THE CHALLENGE OF A DATA STORAGE HIERARCHY

Mr. Michael Ruderman
Vice President for Marketing
Mesa Archival Systems, Inc.

MR. RUDERMAN: Good morning. I have the pleasure of standing between you and lunch. So, I will try and speak quickly and get through as much of this material as we can. I'll tell you a little bit about Mesa Archival Systems first.

We are a small private company located in Boulder, Colorado. We were founded as a technology spin-off of software originally developed at NCAR, the National Center for Atmospheric Research. Currently, their version of this software is managing 19 terabytes of data on over 102,000 3480 cartridges. They are in the process of moving from 3480 to 3490 tape cartridges, but it is clear that this represents a large amount of data from a variety of systems and is composed of fairly large set of bitfiles.

Something else you ought to know about Mesa Archival is that we are part of the winning team of the recent NASA Goddard mass storage award that hopefully, barring any obstacles--legal or otherwise--you will all be able to see running here at NASA Goddard in the next couple of months. We are looking forward to that. (Showing of viewgraphs) MR. RUDERMAN: Now, before going into the storage hierarchy aspect of the system, I will put it in some perspective with a brief overview. What is the system in general?

It is a data archiving system; it fits into the kinds of environments that all of you have or are working towards, whether you are vendors or users, with multiple heterogeneous client systems networked to a central archive server.

That is the simple explanation.

(Change of viewgraph)

MR. RUDERMAN: We can picture our system, as illustrated. Obviously, any kind of client system on the network can be connected through the Unix interface of our system. We are strictly a software system sitting on a mainframe, managing the data into permanent file storage. We do not have time today to go into all the details of how we do everything. But suffice it to say, it is a central archive manager, with standard network access and it is a standard commercial product--that is very important.

We came from a lab environment; however the version that is now available is not lab code. It is commercial product code. It sits in a high-performance computer, accessible from multiple processors.

(Change of viewgraph)

MR. RUDERMAN: And just to reiterate, when we took this code from NCAR and designed it for commerciality, it had to be designed for change. Much of what you are hearing during this conference is a lot of information, from both users and suppliers, about a lot changes taking place in the area of mass storage devices.

We believe one of the most important ideas to keep in mind is that data must outlive any device or any media; the data is more important than any of those things.

The hardware is going to keep changing; and so, you have to have a software control system that is independent of any hardware system. We believe in that.

(Change of viewgraph)

MR. RUDERMAN: Let's take a look at the next level into the system. As I said, we support standard networks, TCP/IP with FTP access and NETEX with User Access client support. We put no code on client systems. The network access server provides the access; the data library access manager provides all the client and system communication dialogue. As demand warrants, support of additional new network protocols will be added easily.

The archival object database is the heart of the system, and here we have implemented our own object oriented analysis. At the back end of the system there are multiple storage servers, and of course, system administration. This is where we will focus on a little more detail.

Before getting into that, I want to mention that we are an active member of the IEEE Mass Storage Reference Committee and Storage Subsystem Working Group, - we have been involved with it from the very, very beginning. In fact, NCAR was one of the originators of the whole idea back in the mid-1980s.

(Change of viewgraph)

MR. RUDERMAN: This is our version of the IEEE components model, and what I want to talk about today is the storage system side of it. As you can see, we have separate data paths between bitfile movers--the bit movers from the client side and the server side--and here we have multiple storage servers, which is the area that we want to look at. The issue is how do we implement that?

(Change of viewgraph)

MR. RUDERMAN: With the variety of devices available--and you have all been hearing about a whole bunch of them in the last couple of days--the variations cover access speed, permanent versus temporary storage, capacity, cost--cost is a major factor.

I think half of you could go home--those of you who are hardware vendors--if there were infinite rotating disk storage available at a very low price, we probably wouldn't be worrying about this conference too much. Life would be simple, but it's not that way.

And products keep changing. What may seem like simple changes from 3480 to 3490 technology have implications--big implications. New products--D-1, D-2, optical disk, optical tape, etc.. Our objective on the software side is to keep the data independent of any of these hardware changes.

(Change of viewgraph)

MR. RUDERMAN: Now, this morning, you heard Bob Coyne from IBM talk about the storage hierarchy and the issues and problems of a static storage hierarchy. Because there are now multiple devices being implemented in single archives, both for cost reasons, technical reasons, and user reasons, the traditional, what we call "static hierarchy" of primary and secondary storage alternatives, is just not sufficient.

The problem surfaces as soon as you try to figure out what you do with the third one you want to add--the third type of storage device. Where do you put it in the hierarchy? For different kinds of data it belongs in different places.

We have examined this; we have spent a fair amount of time on this over the past year or so. To say that we have solved it, is easy. What we will try and show you is how--the conceptual approach as to how we solved it. This is, in fact, what is available from Mesa Archival Systems today.

(Change of viewgraph)

MR. RUDERMAN: To give you a feeling for what I'm talking about a little more specifically, we can see that from the data library system, we can support magnetic disk, optical disk, cartridge tape, helical scan magnetic tape--all of these kinds of storage devices, multiple devices, heterogeneous devices--simultaneously.

You cannot do that very efficiently with the traditional static hierarchy of a fixed physical system. So, we have developed a **structured hierarchy**, which gives multiple views of those physical storage options. It is dynamic, and it must be able to be varied by user, by bitfile--down to the bitfile level--by class of data, by accounts, by any combination of categories to be derived for each system.

Now, that is an easy thing to say, but it's not so easy to implement. Let me just make a note. I see some of you looking through the proceedings and not necessarily finding all the slides identical. We recently updated some of these and created some new ones for today; and I apologize for any confusion.

We didn't get them sent in, but we will get a new set available for the followon proceedings book. So, you might as well just ignore what's in there at the moment. (Change of viewgraph)

MR. RUDERMAN: It's easy to say, not so easy to do.

The first thing we did is separate the archive system from the storage system conceptually. Bitfiles come into the system. There can be one or more images of them, depending upon the attributes of the bitfile. You may want to keep a copy on disk, put a copy on optical; you may want to keep copies on multiple devices for various reasons.

Each image of a bitfile, unless it is very small, will have multiple fragments; and these fragments reside on specific types of media. The media gets broken down further into packages and surfaces. I'll show you later how that all fits together.

But basically, the first thing to do is to separate the archive system from the storage system and implement the whole concept from the point of view of object-oriented analysis and design.

That makes the use of it simple; it makes the coding of it not so simple, but it has been accomplished.

(Change of viewgraph)

MR. RUDERMAN: Now, let's take a look at this from an object orientation. We have a client file in the system. This client file talks to the bitfile server, requesting services.

The bitfile to be archived will request storage from various storage options. Here we are just showing two; there can be any number of storage options based upon what the physical system has installed.

(Change of viewgraph)

MR. RUDERMAN: As we take a look at this, we see the kinds of objects that sit under each of these categories. In the client file system we have the directories and files and users and groups. The bitfile archive has the bitfiles and accounts that they belong to, if that is appropriate, and the templates.

The templates are very important. The templates describe which of the storage options are available for a particular bitfile or, in other words, which of the options that are physically on the system are the permitted options in this instance. In this way, the storage hierarchy is constructed for each different need.

All of the communication between the various tasks is very standard client server protocol, make a request, receive a reply; it makes use of the system very simple, as well as modification and administration of the system--very, very simple. (Change of viewgraph)

MR. RUDERMAN: This is the simplicity of the storage option view from the bitfile. The bitfile doesn't see much more than that.

However, below this point, we have analyzed and organized all of the information that we believe we need to know. The head of development doesn't like me to say "all" about anything; he's very sensitive about that, but I believe it's all of it that we need to know, and it works!

We have broken this down to fixed and removable media. Disk is obviously fixed, with a one-to-one media/device relationship. Most of the removable media that we are dealing with, on the systems that we support today, use 3480 tape interface. Obviously, a 3480 cartridge tape is a one-to-one; however, the implementations of both helical scan and optical disk have multiple logical 3480 volumes on each physical volume.

The reason for this analysis and breakdown is to make it simple, to be able to both move data and make requests of the system, such as change devices in and out. We are not aware of any devices that we cannot put into this organization, and with it, we will make the subject of managing them very, very simple.

Now let me try and give you an example of what we have been able to do with this approach.

(Change of viewgraph)

MR. RUDERMAN: If we take a look at a particular system, let's say this system has these options available to it. This includes the disks, helical tape, optical tape, and standard cartridge tape.

A particular bitfile in the system may have attributes such that it can reside on disk or helical tape or optical tape--it doesn't matter. If there is an image sitting on disk at a point in time and this disk reaches its threshold and an event is triggered such that it needs to be migrated or "scrubbed," as we call it, the option is: Do I put it here on helical tape or do I put it here on optical disk?

We have built the system and designed the system in such a way that, as we can get more information feedback from the hardware system, we can make that selection much more sophisticated. What I'm driving at here is that the system manager may have said: Because helical is faster, the next best choice for this file, when you migrate from the disk is to go to helical tape.

If at a particular moment in time the helical system has large cues and is overloaded, from a performance point of view, if the optical tape were available, it might make more sense at that point in time to go there.

The kind of information required to be able to make those selections automatically, transparent obviously to the user and transparent to the bitfile, will require some additional information that tends not to be available today from most robotic suppliers.

And that gets us into the challenge that we see in dealing with robotic storage devices, and we have dealt with quite a few of them. When there are multiple and/or shared robotic systems in a single archive, there are some potential problems that we think most of the robotic manufacturers have not anticipated. They tend to think they are the only devices on the system and they have not anticipated the need for programmatic dialogue about their own status.

But I believe that you will find that the user community is going to demand multiple heterogeneous storage systems to be installed, and the ability to install and remove and add and change any device at any time without taking down the archive demands the approach we have taken.

So, what we have found is that the ability to do reads and writes is easy; anybody can do that. But the lack of a standard programmatic interface between software archive systems and the storage devices is a problem. The one that we have implemented, and we are pushing to become IEEE standard is the standard client server protocol. It's the same as the ISO managed object interface for networks. It's the same protocol we use internally between all of our task communications.

(Change of viewgraph)

MR. RUDERMAN: This is something we have all seen in any client server relationship. With a storage server in this case, the client would make a request to the server, and get a reply and/or be triggered with an event. (Change of viewgraph)

MR. RUDERMAN: For instance, if an automated media library is shared, then how do we know if a package we need is not being used elsewhere or has been ejected from the device?

Because we are dealing with high performance requirements and massive amounts of data, we would like to know that before we issue a read or a write so that we don't hang up the archive system. The objective is to improve performance, not issue commands to any device when a piece of media or the device itself is not available.

This is what we are looking to see. We have a design for it; we have implemented it with certain devices, and others of you in the vendor world need to think about how you are going to coexist in this environment.

(Change of viewgraph)

MR. RUDERMAN: Just summarizing. The devices currently supported, that we have experience with, the 3480 cartridge, the STK silo, the Memorex tape library, Dataware optical disk, and Masstor helical scan tape.

We see in the future adding additional archival devices. We are very interested in lots of the new devices, in D-1 and D-2 areas especially. We also will be expanding network connectivity to new standards, as they emerge.

We intend to be very oriented towards standards. Additional operating system support is probably one that a few of you might be interested in. For those who don't know, right now we are MVS-based. We definitely have plans for expanding beyond MVS to Unix. A fundamental objective must be that all changes need to be transparent to the client systems.

