6 The System Components

Introduction

A digital asset management system uses a suite of applications linked together to provide a seamless experience to the user. It is not just a piece of shrink-wrap software that can be used straight out of the box. Rather it is a collection of many components that have been carefully selected to meet the needs and budgets of the enterprise. The system should also have links to existing software applications: accounts, customer records, and planning systems. To maximize the operational benefits, it is advantageous for these applications to exchange data with the digital asset management. Like any enterprise-wide deployment of software, much of the installation costs will be for the professional services. These are usually required to merge the many stand-alone applications into a seamless operating environment, and to customize the business logic to suit the needs of the customer.

This list details some of the components in a typical installation:

- Content repository
- Digital asset management application
- Databases
- Search engine
- Indexing workstations
- Rights management application
- Web portal

The complete digital asset management system is rarely the product of a single vendor. Most vendors will partner with specialists for services like the search engine, video logging, speech recognition, and the underlying database management system (DBMS). The core digital asset management acts like glue. It binds all these functions together using the business rules. The users see a unitary view of all the systems assembled at the presentation layer.

This modular approach allows the best-of-breed solutions to be used, and should retain flexibility for the customer to stick with their favored databases and operating platforms. Clearly, with all these different products, a system's integrator will be required to act as a prime contractor to ensure that the final solution operates satisfactorily as a whole and meets the original agreed requirements.

I shall now take a number of views of the system architecture. One view is the software system architecture, usually a multitiered system, with presentation, business, and data layers. Another view is content-centric, with the system in shells around the core assets, modifying the view the user sees.

One of the key functions of any digital asset management is to relieve the users of the need to perform repetitive tasks. This frees them to concentrate on the more creative side of media creation and publishing. The design of the user interface is the key to the success of the product. The other issue of great importance to the user is the way that the system fits with the existing corporate processes. If too many changes are made to the ways of working, then the users will feel alienated from the asset management. Limited scale trials are very useful for the shakedown of workflow issues. The goal is for the users to feel excited about the opportunities that the system presents, not to resent what is viewed as an imposed system that is designed to reduce manning levels. The goal is a win–win for the corporate management and the knowledge workers.

Mainframes to Multitier

Looking back through the history of enterprise computing (Figure 6.1), it all started with the mainframe and simple clients based upon the video terminal or visual display unit (VDU). Smaller mainframes (the minicomputer) were developed to suit the needs of medium-size businesses, and then IBM launched the personal computer (PC). This allowed even the smallest business to produce professional-looking correspondence and to run accounts software.

As the power of the PC increased, it became feasible to run graphics-rich programs. Although the PCs had clunky text interfaces with box-like graphics, the early Apples pioneered the way for more fluid interfaces using mouse-driven windows. These led to the WYSIWYG (what you see is what you get) user interface that we all demand today.

As the costs of the desktop PC dropped to little more than the desks they sat on, business users demanded the same user interface as the PC. A good example is the word processor. The first products used the mainframe computer to run the application. To see the final layout, you had to make a printout. To produce a complex layout was an iterative process involving many intermediate printouts. The WYSIWYG word processor application did away with this, dramatically improving productivity.

After the replacement of the video terminals with PCs, the minicomputer that ran the word processing became redundant. A central file server replaced it.

