive a great many, simplified receivers. The very success of the medium for mass entertainment and entrenched commercial interests impedes change and begs the question of why one would care to fix something that is demonstrably not broken.

However, in spite of fifty years of minimal engineering optimization, television is now becoming a digital medium. The sheer increase in program capacity afforded by compression motivates such a change, as does the opportunity for new distribution channels and quality options. Commercial ventures that are more than technologically motivated are in place in North America (DIRECTV** system) and emerging in Europe (Astra digital video broadcasting). The Federal Communication Commission (a United States government agency) inquiry into advanced television services (ATV) is nearing completion and will guide the transition of the UHF (ultra high frequency) broadcast band from sparsely populated analogue telecasts to up to one gigabit per second digital radiation. (It remains to be seen whether that bandwidth will be

©Copyright 1996 by International Business Machines Corporation. Copying in printed form for private use is permitted without payment of royalty provided that (1) each reproduction is done without alteration and (2) the *Journal* reference and IBM copyright notice are included on the first page. The title and abstract, but no other portions, of this paper may be copied or distributed royalty free without further permission by computer-based and other information-service systems. Permission to *republish* any other portion of this paper must be obtained from the Editor.

allocated solely to higher definition services or whether it will be generalized and opened to any commercial use.) International organizations such as the Digital Audio Visual Council (DAVIC) are developing open standards for networked access to television and sound. As the last data type to fall, digital television now flows through the same channels as the rest of the media that we encounter in our daily communicative lives.

At the same time, we are witnessing the meteoric rise of the Internet as a *de facto* global infrastructure for both personal and commercial information. This network was originally envisioned as reliable support for distributed computing (1969). By the mid-1970s, an Advanced Research Projects Agency (ARPA) study found that three-quarters of all ARPANET (ARPA network) traffic was electronic mail (e-mail), and in the 1980s the network became a locus of commercial and personal interactions. By combining the network with consumer use of the personal computer, the network has now become an entity in itself ("the Internet"). In addition to becoming a foundation for global information infrastructures targeting commerce, research, education, and medicine, the Internet has become a democratized, participatory society. Its growth demonstrates the power of *community* versus *interactivity*; participation eclipses transaction-based full-service networks and asymmetric client-server designs.

In contrast to video service, the Internet is inherently symmetric. Any computer connected to it has the same potential access and abilities as any other—there is no systematic bias or barrier built into it. Many researchers attribute its popular growth to this simple feature. Equality of access is the basis for international communities, newsgroups (forums for discussion and information postings), chat rooms, and the plenitude of home pages (user-owned and -maintained nodes for public access of personal, commercial, and reference data) on the World Wide Web (a portion of the Internet where data exist in an *ad hoc* standard format). Whereas library retrieval and interactive information and transactional services have been slow to diffuse through society, the community aspects of the Internet and the reduced cost of entry now accelerate its use.

This impels us to consider mechanisms for media distribution that do not emulate broadcast. In particular, we address extending the ethos of the Internet and its hyper-library nephew, the World Wide Web,¹ to accommodate high bandwidth, real-time, synchro-

nized data such as audio and video. In effect, we build a bridge between the fully distributed nature of the Internet and the requirements of audio and video media (see Figure 1).