(Change of viewgraph)

MR. RUDERMAN: When we talk about adding new operating system support, what we are really saying is that we don't see the world of data archiving suddenly and totally abandoning MVS for Unix. MVS systems are very, very powerful for moving massive amounts of data, simultaneously.

We see the next version of our software to be able to have multiple distributed servers, both for bitfile serving and for storage servers and multiple processor hosting, including both MVS and Unix.

That's what we wanted to communicate with you today, and I'll think we'll make it to lunch. You may ask some questions if you'd like.

DR. FREESE: Thank you, Michael.

(Applause)

DR. FREESE: Questions, comments, discussion from the floor?

MR. SAVAGE: I do have one question. (Inaudible) DR. FREESE: Could you paraphrase that?

MR. RUDERMAN: Yes. The question had to do with, I believe, an interpretation of the NCAR system, which was that the archive manager was really just directing the requesting system or telling it where the data was, as opposed to actually shipping it to it.

Yes, what you are referring to is the fact that there is direct data transfer between the IBM disks and the Cray at NCAR. This is done for performance reasons, so the data does not have to be sent through the IBM mainframe.

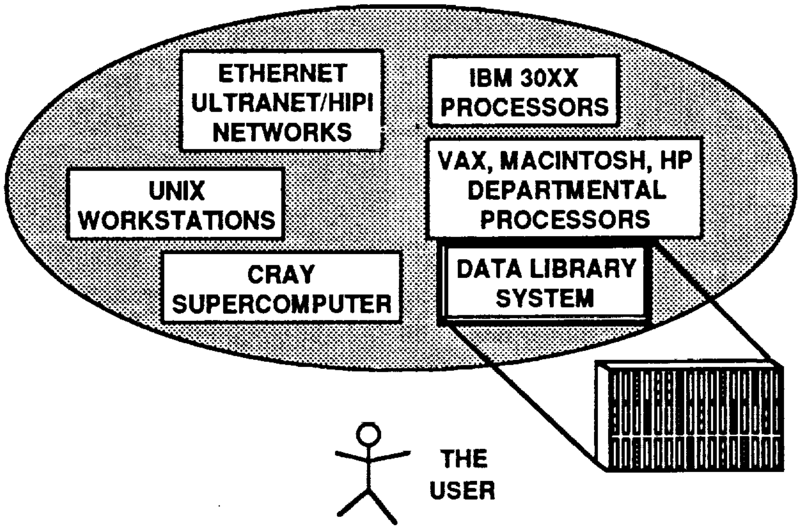
MR. SAVAGE: The IBM machine would look it up, find out what disk it was, and ship that information to the Cray. The Cray would then create a channel program and send it down over that channel to actually directly read the disk.

DR. FREESE: Any other questions or comments?

Data Archiving.
Michael Ruderman
Mesa Archival Systems, Inc.

MESA Archival Systems, Inc.

Computing Environment



ETHERNET ULTRANET/HIPI NETWORKS

IBM 30XX PROCESSORS

UNIX WORKSTATIONS

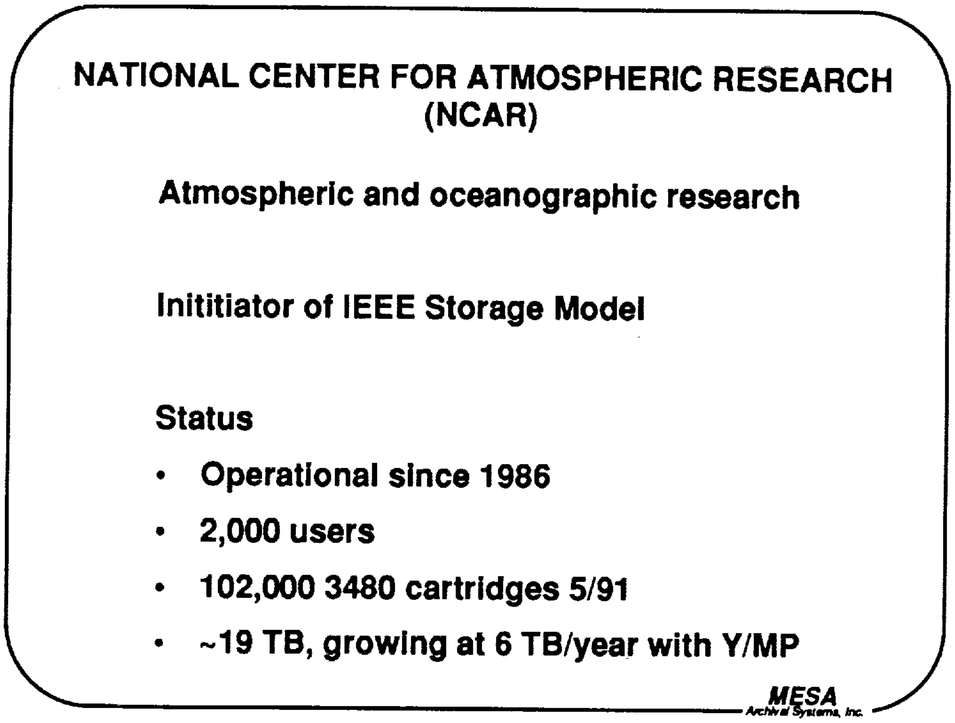
VAX, MACINTOSH, HP DEPARTMENTAL PROCESSORS

CRAY SUPERCOMPUTER

DATA LIBRARY SYSTEM

THE USER

MESA Archival Systems, Inc.



NATIONAL CENTER FOR ATMOSPHERIC RESEARCH (NCAR)

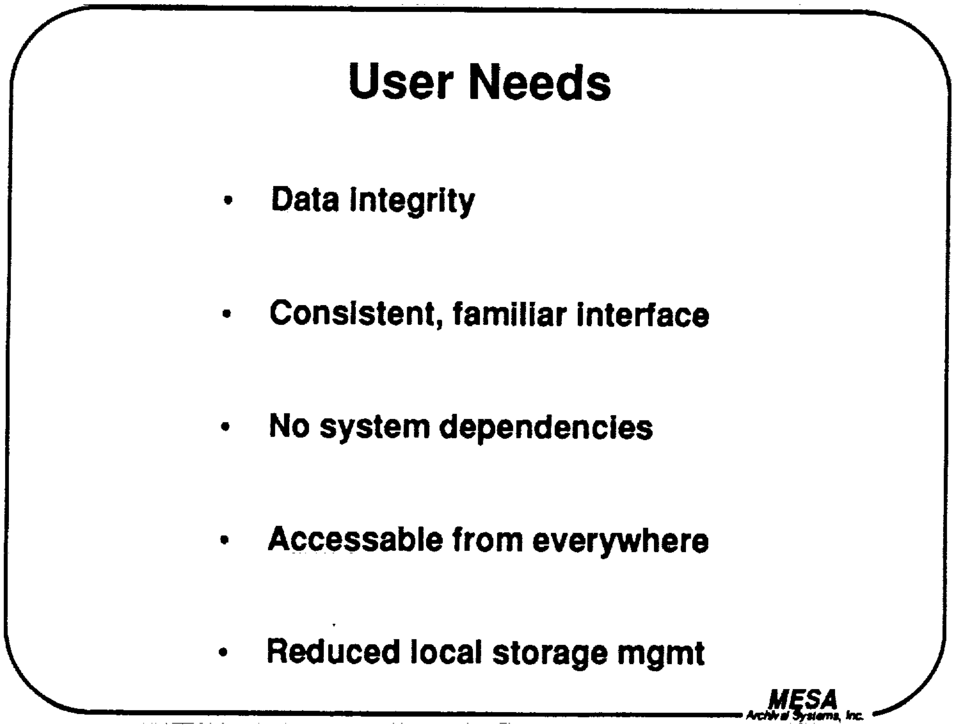
Atmospheric and oceanographic research

Inititiator of IEEE Storage Model

Status

- Operational since 1986
- 2,000 users
- 102,000 3480 cartridges 5/91
- ~19 TB, growing at 6 TB/year with Y/MP

MESA Archival Systems, Inc.



User Needs

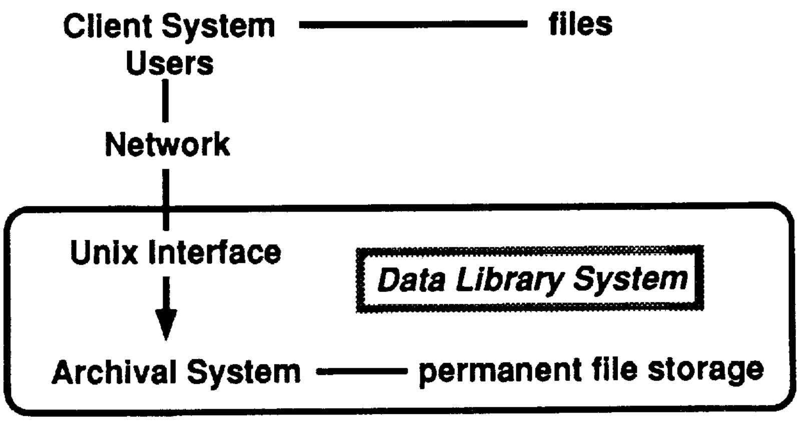
- Data Integrity
- Consistent, familiar interface
- No system dependencies
- Accessable from everywhere
- Reduced local storage mgmt

MESA Archival Systems, Inc.

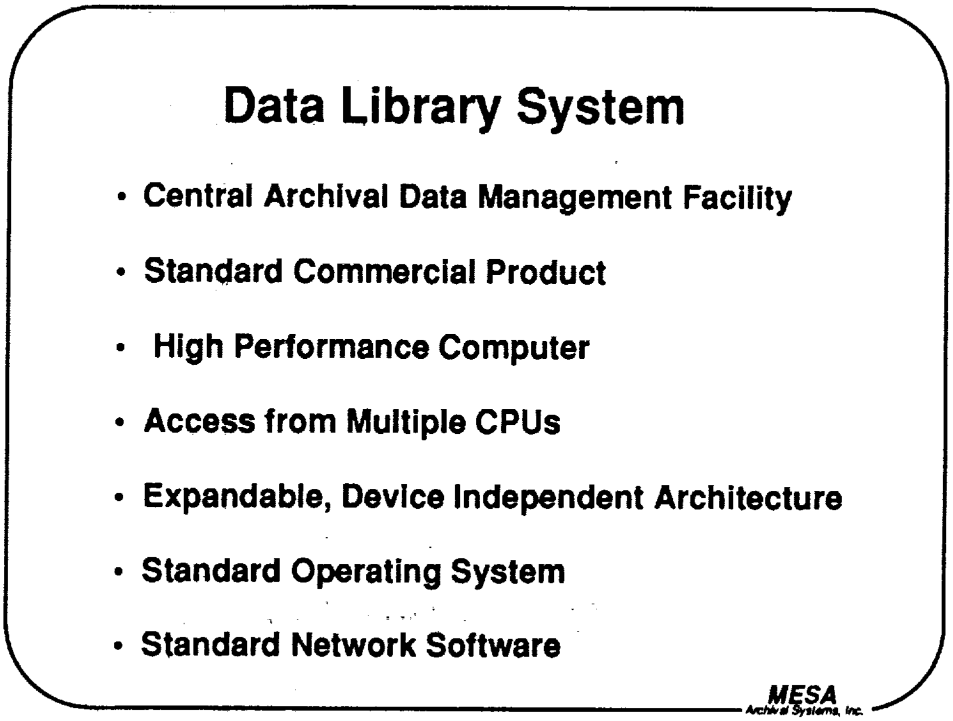
System Manager Needs

- Data integrity
- Storage hierarchy
- Mass storage alternatives
- Ability to deal with change
- Accountability by user
- Performance
- Reduced operations cost

MESA Archival Systems, Inc.