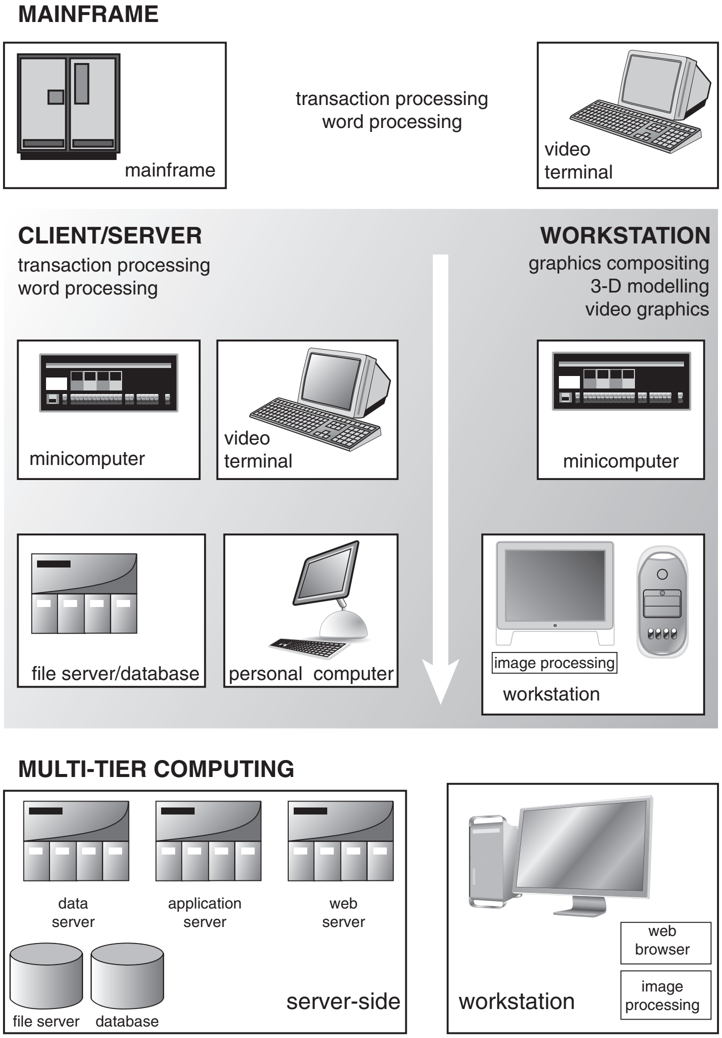

Figure 6.1 The evolution of multitier computing

Client–server

This new architecture was the client–server configuration (Figure 6.2). The user now had the power of the new office application suites. This combination of word processing, spreadsheets, and basic drawing facilities meets most of the needs of the average office worker. Alongside this, the old mainframe systems still ran the company databases for online transaction processing, the sales order processing, inventory control, and manufacturing resource planning—everything that makes the enterprise tick.

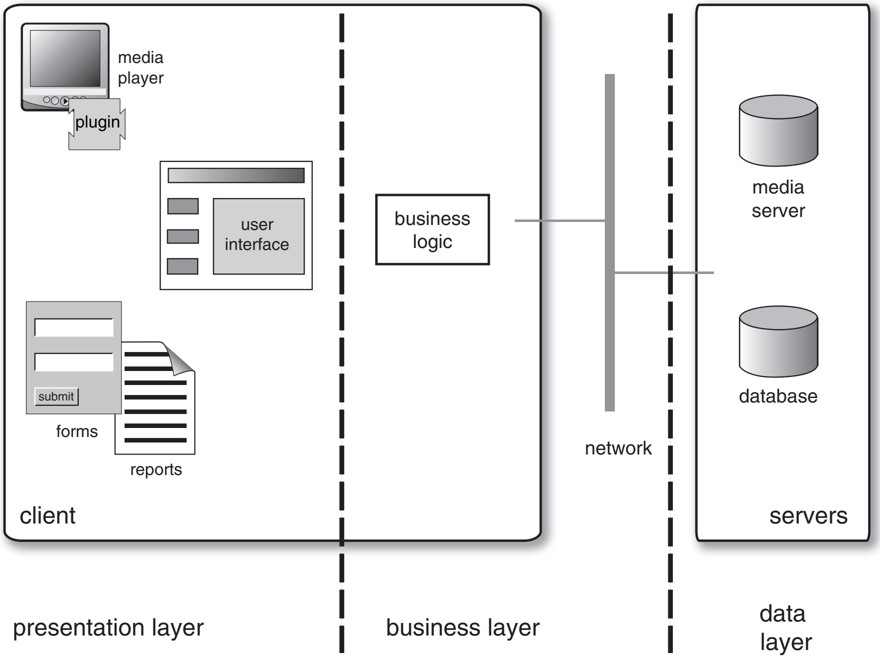

Figure 6.2 The client-server model

Some asset-management products adopted the traditional client–server approach. The client includes the business logic and a presentation layer for the user interface. The server is essentially just a relational database. The clients interact with each other through the DBMS using stored procedures and triggers. Client–server is fine for small systems but, in practice, it does not scale well beyond around 100 clients.