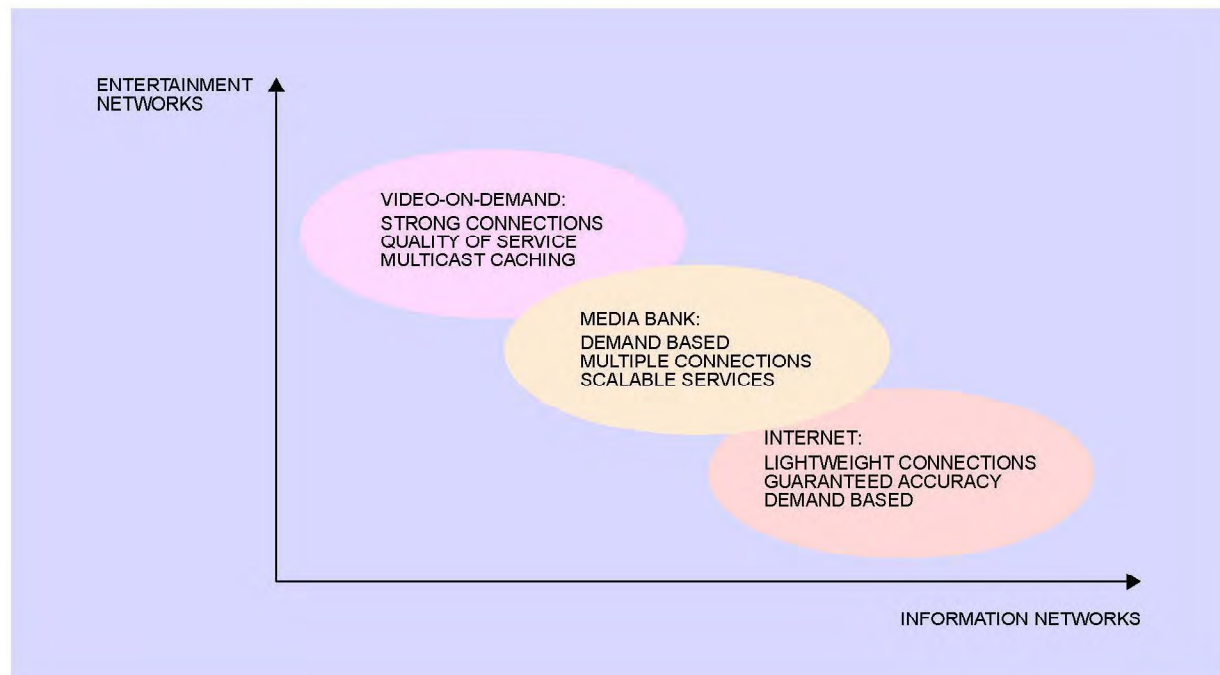
The primary design point is the need for synchronous delivery. Commercial data can be lost in transit with no ill effect—the datagram is simply retransmitted milliseconds later. Accuracy is more important than timeliness (at least at small-enough scale). Conversely, the denouement of a narrative cannot suffer delay. It can be distorted or degraded by transmission errors, but the definition of video is in its continuous flow. Maintaining continuity from distributed resources in the face of errors, sporadic system availability, and variable demand preoccupies designers of video servers.

The second fundamental challenge is that the video of today is inherently formatted information. Unlike text, which can consist of individual characters whose pagination, typeface, and layout are often packaged separately from the message, a video sequence embeds the data in a predefined raster size, frame rate, aspect ratio, and color space. The demands of these parameters ripple throughout the system, placing constraints on channels, storage media, processing hardware, and display technology. This mitigates strongly against content dispersion and reinforces a necessity for concentration of information.

In 1996, these concerns can no longer be considered in a vacuum. Video service of all sorts has broken the surface in discussions of national and global infrastructures and as a new consumer entertainment opportunity. *Video dialtone*, the local telephone entry into television distribution, is being tested in all regions of the United States and many European countries. There are at least 57 small- and medium-scale advanced service tests throughout America, and all seven regional operating companies have filed construction permits with the Federal Communication Commission to install broadband networks. Salem, Massachusetts, site of Alexander Graham Bell's first long-distance telephone call (from the Lyceum auditorium to *The Boston Globe* newspaper office) is poised to be an early test site for long-distance video service. At least two digital direct broadcast satellite systems are commercially deployed; both are showing early signs of success.

Other researchers have embarked on similar paths. In particular, Rowe et al.²⁻⁴ have built distributed video

Figure 1 Comparison of differing network requirements



servers for on-demand access to large audiovisual archives. Their goal is scalable storage of bulk audiovisual data such as movies and classroom lectures. Data in their system are represented in a variety of formats with annotations and reference information included in the design. A "Compound Media Object" model allows flexible assembly of component information into coherent presentations. Synchronization is shared between the servers and the clients.