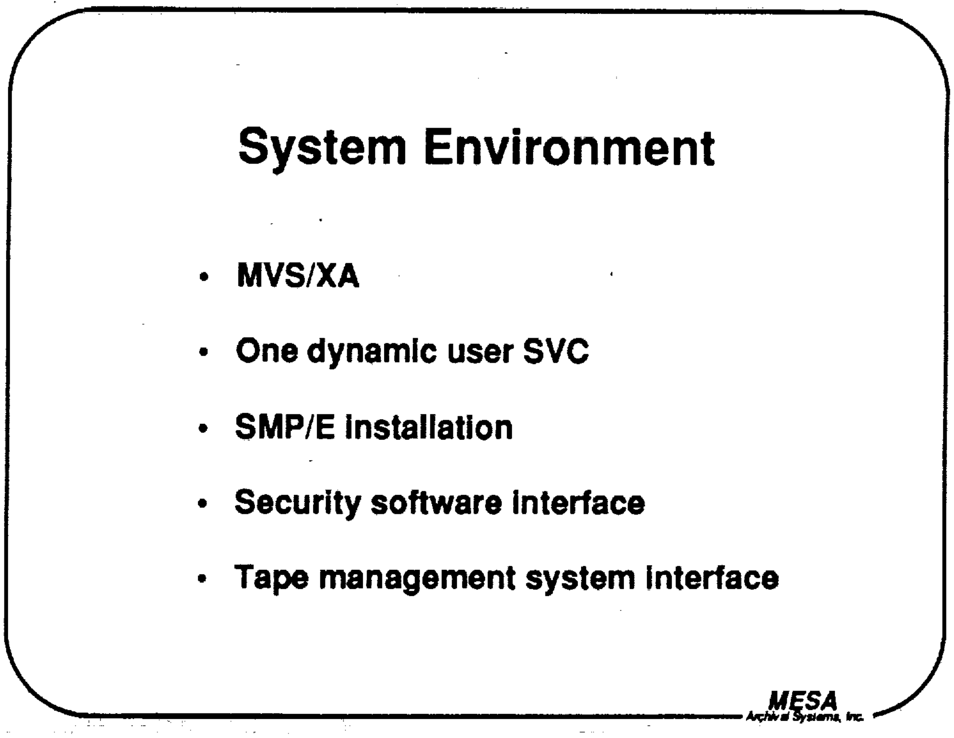
MESA Archival Systems, Inc.



Data Library System

- Central Archival Data Management Facility
- Standard Commercial Product
- High Performance Computer
- Access from Multiple CPUs
- Expandable, Device Independent Architecture
- Standard Operating System
- Standard Network Software

MESA Archival Systems, Inc.

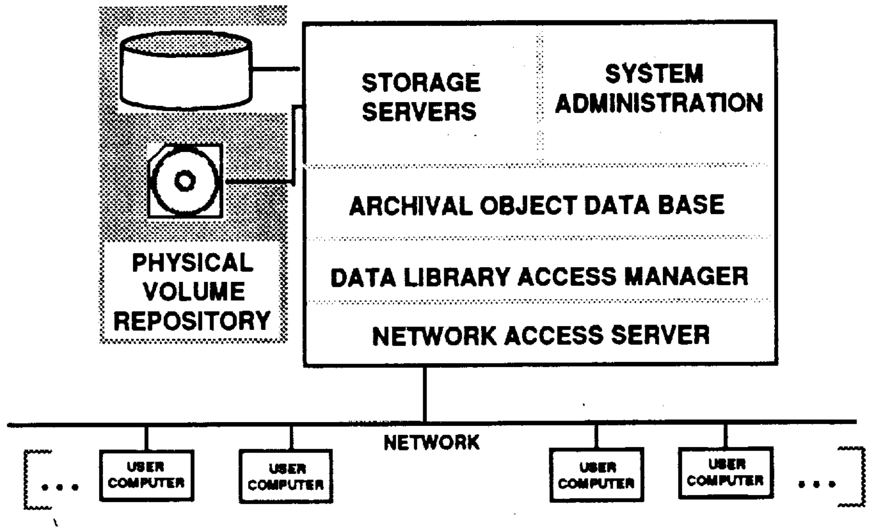


System Environment

- MVS/XA
- One dynamic user SVC
- SMP/E installation
- Security software interface
- Tape management system interface

MESA Archival Systems, Inc.

Data Library System



STORAGE SERVERS

SYSTEM ADMINISTRATION

ARCHIVAL OBJECT DATA BASE

DATA LIBRARY ACCESS MANAGER

NETWORK ACCESS SERVER

PHYSICAL VOLUME REPOSITORY

NETWORK

USER COMPUTER

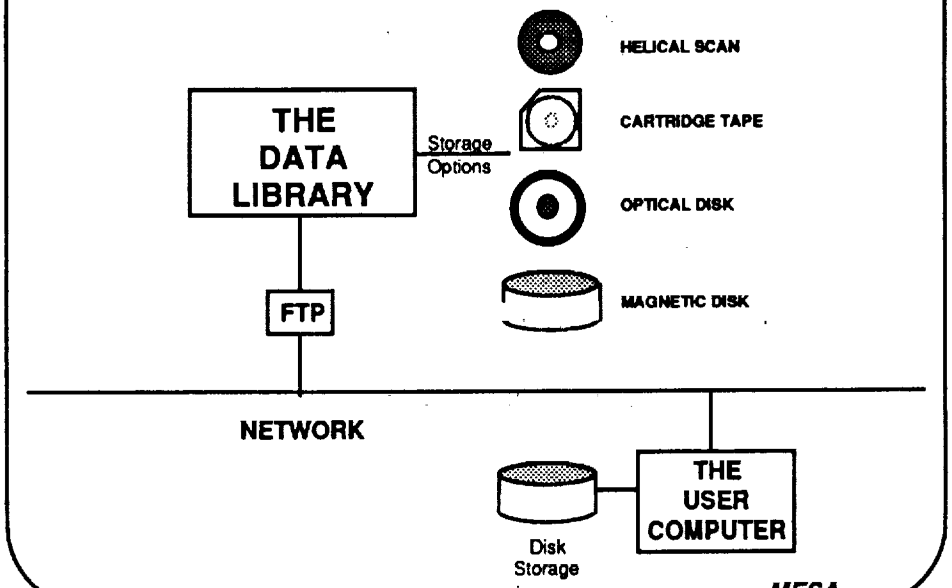
USER COMPUTER

USER COMPUTER

USER COMPUTER

MESA
Archival Systems, Inc.

DLS Environment



HELICAL SCAN

CARTRIDGE TAPE

OPTICAL DISK

MAGNETIC DISK

THE DATA LIBRARY

Storage Options

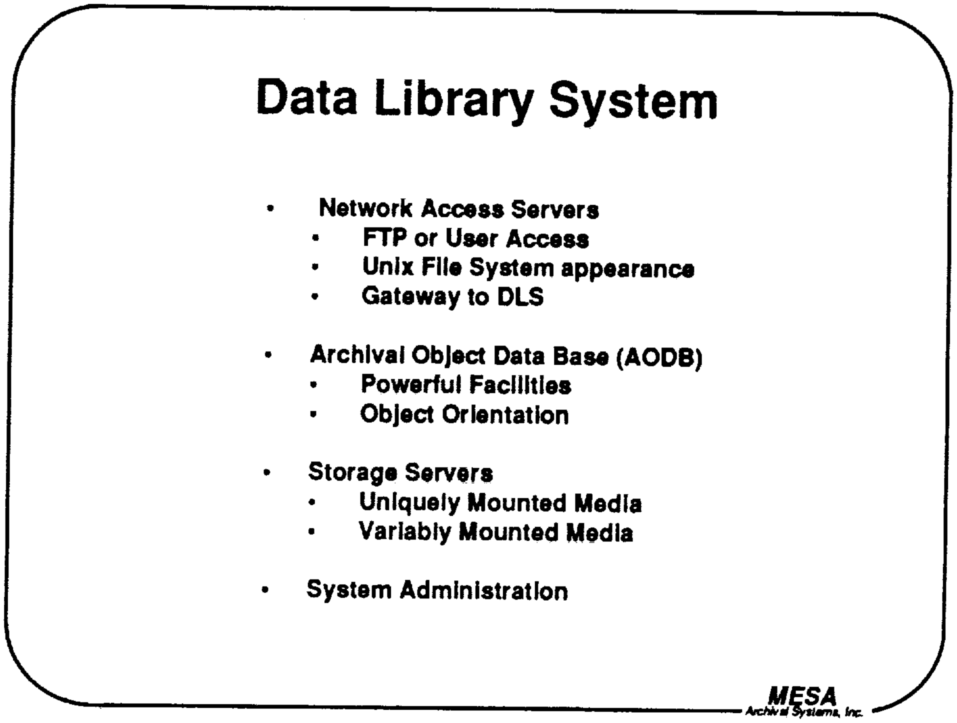
FTP

NETWORK

THE USER COMPUTER

Disk Storage

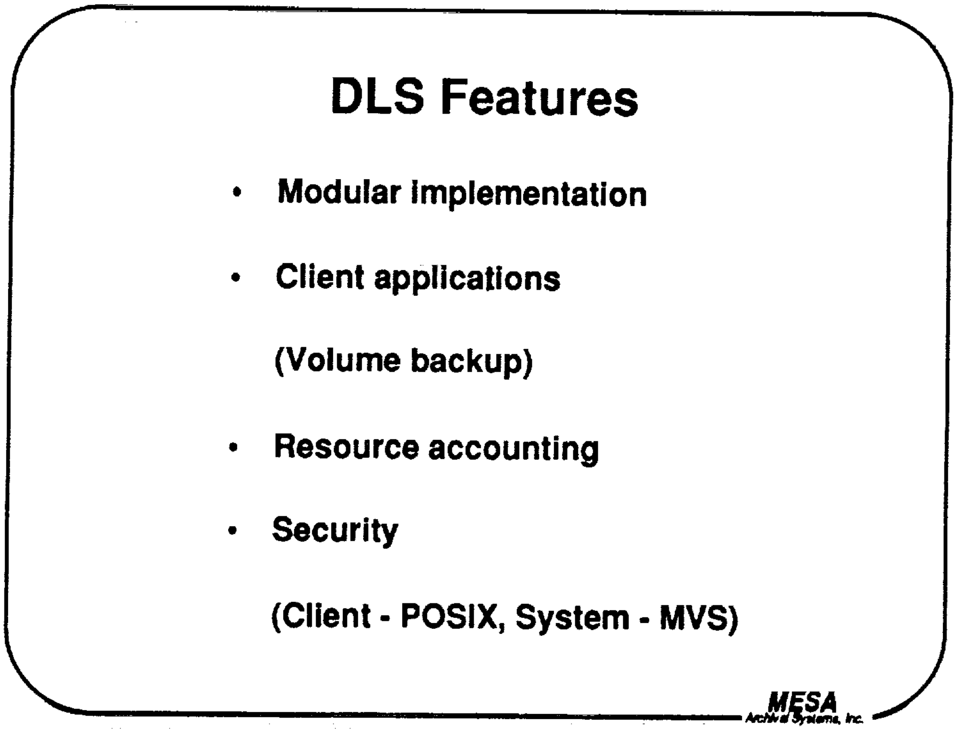
MESA
Archival Systems, Inc.