Initially everybody was happy with the wonders of modern computers throughout the business. However, it soon became apparent that running thousands of PCs with thousands of instances of the applications was very expensive. First there were the costs for the per-seat licensing. Then there were the costs involved in updating machines when new software releases came out (all too frequently). But the real killer was the ease with which the average office worker could reduce his or her own computer to a state of malfunction, either by misconfiguring the applications, or by illicitly loading unofficial software. The machine would grind to a halt, ready for rescue by the information technology (IT) department.

The cost of all this downtime, and the staffing levels required in IT, began to be of great concern to senior management. There had to be better ways.

One approach was to use the "network" PC. This had no removable storage—no CD or floppy drives—so it was difficult to load unauthorized software. Modern operating systems allow remote access by a system administrator, so that software updates could be performed from a central point over the corporate network. This is great in a controlled world, but what about the marketing communications department? How do they exchange files with external designers? How do staff in personnel make backups of confidential information? Most enterprises accept that distributed applications are not going away. The network PC has a place, but it is not the answer for the typical distributed-asset management application.

IT professionals have taken two paths to counteract these problems. One is to retain the advantages of the client–server environment. The solution is to use a desktop management system. This allows the IT managers to have complete control over software distribution to the clients. The operating system access is configured to forbid local software installation and configuration. Applications, patches, and updates are all loaded from a central point. The systems also offer remote troubleshooting, and can usually self-heal local software faults. Such a system removes most of the problematic maintenance issues that have dogged client–server and distributed installations.

The Thin Client

The other route is the thin client. This is a return to the simple terminal of the old mainframes. Today's user has many expectations from a user interface that the VT100 style terminals could not offer. Perhaps the most important are support for graphics and mouse-driven user interaction.

The current implementation of a thin client is to use a web browser on a standard PC. One option is to add additional functionality to the web environment by adding a Java virtual machine. This supports applets to run small programs on the client. The business logic is then transferred from the client to a third tier, the application server.

The two paths are best suited to different applications. The client–server is ideal for office applications, where the work is file based. The user loads a file from a central server, edits it locally, and then saves back to the server. The user has all the advantages of a powerful application running locally. Imagine the latencies of using a spell checker while you type, if the word processor were running on a remote server.

Data-record-oriented applications are more suited to the central data server. Information is viewed and updated from the client, but the data resides on the central server.

Digital asset management has elements of both. There is much data processing, relating to metadata queries and updates. There is also file processing, the creative processes applied to the content: editing, format conversion, and compositing.

The outcome is a hybrid approach. Many of the productivity tools run locally on the client machines. They can be centrally maintained using a desktop-management system. The craft-oriented applications, like the nonlinear editor, may still run as a stand-alone application, maintained locally at the workstation. The database-intensive applications, with workflow management and peer-to-peer collaboration, can use a web browser to view a presentation layer in the application server.

Three Tier

The complex business logic of asset management leads naturally to a central application server. This can support the collaborative workflow and communications between the client workstations with central business logic to manage communication between the clients. This logic forms a third layer between the database and the client (Figure 6.3).