The design ethos used for building the MIT Media Bank is similar. It is driven by four main ideas:

- Cross network operation
- Seamless distribution of services
- Stored descriptions
- Multiple format primitives

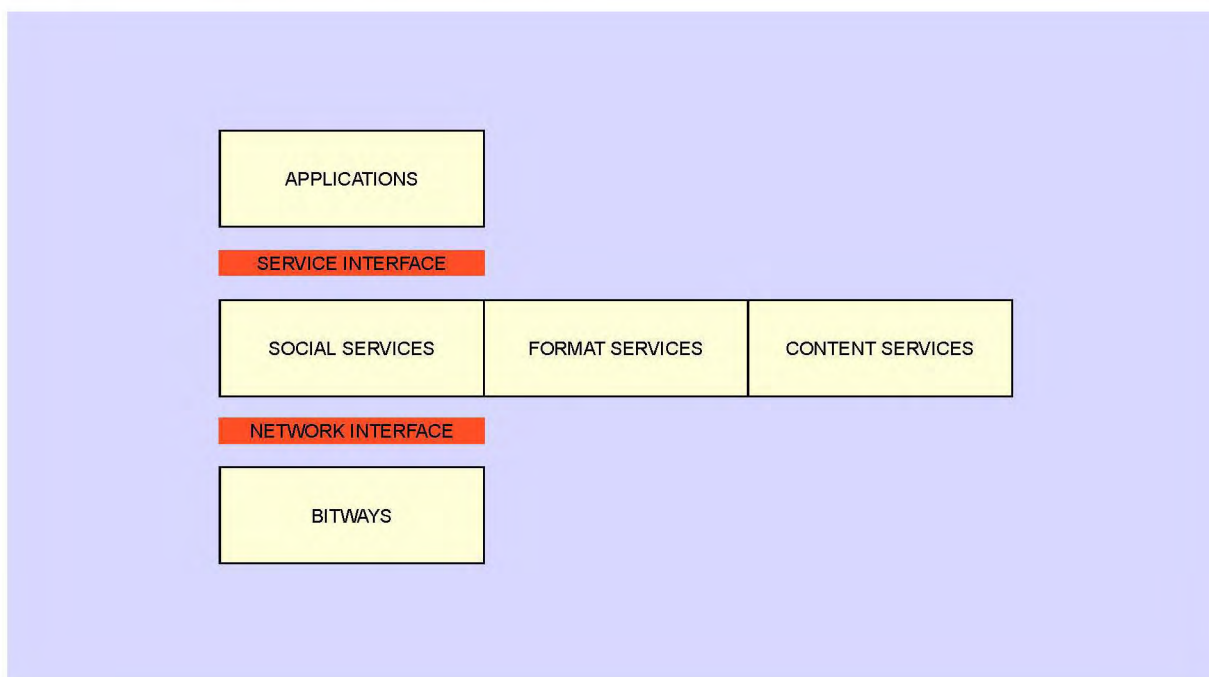
Entertainment program assembly illustrates many of these principles. Elements of a movie consist of a separately stored series of images, sounds, and descriptive data. These data may be contained in a single storage system or be replicated on several. Also, the data may exist in a diverse set of formats, for exam-

ple, JPEG (Joint Photographic Experts Group),⁵ MPEG (Moving Pictures Experts Group),^{6,7} AC-3 (audio), and other formats. A separately stored, list-based structure contains the information needed to assemble the diverse parts into a coherent presentation and maintains a directory that locates the original material.

For example, in the case of a movie, the editing instructions to assemble diverse versions of the plot (X-rated; PG, or parental guidance required; 30-minute showing; etc.) may exist at one site but the images may be stored separately from language-dependent sound tracks. The instructions for billing and authentication of the viewer may reside elsewhere. The movie becomes a coherent event when it is requested—its display draws upon resources as needed. Just like the leaves on trees that "make no sound" if they fall in an unoccupied forest, this movie literally does not exist unless someone demands a viewing.