Data Library System

- Network Access Servers
  - FTP or User Access
  - Unix File System appearance
  - Gateway to DLS
- Archival Object Data Base (AODB)
  - Powerful Facilities
  - Object Orientation
- Storage Servers
  - Uniquely Mounted Media
  - Variably Mounted Media
- System Administration

MESA Archival Systems, Inc.



DLS Features

- Modular implementation
- Client applications

(Volume backup)

- Resource accounting
- Security

(Client - POSIX, System - MVS)

MESA Archival Systems, Inc.

Networks

Protocols

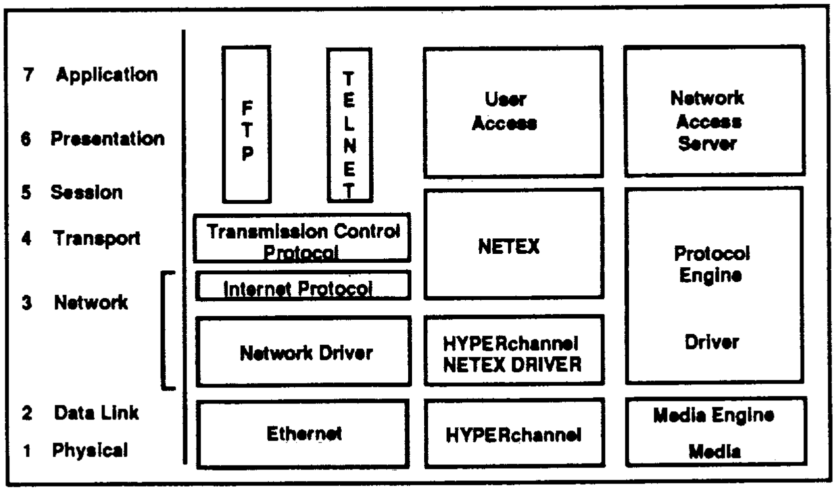
- FTP, User Access
- TCP/IP, NETEX

Networks

- Ethernet
- Ultranet
- HYPERchannel

MESA Archival Systems, Inc.

OSI Model



7 Application

6 Presentation

5 Session

4 Transport

3 Network

2 Data Link

1 Physical

FTP

TELNET

Transmission Control Protocol

Internet Protocol

Network Driver

Ethernet

User Access

NETEX

HYPERchannel NETEX DRIVER

HYPERchannel

Network Access Server

Protocol Engine

Driver

Media Engine

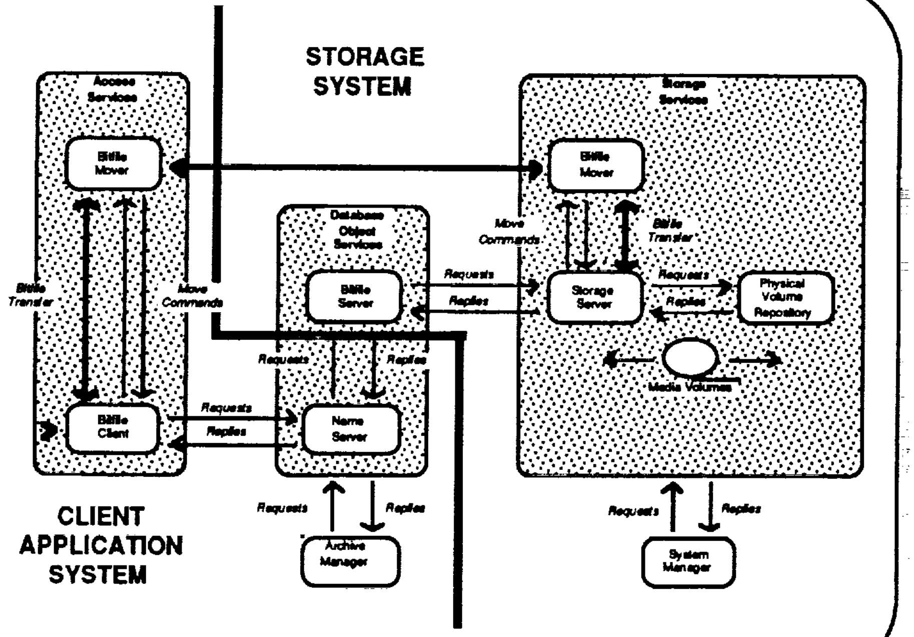
Media

MESA Archival Systems, Inc.

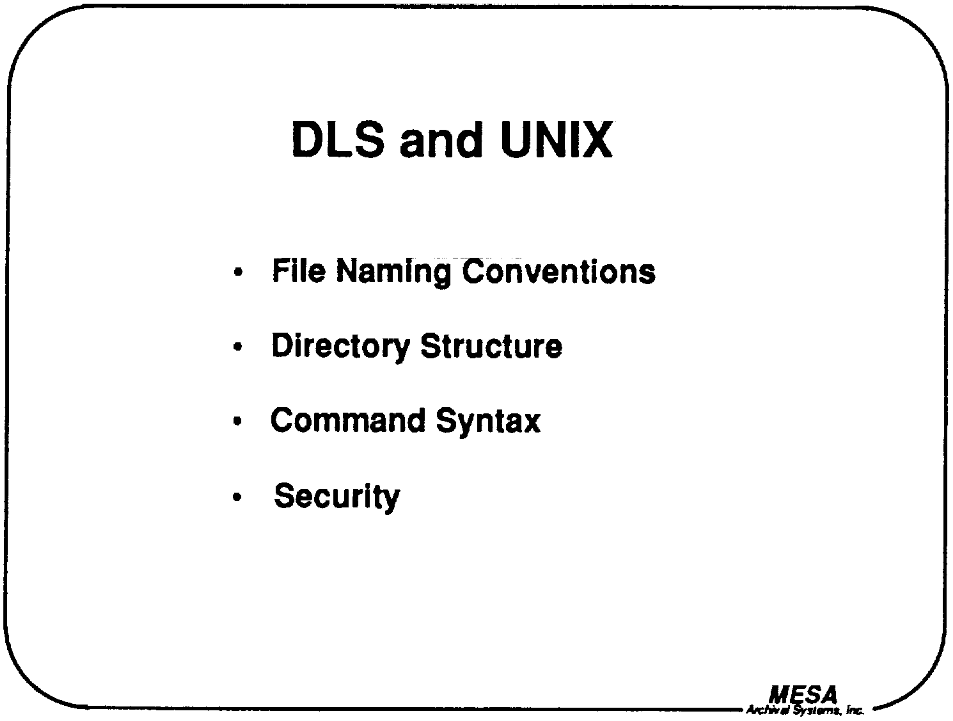
IEEE Mass Storage Reference Model

- Deals with Named Files - "Bitfiles"
- File Structure Insensitive
- Modularity of Design
  - Application Client
  - Bitfile Server
  - Storage Server
  - Physical Volume Repository
  - Bitfile Mover
  - Name Server
  - Site Manager

MESA
Archival Systems, Inc.



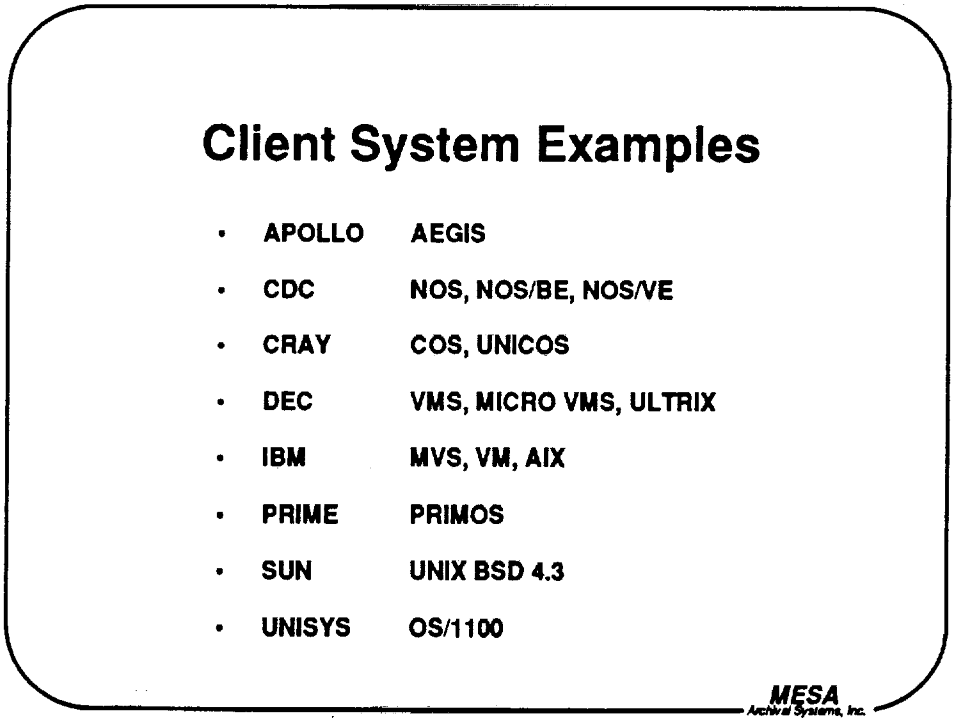
MESA
Archival Systems, Inc.



DLS and UNIX

- File Naming Conventions
- Directory Structure
- Command Syntax
- Security

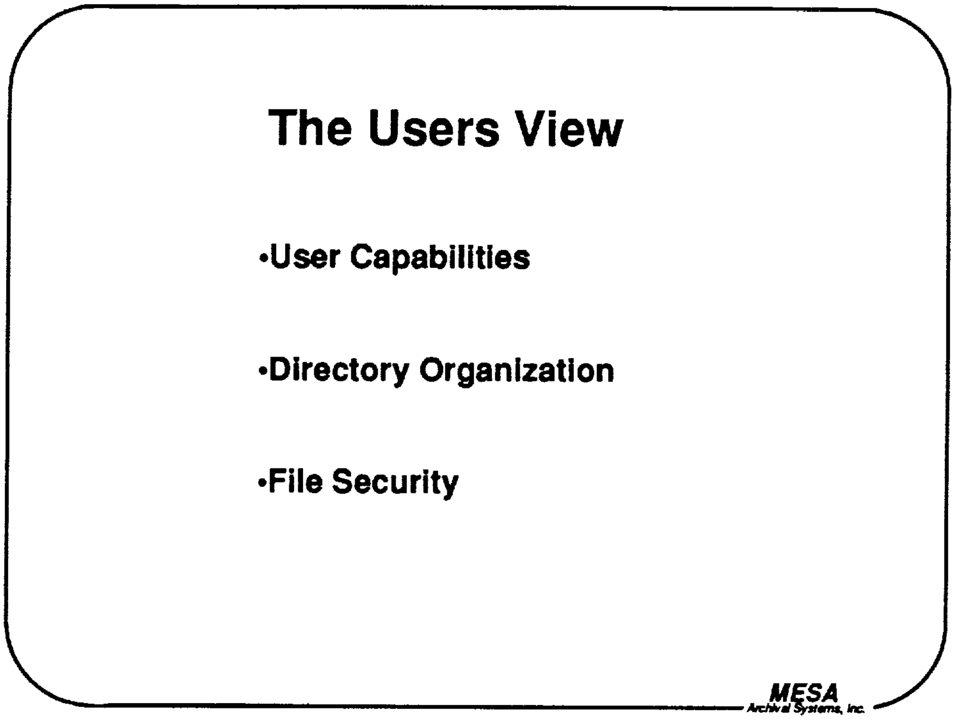
MESA Archival Systems, Inc.