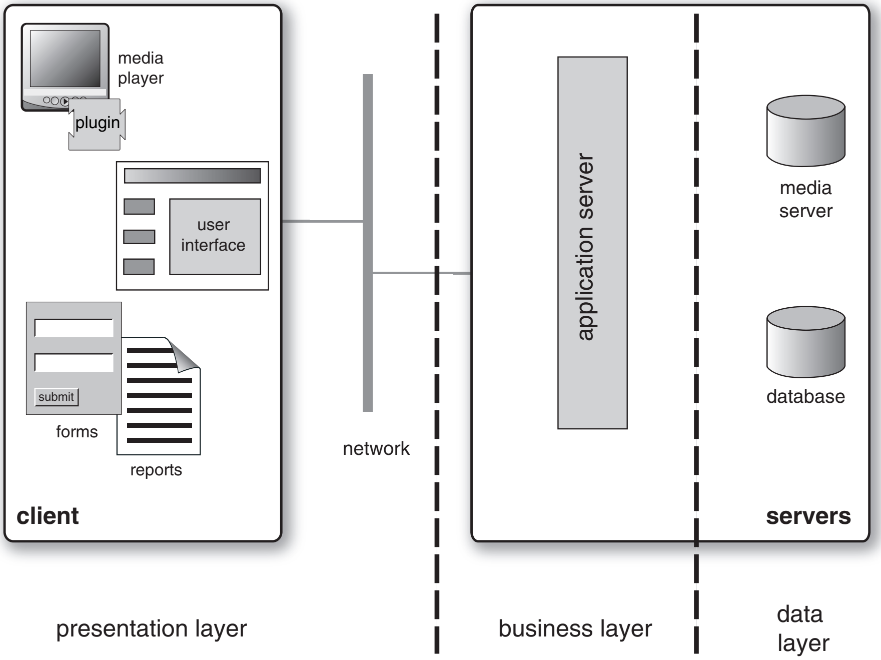

Figure 6.3 The three-tier model

Many enterprises planning to purchase digital asset management will want to minimize the cost of each client. Many users only want to search, and then preview files. Such an application does not warrant the cost of a fully featured client running a local application. The enterprise may well want to share asset information with suppliers and customers.

The requirement to minimize cost dictates a thin client, especially for remote users. Suppliers of creative services may well use Apple Macs. Other partners may use Linux.

Your own enterprise may use Windows for office applications and UNIX for transaction processing. All this demands a client that is agnostic to operating system, and that can use widely available communications protocols. Well, there is one such client—the web browser.

Rather than running the two tiers of the client–server model, the web browser requires at least three tiers with a web server generating the presentation.

Multitier

The multiple or *n*-tiered approach to computing splits an application into manageable chunks (Figure 6.4). Typically, there is a data layer, a presentation layer, and a layer of business logic linking the two. This has many advantages for the IT department. The data layer can be a standard relational database:

DB2, Oracle, or SQL Server, it does not matter. These are mature products, with predictable behavior and formal maintenance procedures. The presentation layer is just a web server, again no great problem to operate and maintain. With this architecture the digital asset management can be broken away into a separate box called the business logic. Here the rules for manipulating the data and the connections to third-party applications can all reside on a separate application server that runs the core asset-management software.