Seamless distribution of services implies that any of the processing functions described can occur in any of the systems attached to the network of the Media

Figure 2 The Media Bank multilayered network structure



Bank. Many may occur in a hypothetical digital television receiver in the home itself, but they could as well be done by authentication servers, editing systems, dubbing servers, storage servers, and the like. Note that this model for scalable storage and distributed processing extends familiar computing notions to entertainment—the World Wide Web does the same thing for paginated text. We argue that we can provide most services and functions by carefully crafting the data formats and basic elements to suit existing storage and processing models, rather than the other way around. A unique storage architecture for time-sensitive data may not be mandatory.

We adopt the style of many emerging national and global information infrastructure (NII/GII) efforts: a multilayered network structure that simplifies *applications*, by providing generic and specific *services* that are delivered through multiple *bitways* (see Figure 2). This model implies open interfaces between the layers, and stresses the generic intermediate or *services* layer. For the special case of media service, we expand that layer into three components, called respectively, social services, format services, and content services. Each has different characteristics.

Social services include billing, authentication, and any other information services (lexicographical, e-mail, etc.) that are normally associated with networked computing.

Format services are operations performed on the data on behalf of any client—for example, translation from MPEG to another encoded representation, stereo to monaural conversion, or recording of any kind. These are operations that occur on the interface to access the data and facilitate use in a diverse environment.

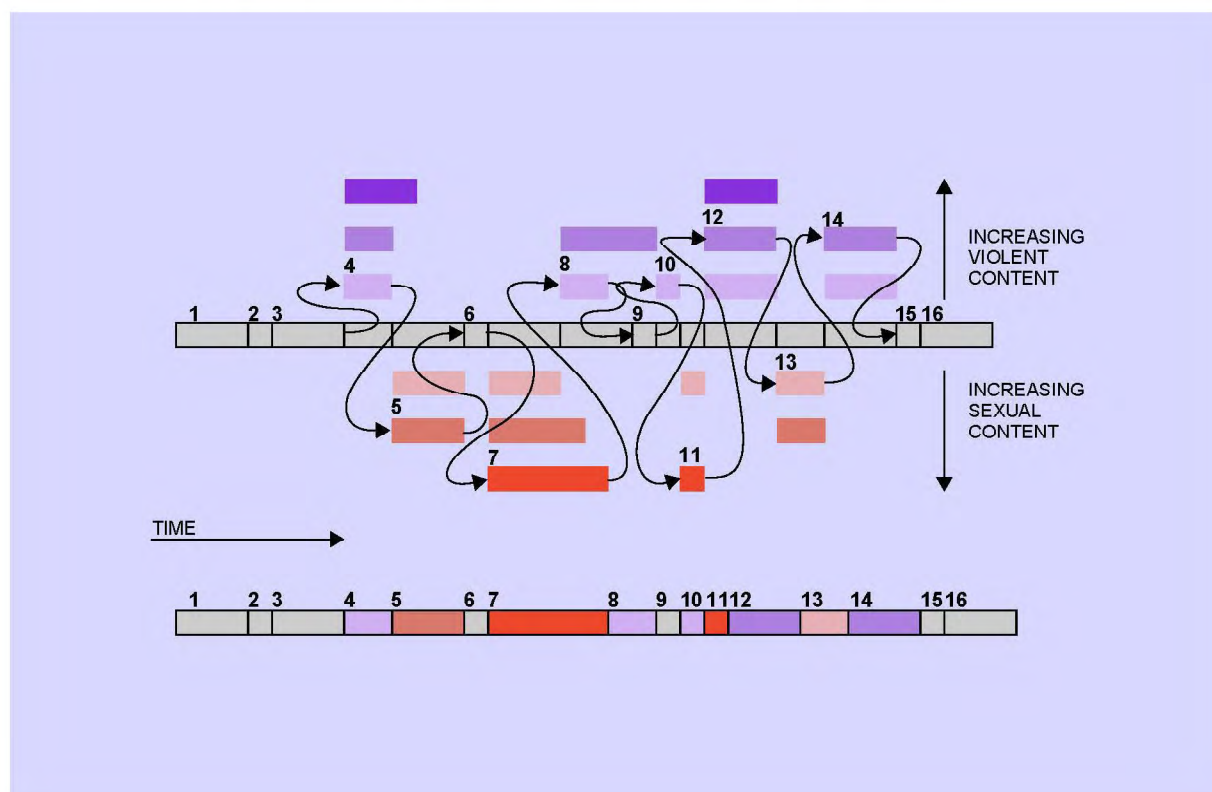
Content services are distributed processing tasks that relate to annotations and assembly rules.