Client System Examples

• APOLLO	AEGIS
• CDC	NOS, NOS/BE, NOS/VE
• CRAY	COS, UNICOS
• DEC	VMS, MICRO VMS, ULTRIX
• IBM	MVS, VM, AIX
• PRIME	PRIMOS
• SUN	UNIX BSD 4.3
• UNISYS	OS/1100

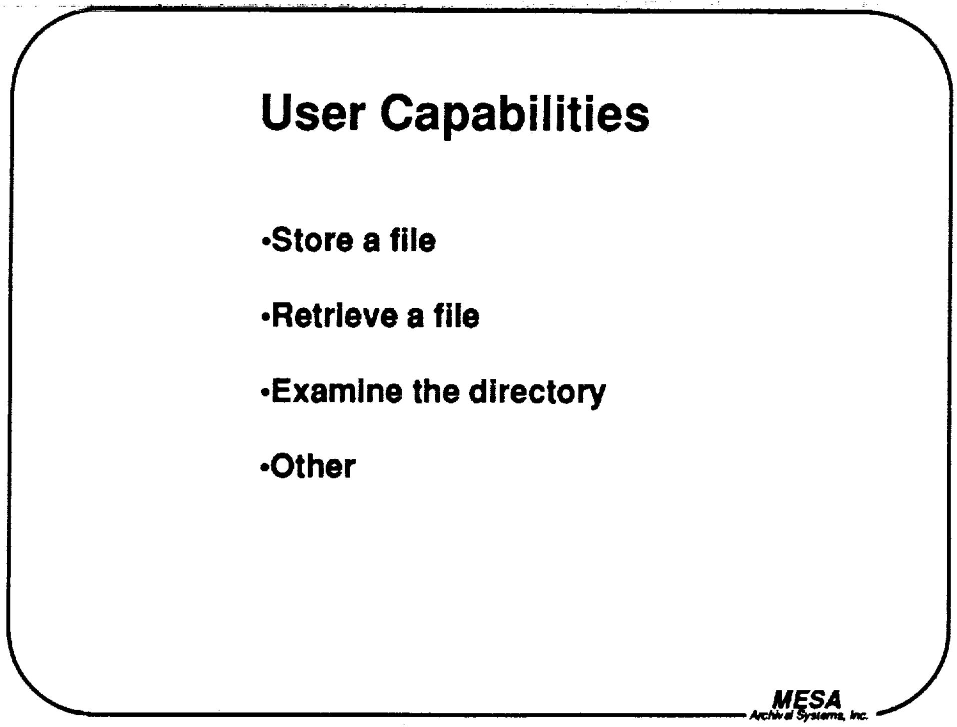
MESA Archival Systems, Inc.



The Users View

- •User Capabilities
- •Directory Organization
- •File Security

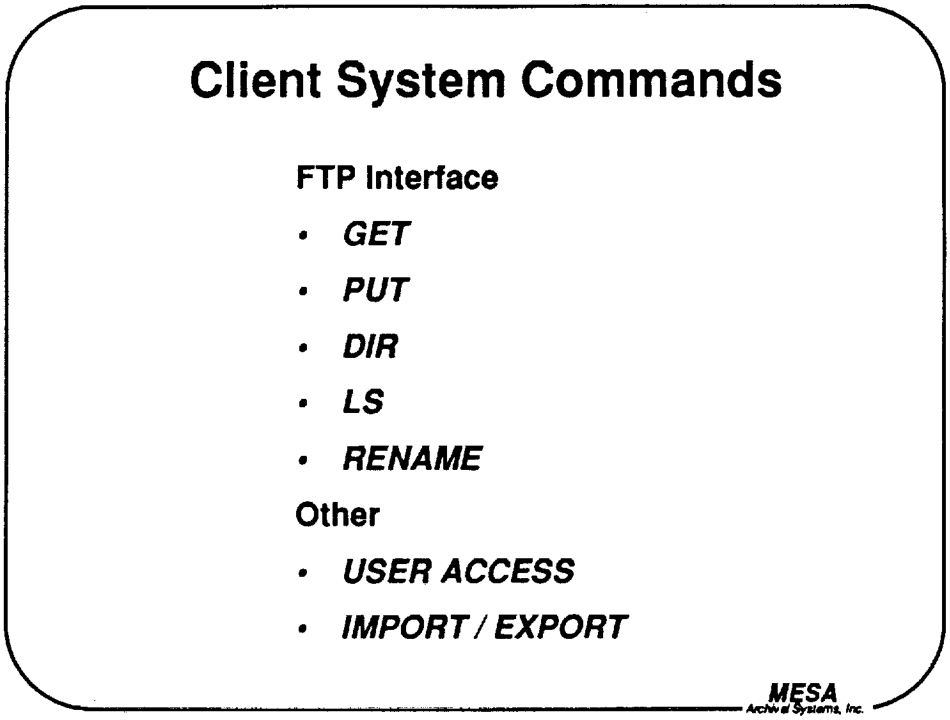
MESA
Archival Systems, Inc.



User Capabilities

- •Store a file
- •Retrieve a file
- •Examine the directory
- •Other

MESA
Archival Systems, Inc.



Client System Commands

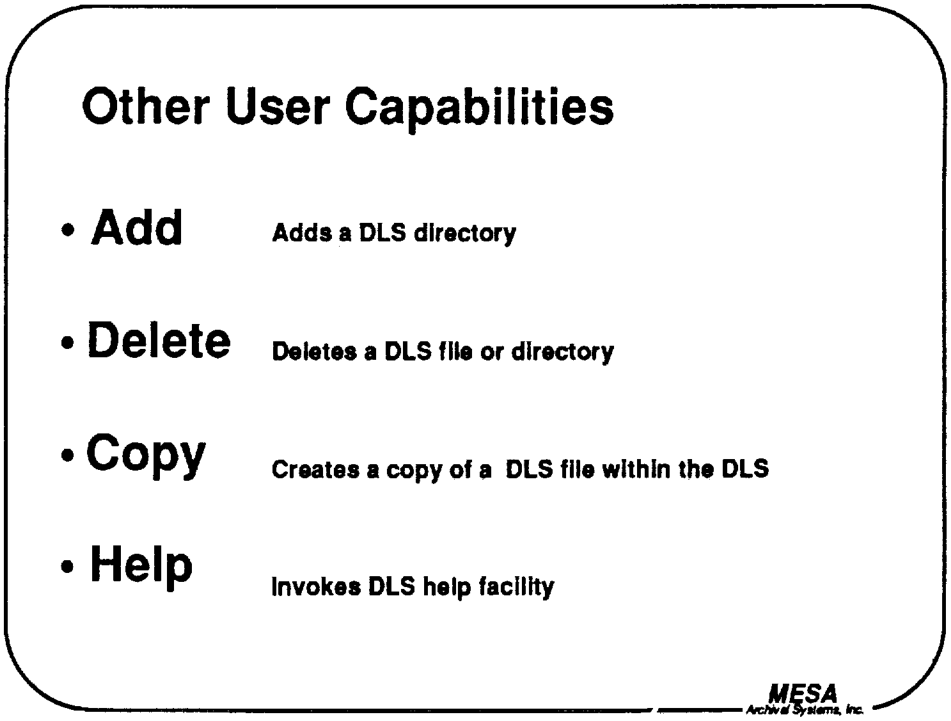
FTP Interface

- *GET*
- *PUT*
- *DIR*
- *LS*
- *RENAME*

Other

- *USER ACCESS*
- *IMPORT / EXPORT*

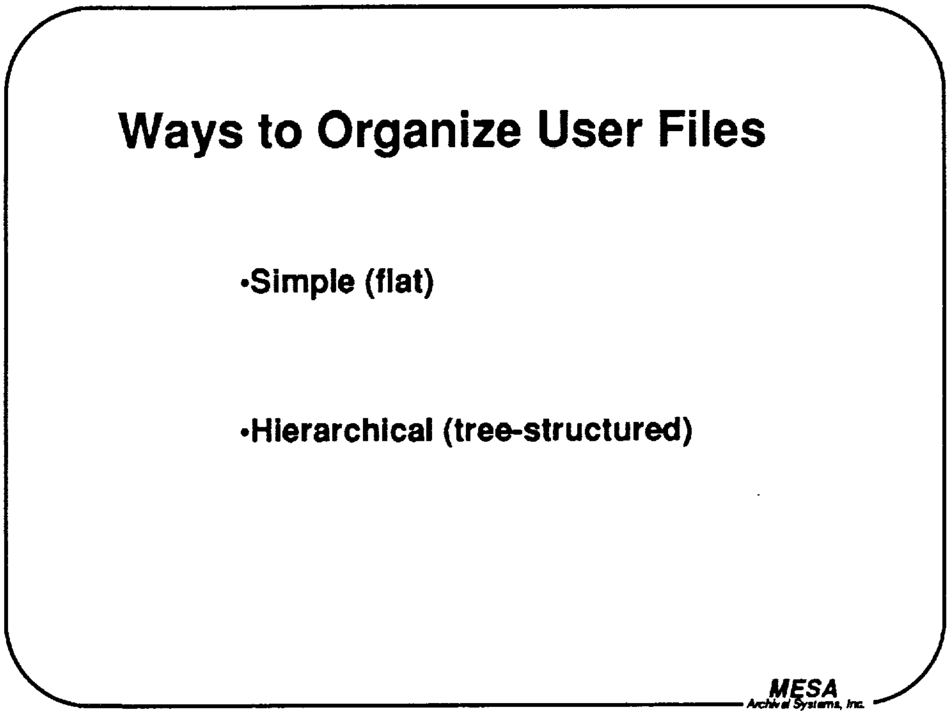
MESA Archival Systems, Inc.