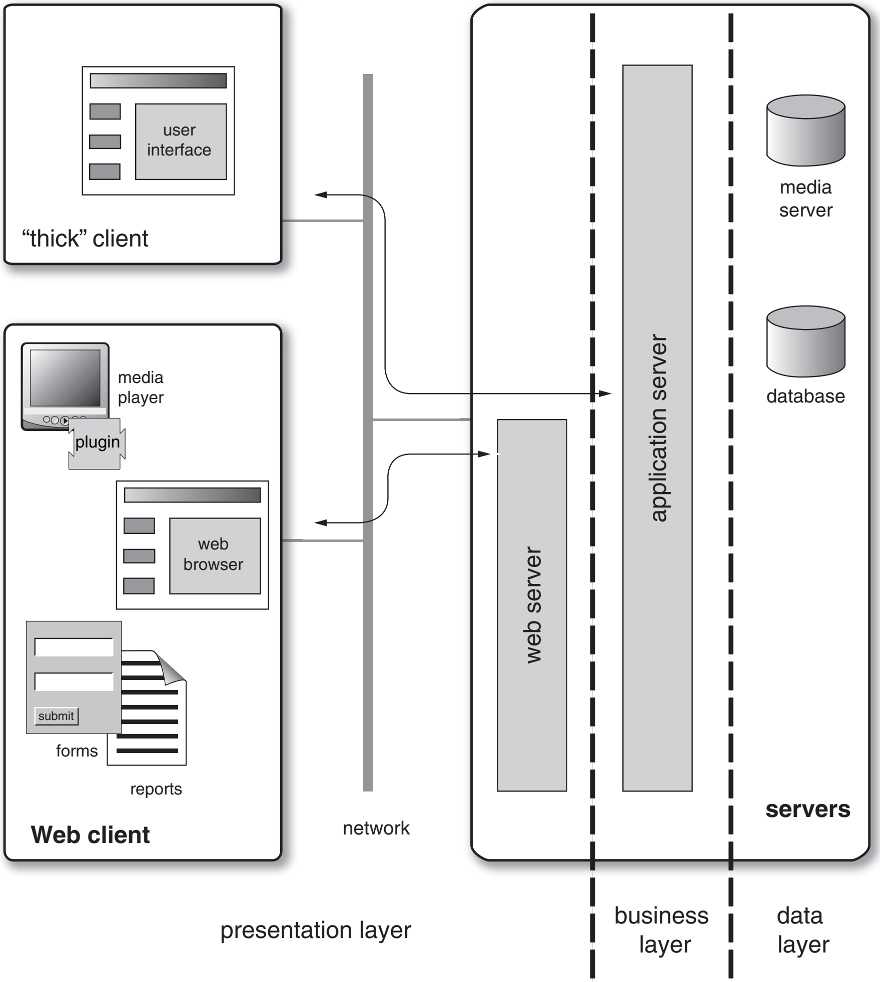

Figure 6.4 Multi- or *n*-tier computing

The central server has now become fragmented into functional blocks: data server, web server, and application server.

Client–SOA

The rise of the service-oriented architecture (SOA) for computing has led to a variant of the client-server that uses web services. The SOA uses loosely coupled services and is especially useful for communication between the DAM system and third-party applications.

Application Server

The business logic runs on the application server. This provides the services and framework to run the software objects that implement the business logic. The framework has two subsystems. One is to implement the rules of the business logic, the processes like the workflow and categorization. The second is the transaction management for updating metadata records in the database.

The Data Tier

The data tier is usually a relational database, although this is for administrative convenience rather than elegance of design. It could alternatively be an object database. The database has a management system, DBMS, to control the connections and for the general maintenance of the database (Figure 6.5). The application server is a client to the database. The DBMS manages the transactions to enter and update data, and the information retrieval to generate views and reports of the data. The database administrator (DBA) has a number of tools to maintain the database.

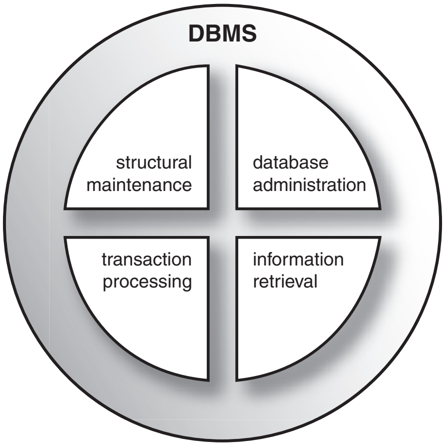

Figure 6.5 Elements of the database management

Originally, each database product required a different driver for the applications to perform operations on the data. Microsoft developed the language-independent open database connectivity (ODBC) interface that rationalized the database connections into a single standard. This was later joined by Java database connectivity (JDBC), which exposes a Java application program interface to the database drivers. These technologies have much simplified the connection between the business logic and the database. There are limitations to ODBC; it was designed to use SQL for access to relational databases. There are now many data sources that are not relational, mail systems, object databases, and file servers. There have been newer and more efficient components that can provide a more universal data access than ODBC.

Limitations of the Web Tier

Some clients need more functionality than can be provided by the basic web browser. For example, media ingest requires PCs equipped with specialist video-encoding cards and extra processing power for the speech analysis. Many digital asset management products use a hybrid approach. Simple search and retrieve operations can be made from a web browser, but more complex ingest and editing operations use a full client application.

So far, I have been describing closed systems, a corporate network. Most collaboration in asset management extends outside the enterprise. Customers and partners will need controlled access to the asset management. Again, the web browser provides a simple option. The remote user needs no special software application; the technology is familiar and ubiquitous.

The Content Model

The Society of Motion Picture and Television Engineers/European Broadcasting Union (SMPTE/EBU) Task Force on Harmonized Standards in 1998 developed a system model to represent the relationships between signals, processes, and control systems (Figure 6.6). This report has become a very popular starting point for building object models of content.