There are only three elements needed to develop a Media Bank:

- An indexing and storage scheme
- A suite of client programs to access data
- A delivery protocol

In essence, the indexing system is optimized for representation of audiovisual elements stored in elemental or byte-sized chunks. These are called the Media

Figure 3 Personalized movie constructed from scenes of varying violent and sexual content



Bank protocols, and they are overlaid onto a suite of access and delivery mechanisms that are themselves optimized for networked delivery of randomly sized and located elements. Scalable and layered video are prime examples. Part of this component of the protocol provides for file access into variable sized arrays, needed, for example, to access selected frames of a coded video representation.

Access programs are the applications. These provide means to peruse the information distributed through the bank and to simulate television receivers. A growing repertoire of such programs is available to summarize information, transform it between coded formats, merge text and visual data, and incorporate authentication and payment mechanisms.

The delivery protocol is divided into two domains: networked distribution of synchronous information and specific protocols for access to *meta-data*⁸ about the data.

Indexing and data representation

Choosing an efficient, malleable, and intuitive way to index and represent data is one of the most important design tasks of the Media Bank. Without an appropriate representation, indexing becomes slow, and delivering data from multiple isochronous sources to a client becomes impossible.

The representation chosen separates media objects into two parts. The first part consists of meta-data about the object. For example, the meta-data for a video clip object would describe the length of the clip, its resolution, its compression format, and information needed for indexing and retrieval. The meta-data could also include a function that describes a browser for viewing the video. The actual data for the video clip are not stored with the meta-data. Instead, they are stored in the second part, and referenced from the meta-data part. Several key benefits are afforded by segregating an object's meta-data and data.

The first benefit is that one need not retrieve the entire object to learn its characteristics, one need only retrieve the lightweight meta-data. Thus, one avoids the overhead associated with retrieving the entire object for the purposes of determining whether or not it is what is wanted, only to discard it upon finding that it is not. In other words, the overhead of filtering on the basis of content is greatly reduced.

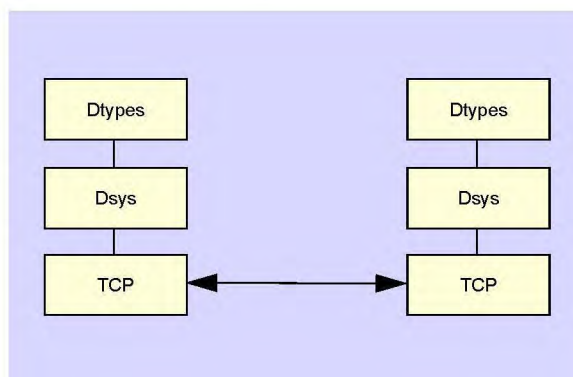
The second benefit arises out of the ability to derive objects from other objects. Consider the end result in the production of a movie or a television documentary. Typically hundreds to thousands of feet of film are logged and whittled down into a finely tuned, highly crafted linear result. The content of the movie becomes fixed and opaque. One cannot examine scenes from a different camera angle, tone down violent content, or request the 16:9 aspect ratio formatted version. Nor can one repurpose the material without having access to the material itself. Having the ability to derive objects from other objects removes all of these restrictions. Now, instead of the movie being viewed as a fixed single product, it may be assembled at the time of viewing. The decision of which particular clip and which parts of the clips are used becomes a function of the viewer's own preferences (see Figure 3).