Other User Capabilities

- **Add** Adds a DLS directory
- **Delete** Deletes a DLS file or directory
- **Copy** Creates a copy of a DLS file within the DLS
- **Help** Invokes DLS help facility

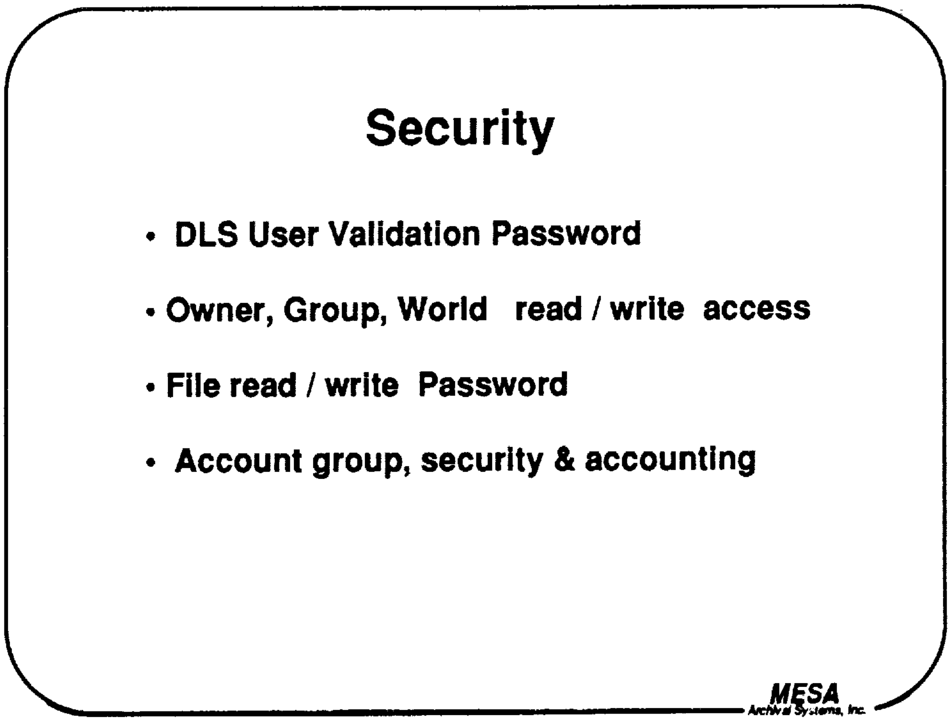
MESA Archival Systems, Inc.



Ways to Organize User Files

- Simple (flat)
- Hierarchical (tree-structured)

MESA Archival Systems, Inc.



Security

- DLS User Validation Password
- Owner, Group, World read / write access
- File read / write Password
- Account group, security & accounting

MESA Archival Systems, Inc.

Multiple Mass Storage Devices

Differing User Storage Needs

- Access Speed
- Permanent Retention
- Cost/MB
- Interchangeability

Continuing Product Evolution

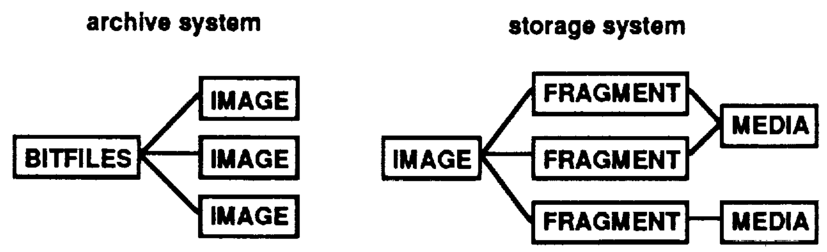
- 3480 ' 3490

New Mass Storage Devices

- D1/D2
- Optical tape

MESA Archival Systems, Inc.

Archive / Storage System

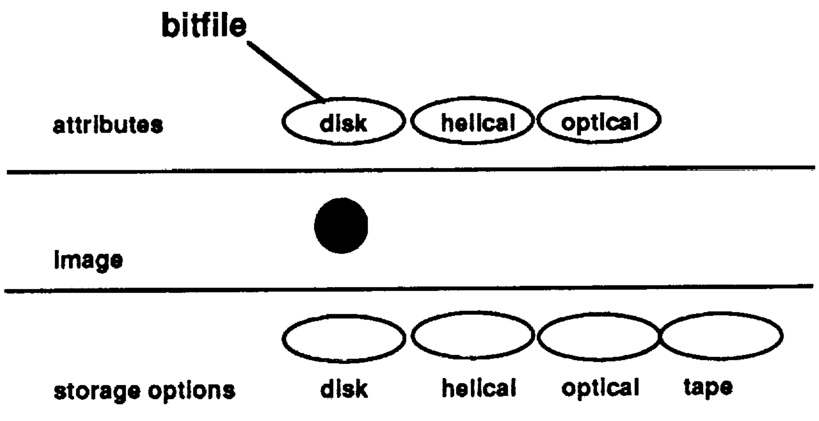


MESA Archival Systems, Inc.

Constructed Hierarchy

- Multiple views
- Dynamic
- Vary by bitfile or user

MESA
Archival Systems, Inc.

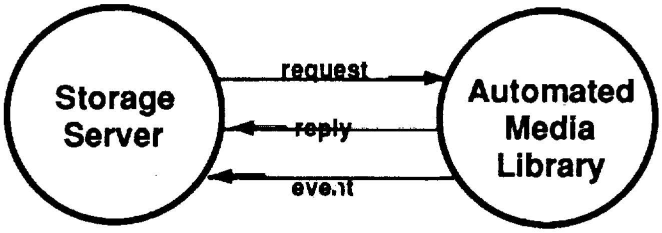


MESA
Archival Systems, Inc.

Robotic Challenge

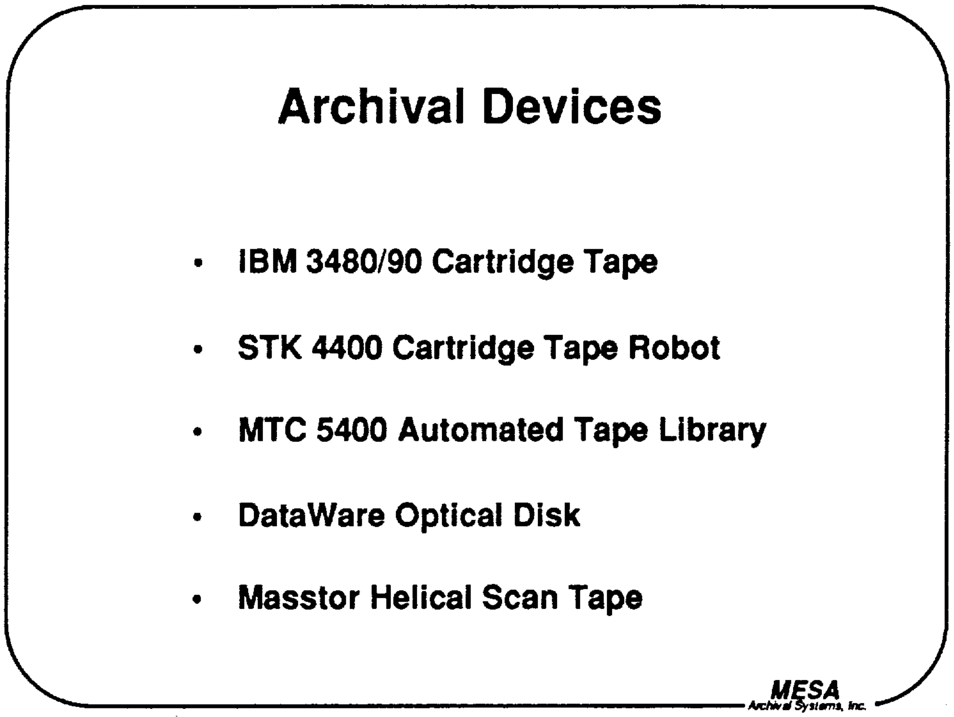
- Multiple / shared robotic systems
- 3480 service interface not sufficient
- No standard programmatic interface
- Client / server protocol recommended
  (ISO managed object)

MESA Archival Systems, Inc.



package	injected - in the robot injectable - out of the robot
surface	mounted - in a drive mountable - out of a drive
media	opened - has an active file openable - has no active file

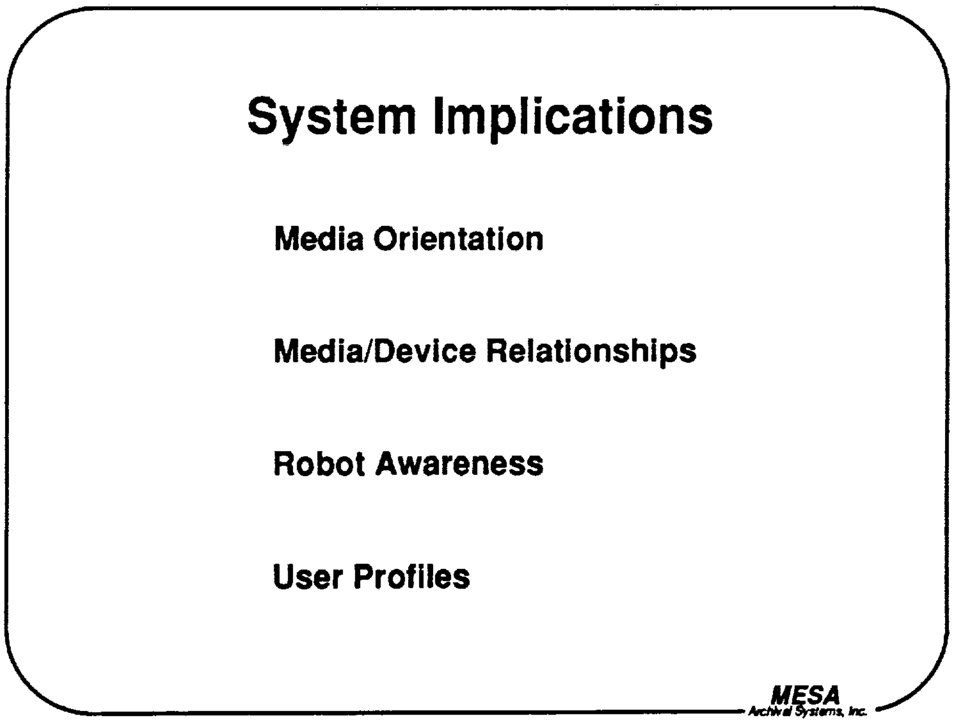
MESA Archival Systems, Inc.



Archival Devices

- IBM 3480/90 Cartridge Tape
- STK 4400 Cartridge Tape Robot
- MTC 5400 Automated Tape Library
- DataWare Optical Disk
- Masstor Helical Scan Tape

MESA Archival Systems, Inc.



System Implications

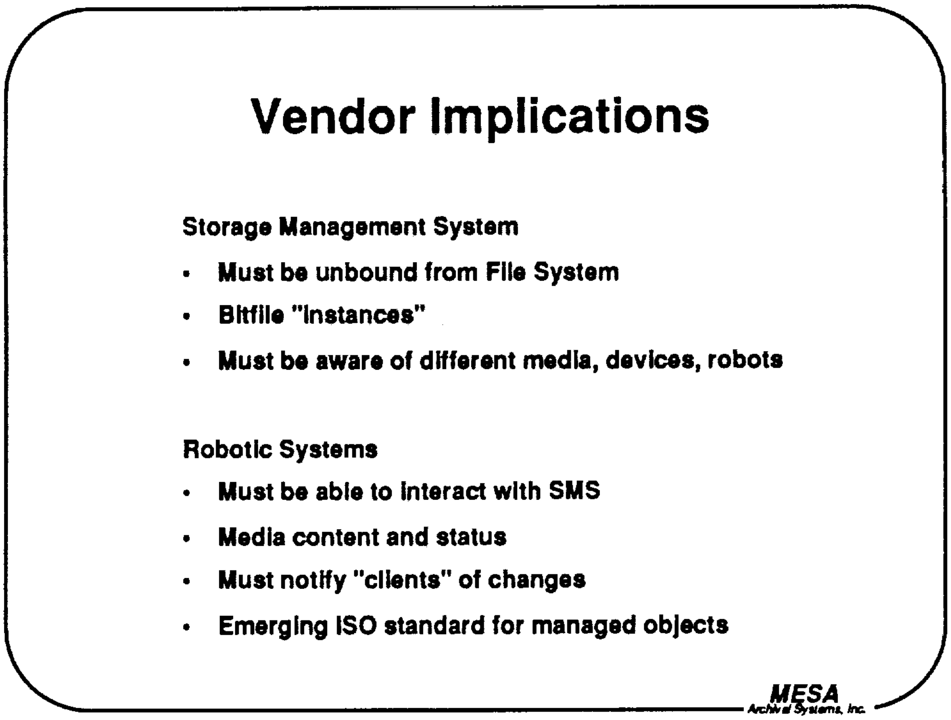
Media Orientation

Media/Device Relationships

Robot Awareness

User Profiles

MESA Archival Systems, Inc.