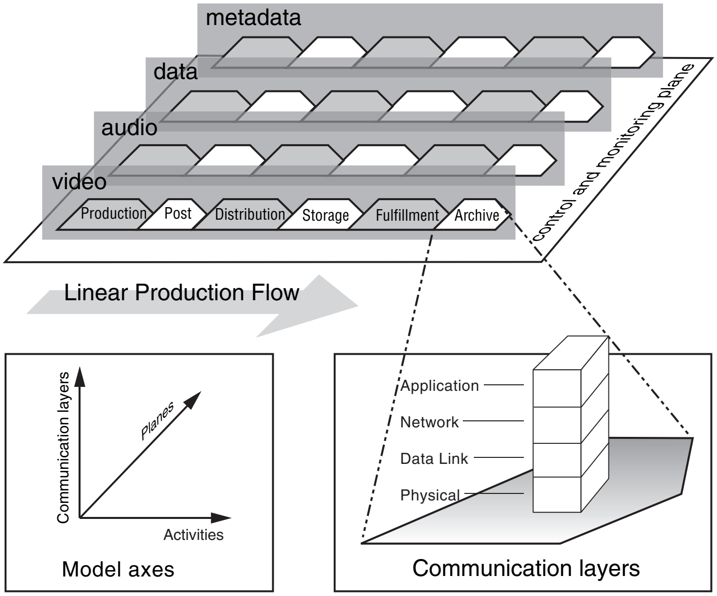

Figure 6.6 System model for television production

The model has three orthogonal axes: activities, content planes, and communication layers. The communication layers represent the intercommunication between peer entities. The layers are similar to the International Standards Organization (ISO) open systems interface: application, network, data link, and physical. A control and monitoring plane lies under the activities and content planes.

This model focused on television content, so splits content into audio, video and data (all classed as essence) and metadata, but the principle could be applied to other forms of content. Essence is the raw content that represents the pictures, sound, and text that are delivered to the user. Data essence can be graphics, animation, text files, and still images. An example of data essence is the closed caption.

The activities are those typical of the processes in the workflow of television production. The production phase represents planning and shooting. Post is the post-production, where the original footage is edited into a final program, the graphics added and the sound design implemented. Distribution is the dissemination of the finished program to the publishers. These could be broadcasters or DVD distributors. The storage operation is optional, as the product may be aired immediately. Fulfillment is the delivery to the consumer. This could be as broadcast television transmission or as sell-through media: DVD and VHS. Finally, the program is archived to a vault.

This flow is linear, but many programs return from the vault to the post-stage for repurposing and reuse in a cyclical process.

The Asset Management

The asset management can be looked at as a number of blocks. This view is somewhat flexible, what is content, what is data? In this view content that is stored in a file system like the network file system (NFS) is called data, content that is stored as video on tape or in special video servers is treated separately as content and managed through the media management component.

Most products are modular, and sit like a hub at the center of media operations and processes. The core asset management should offer a number of different functions:

- Indexing and categorization
- Search
- Content editing
- Workflow management
- Task and project management
- Resource management
- User management
- Storage management

The asset management functions will be supported by a number of services:

- Load balancing and fault tolerance
- Access control
- Security
- Configuration

Figure 6.7 shows the Mediaflex system from Transmedia Dynamics. This has most of the elements of a fully featured asset-management system.