The third benefit is really a corollary of the second taken from the content provider's point of view instead of that of the consumer.

Content providers have major concerns regarding access to their products. Clearly, controlling access to the bits of an object is important because it controls consumption of the object and hence enables the content provider to charge for the privilege of access. On the other hand the content provider also wants consumers to consume and to reuse the bits as often as possible, because greater use and reuse leads to greater revenue. Unfortunately, these conflicting goals are ill-served by the current broadcast model employed by practically all content providers today. As soon as a newspaper leaves the press or a television show is aired, access to the original material and editing decisions is lost and the material is rendered opaque for repurposing.

In our model one need not have access to the original footage in order to derive new content. Instead, all one need do is to derive a new object by combining select pieces of the proprietary one that remains secure on the content provider's own server. The end result is a

Figure 4 Layered transportation stack for Media Bank object delivery



new object that is an *edit decision list*, an object that can be exchanged privately without breaking copyright laws or losing functionality. Should a client wish to use the new object, the client still has to access the content provider's secure servers for the object data, and pay the appropriate royalty or access fee for the privilege.

Media Bank objects are defined and distributed via a layered set of protocols (see Figure 4). At the lowest layer, delivery of meta-data is via Transmission Control Protocol (TCP). Above TCP is a protocol for the distribution of objects where the object type is carried along with the data, called *Dsys*.⁹ *Dsys* allows diverse objects to be distributed, without advance knowledge at the receiver about the specific storage formats or requirements of each data element. Data for this protocol are a set of list-processing-like (LISP-like, actually *Scheme*¹⁰) vectors called *Dtypes*.¹¹ *Dtypes* include basic elements such as numbers and text arrays as well as extensions to define special-purpose types such as encoded bit streams.

Media Bank objects are accessed, located, and manipulated by a set of Scheme functions interpreted both at the point of origination (the server) and at the receiver.⁹ This generalization allows all Media Bank objects to be both data elements as well as programs, and permits them to be interpreted and executed both locally and by remote agents. In operation, this means that a viewer for any object can be a program specific to that object, with user controls carried along with the data. Objects can therefore be assembled to provide a different television receiver or user interface for each movie, or could be organized into networked

Figure 5 Set of three objects for the trailer to the movie *Bad Boys*

```
#((uniqueid . "<badboys-video-mjpg>")
(name . "badboys-video"
(type . (item image mjpg))
(version . 8166411078)
(owner . "Henry Holtzman")
(hardware-formats . #(mme))
(data . "x-mbd://bad-taste.media.mit.edu:41503/raid4/mbdata/trailers/badboys/badboys.movie")
(frames . 3277)
(width . 320)
(height . 240)
(fps . 24))

#((uniqueid . "<badboys-audio-raw>")
(name . "bad-boys-audio")
(type . (item audio raw))
(version . 8166411078)
(owner . "Henry Holtzman")
(data . "x-mbd://rocky-horror.media.mit.edu:41503/mbdata/trailers/badboys/badboys.audio.raw")
(duration . 24100476/176400))

#((uniqueid . "<badboys-movie>")
(name . "badboys-movie")
(type . (item application playlist))
(version . 8166411078)
(owner . "Henry Holtzman")
(script .
#(
(play-multiple
#(
#(
#(play #((object . "<badboys-video>")))))
#(
#(play #((object . "<badboys-audio>"))))))))
```

games. Further, the execution of the code associated with any object can be delegated, facilitating automatic translation of data types, accounting, and security.