Vendor Implications

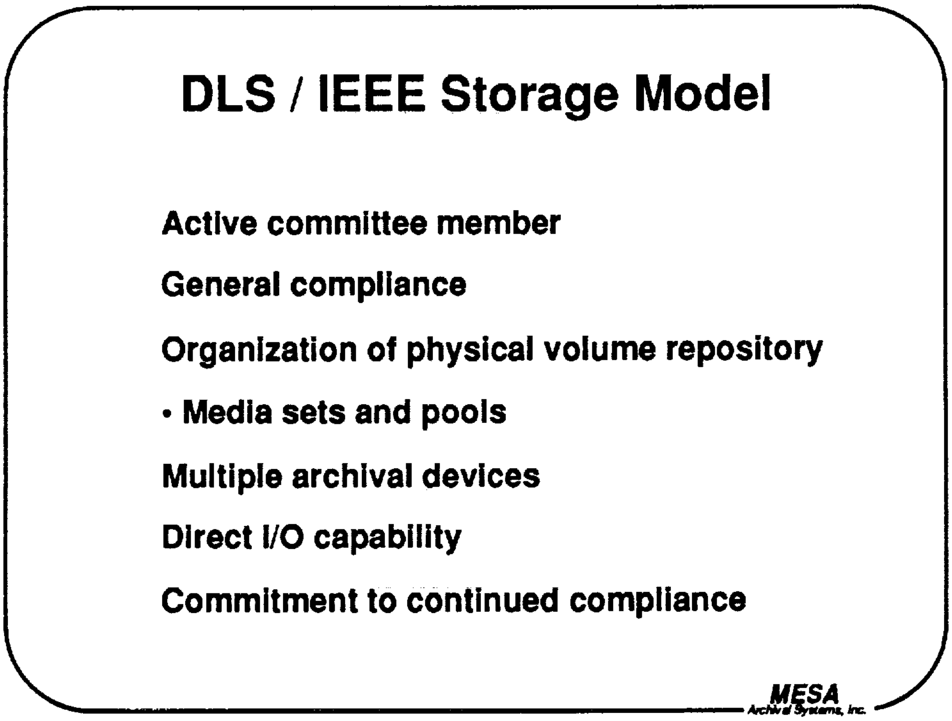
Storage Management System

- Must be unbound from File System
- Bitfile "instances"
- Must be aware of different media, devices, robots

Robotic Systems

- Must be able to interact with SMS
- Media content and status
- Must notify "clients" of changes
- Emerging ISO standard for managed objects

MESA Archival Systems, Inc.



DLS / IEEE Storage Model

Active committee member

General compliance

Organization of physical volume repository

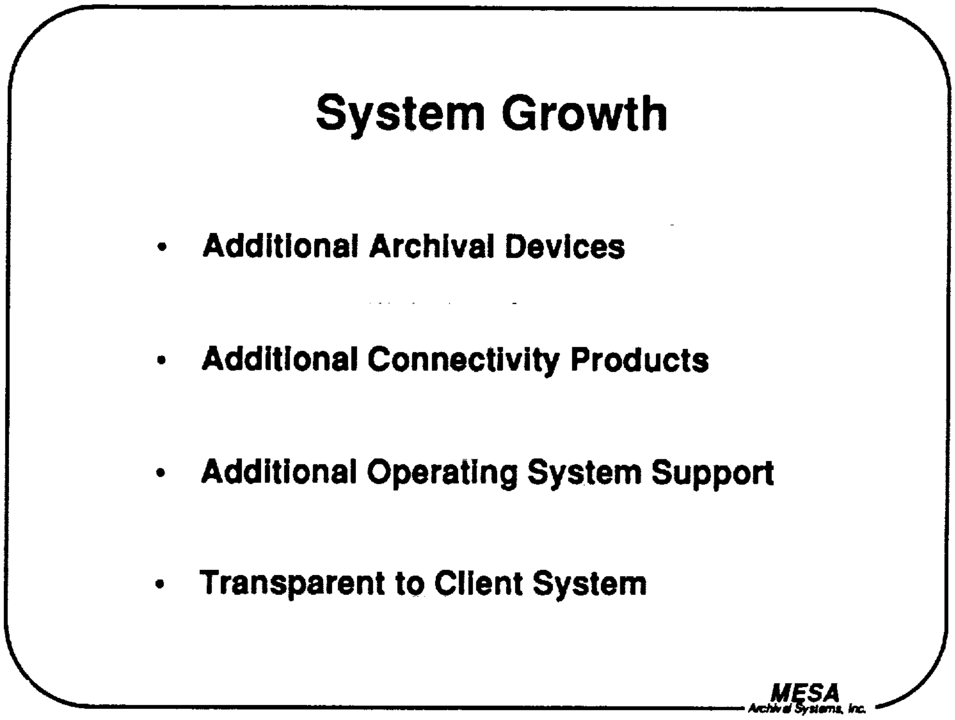
- Media sets and pools

Multiple archival devices

Direct I/O capability

Commitment to continued compliance

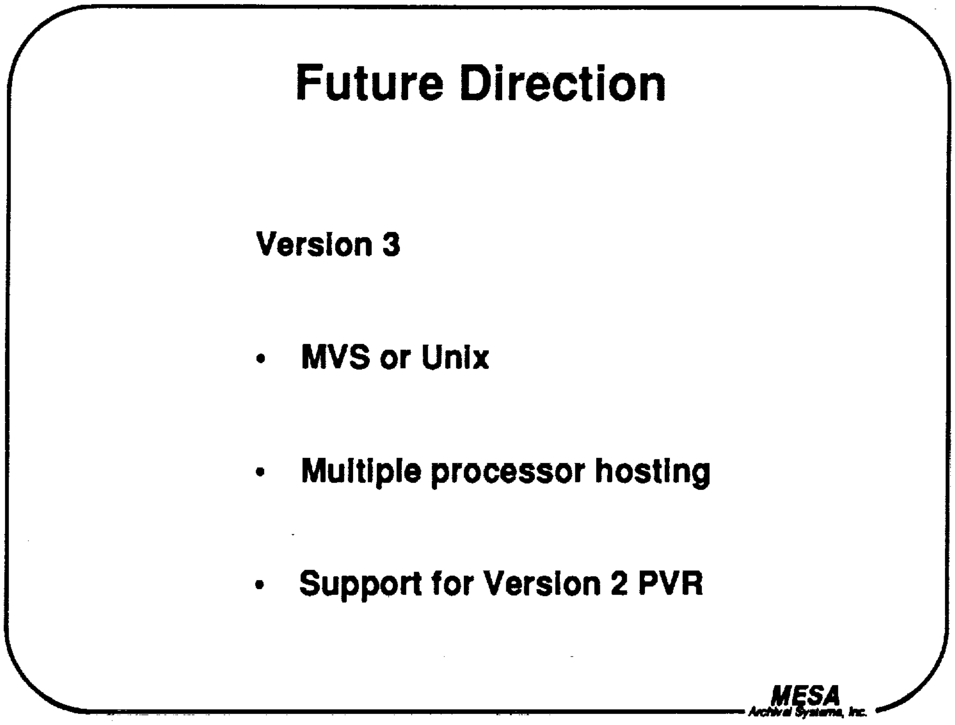
MESA Archival Systems, Inc.



System Growth

- Additional Archival Devices
- Additional Connectivity Products
- Additional Operating System Support
- Transparent to Client System

MESA Archival Systems, Inc.

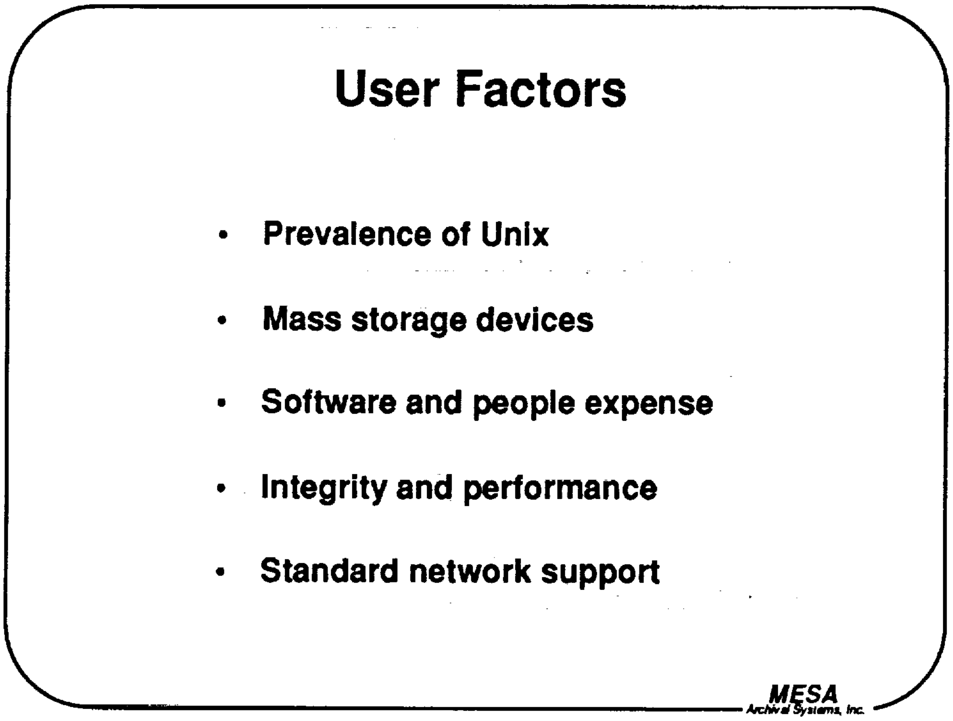


Future Direction

Version 3

- MVS or Unix
- Multiple processor hosting
- Support for Version 2 PVR

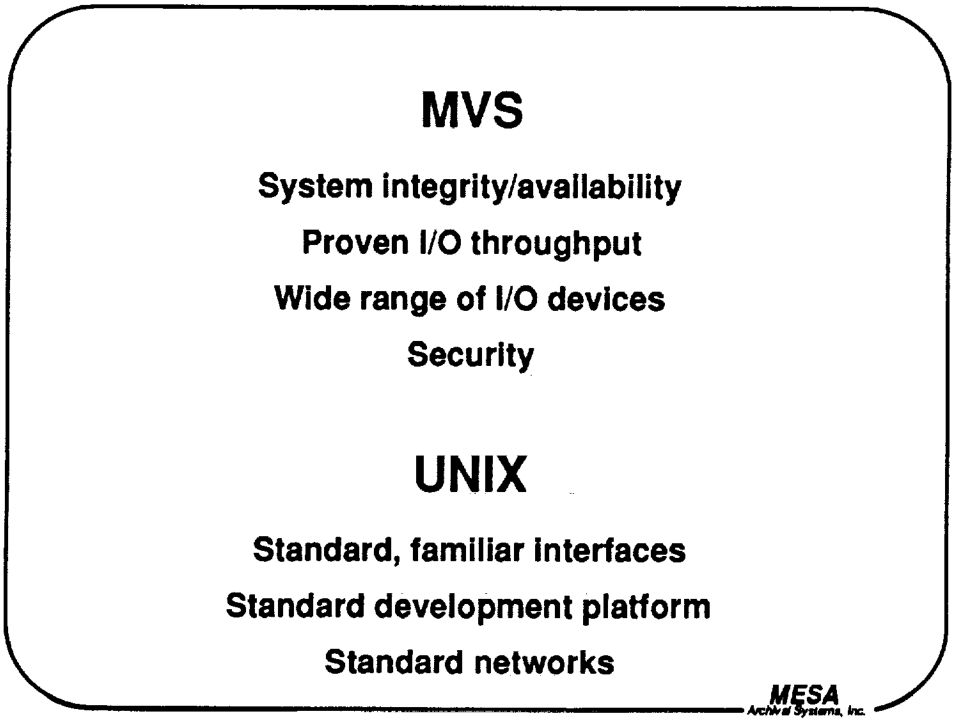
MESA Archival Systems, Inc.