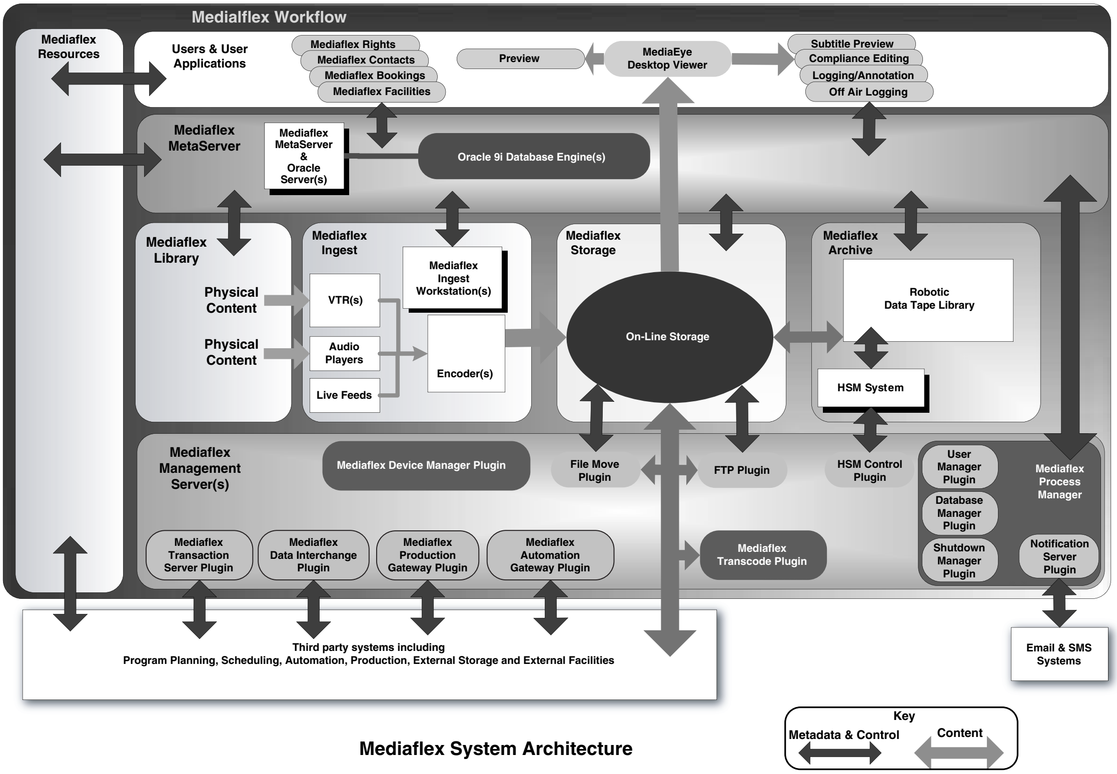
Medialflex Workflow
Mediaflex Resources
Users & User Applications
Mediaflex Rights
Mediaflex Contacts
Mediaflex Bookings
Mediaflex Facilities
Preview
MediaEye Desktop Viewer
Subtitle Preview
Compliance Editing
Logging/Annotation
Off Air Logging
Mediaflex MetaServer
Mediaflex MetaServer & Oracle Server(s)
Oracle 9i Database Engine(s)
Mediaflex Library
Physical Content
Physical Content
Mediaflex Ingest
VTR(s)
Audio Players
Live Feeds
Mediaflex Ingest Workstation(s)
Encoder(s)
Mediaflex Storage
On-Line Storage
Mediaflex Archive
Robotic Data Tape Library
HSM System
Mediaflex Management Server(s)
Mediaflex Device Manager Plugin
File Move Plugin
FTP Plugin
HSM Control Plugin
User Manager Plugin
Database Manager Plugin
Shutdown Manager Plugin
Mediaflex Process Manager
Notification Server Plugin
Mediaflex Transaction Server Plugin
Mediaflex Data Interchange Plugin
Mediaflex Production Gateway Plugin
Mediaflex Automation Gateway Plugin
Mediaflex Transcode Plugin
Third party systems including
Program Planning, Scheduling, Automation, Production, External Storage and External Facilities
Email & SMS Systems
Mediaflex System Architecture
Key
Metadata & Control
Content

Figure 6.7 A typical media asset-management system. © Transmedia Dynamics

Indexing and Categorization

The index module uses metadata generated at the ingest stage to create an index and catalog that can be used by the search engine. The indexing can be fairly simple using a few key fields: file name, title, author, and keywords. For a large repository, a more sophisticated index will be required. This will avoid the common problem of huge result sets with little relevance to the search criteria. One technique to return smaller and more relevant result sets, is to use concepts rather than keywords.

Search

Search is an essential facility for any asset management. The user can use the search engine to find content within the repository. Search engines vary from a basic keyword search through to natural-language search. Concept searches may well suggest allied content that may be relevant.

Content Editing

Many asset-management applications include basic video-editing facilities, typically a rough cut. A user can assemble a number of scenes or clips into a contiguous sequence. This assembly of clips