An example of a set of three simple Media Bank objects for the trailer to the movie *Bad Boys* is shown in Figure 5. Notice that the first two objects contain the meta-data for video and the audio to the movie, respectively, while the third object indicates that the two are to play synchronously to form a single presentation. It should be noted that these objects contain the

base minimum in terms of information for their retrieval and playback. Extra name value pairs can be added at will (with the exception of a few reserved names), so that the description about the object is free to grow over time.

It should be noted that fixing the location of the object data in the data field of the meta-data for the object does not result in the same problems of brittleness as for the Internet World Wide Web Uniform Resource Locators (URLs). The reasons for this will become clear upon reading the section on directory servers.

Data delivery protocols

Central to the design of the data delivery protocols¹² is the notion of pull-based delivery. Pull-based delivery can be defined as “data delivery only in response to a request.” While this delivery method is currently used by the World Wide Web for text and still images, it has only recently emerged as a contender for the delivery of large data types such as audio and video. Traditionally these data types have been delivered in an asymmetric broadcast fashion from a central head end that pushes data onto the network regardless of whether a client can handle the data or not. By employing the pull-based paradigm, one breaks nearly all of the usual assumptions made regarding the delivery of digital video and audio. Instead of a distribution mechanism that broadcasts from a centralized location to low-capacity clients, uses bandwidth asymmetrically, and puts the synchronization control at the server, the Media Bank uses an architecture in which

- The data are fully distributed across many servers in different locations.
- Multiple instances of the same objects can be duplicated at different servers.
- Bandwidth is considered to be used in a symmetric fashion.
- The clients are endowed with a degree of intelligence to perform a range of tasks that were previously the sole purvey of the server.

Two main types of servers in the Media Bank provide the bulk of the functionality listed above. The first kind of server is the directory server that provides a directory service for finding out where an object is located. The second kind is the object server on which the actual data are stored. Between them the object and directory servers provide access to objects stored in the Media Bank in a seamless fashion that:

- Is fault tolerant
- Allows both object and directory servers to be added and removed from the Media Bank at will
- Manages the load of both computation and network usage across the servers of the Media Bank and the network links that connect them

Directory servers

Directory servers provide the means for locating objects stored in object servers in a manner akin to that used by the UNIX** Domain Name Service for resolving machine names to Internet Protocol (IP)

Figure 6 Example of a network unique identifier (NUID)

```
#((uniqueid . "badboys-video-mjpg")  
  (host . "rocky-horror.media.mit.edu")  
  (port . 41501))
```

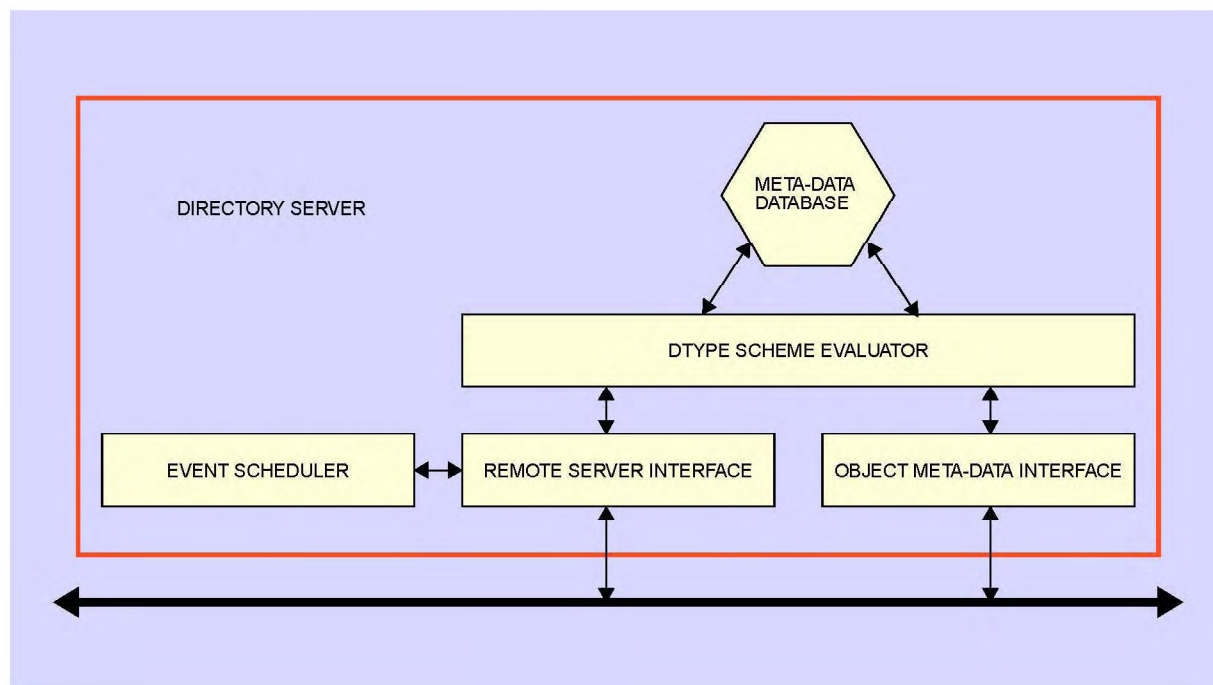
addresses. The idea of directory service is not new,^{13,14} and is the subject of ongoing research particular amongst the Internet World Wide Web community. This section describes our implementation of a directory service.

Objects in the Media Bank are identified by one of two mechanisms: names or unique identifiers (UIDs). The difference between the two is that there may be multiple objects with the same name in the Media Bank but only one object instance per unique identifier. Object equivalence is assigned on the basis of name. In addition, unique identifiers come in two flavors: simple and network. A simple unique identifier assumes that the host and port of the object server is already known and that the unique identifier identifies a single object stored on that particular object server. A network unique identifier (NUID) differs from a unique identifier (UID) in that it includes the host and port number of the object server with the simple UID to form an identifier that is unique to the entire Internet, not just a particular machine. An example of a NUID for the object with the name “badboys-video” is shown in Figure 6.

Apart from the abstraction of names and unique identifiers, Media Bank directory servers provide a number of additional services. Specifically, they provide:

- The ability to detect dead or unresponsive object servers and remove references to them when responding to requests to resolve object names

Figure 7 Directory server functional block diagram



- The capability to store a subset of the meta-data information about an object that may be used by the client to prune unnecessary accesses to object servers for an object that does not fully meet the preferences specified by the client
- The facilities to manage the computational load between servers capable of serving equivalent objects

The inclusion of these three additional services, along with the naming abstraction scheme are key features of the Media Bank as they solve the problem of brittle URLs that is inherent with the current implementation of the World Wide Web. No longer will the going down of one server automatically result in catastrophic blocking of a request for an object, since the network unique identifier of another equivalent object will be returned, should one be available.

The implementation of these services is reasonably straightforward. Each directory server consists of five main functional units, as shown in Figure 7.

At the core of the directory server is the Dtype Scheme evaluator. Built upon the Dtype data manipu-

lation and Dsys networking libraries, the Scheme evaluator executes Scheme code applets for the other four functional blocks. It should be noted here that the Dsys networking library (through which the remote server interface and object meta-data interface connect to the network) uses a connectionless protocol on top of a modified TCP/IP stack in which connection establishment timeouts have been shortened to fractions of a second. While in general TCP/IP may not have the best performance in terms of latency, for local networks its performance has been found to be acceptable for real-time access. Work is currently in progress to replace the TCP/IP portion of the Dsys library with a real-time networking protocol that uses multicasting, distributed caching, and Application Level Framing (ALF).¹⁵ This new protocol may also use RSVP¹⁶ and eventually be used in an IPv6¹⁷ environment.

Object servers communicate with the directory server via the remote server interface. Through this interface they inform the directory server of their presence, their current state, and make requests for directory service via regular remote procedure calls (RPCs). Should the directory server accede to a request by an