User Factors

- Prevalence of Unix
- Mass storage devices
- Software and people expense
- Integrity and performance
- Standard network support

MESA Archival Systems, Inc.



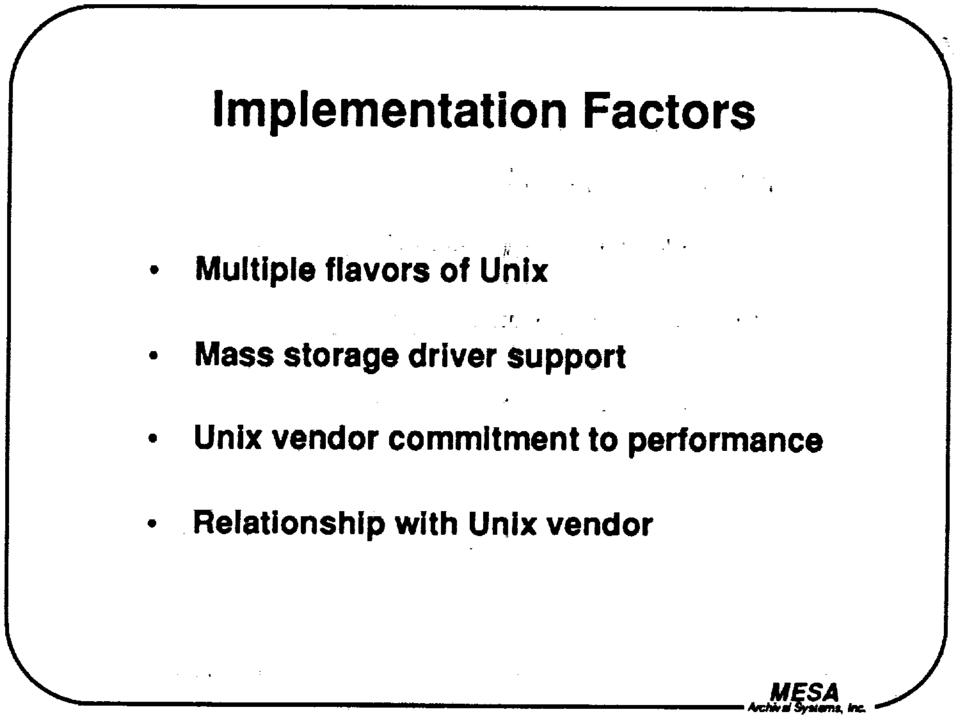
MVS

System integrity/availability
Proven I/O throughput
Wide range of I/O devices
Security

UNIX

Standard, familiar interfaces
Standard development platform
Standard networks

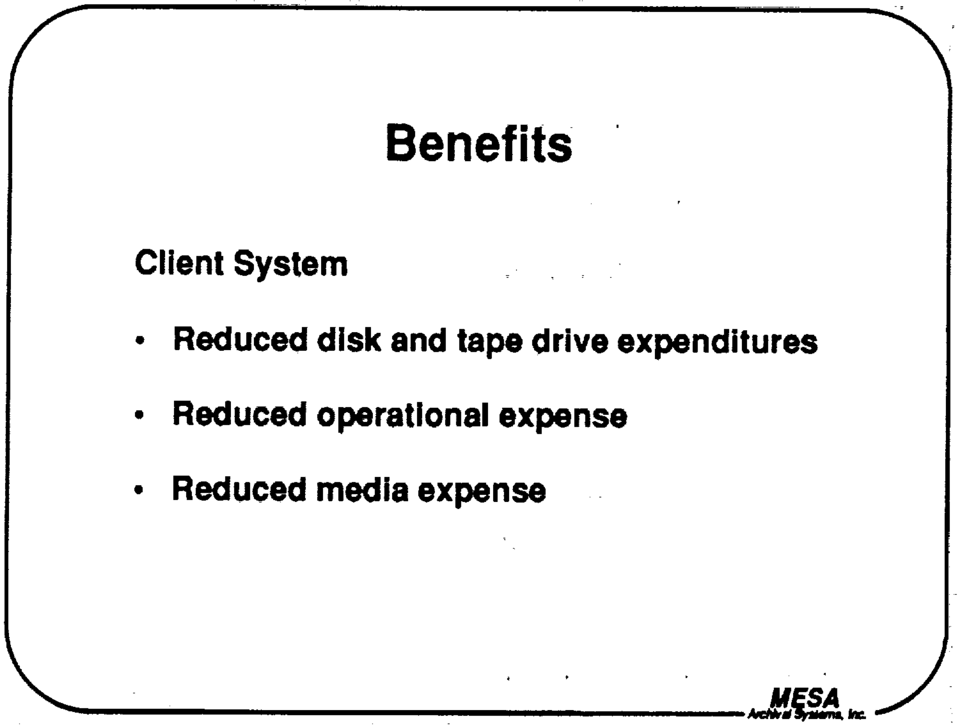
MESA Archival Systems, Inc.



Implementation Factors

- Multiple flavors of Unix
- Mass storage driver support
- Unix vendor commitment to performance
- Relationship with Unix vendor

MESA
Archival Systems, Inc.

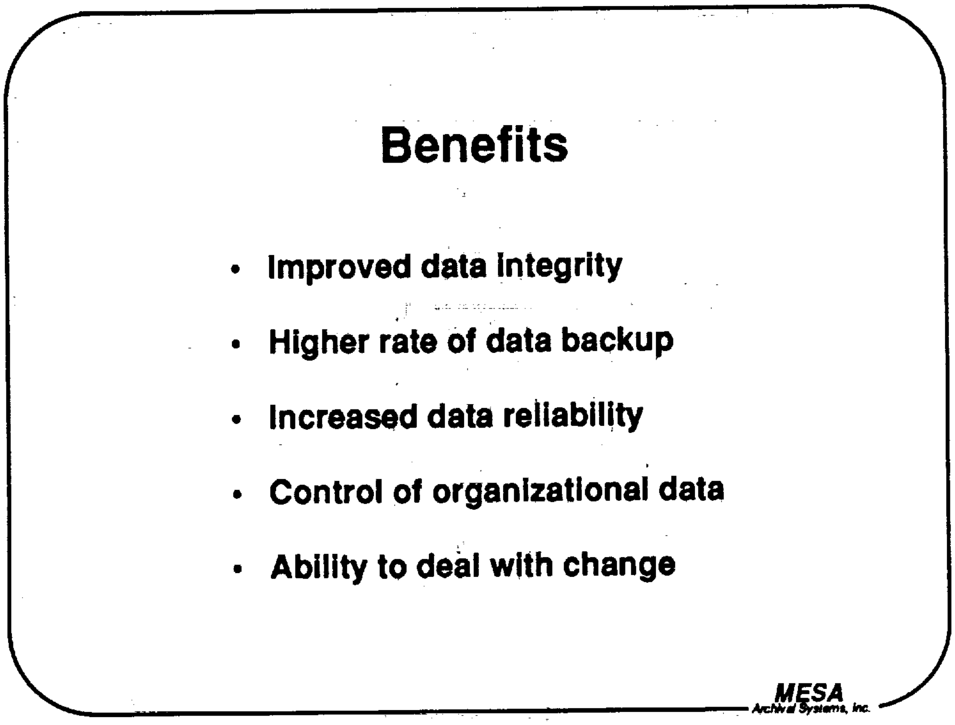


Benefits

Client System

- Reduced disk and tape drive expenditures
- Reduced operational expense
- Reduced media expense

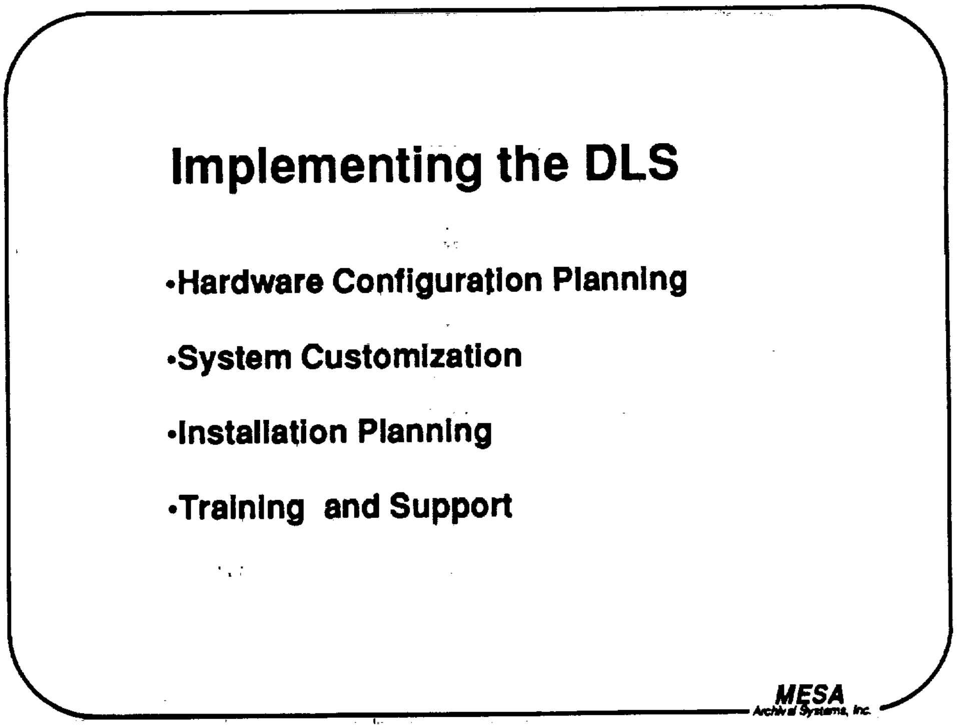
MESA
Archival Systems, Inc.



Benefits

- Improved data integrity
- Higher rate of data backup
- Increased data reliability
- Control of organizational data
- Ability to deal with change

MESA Archival Systems, Inc.



Implementing the DLS

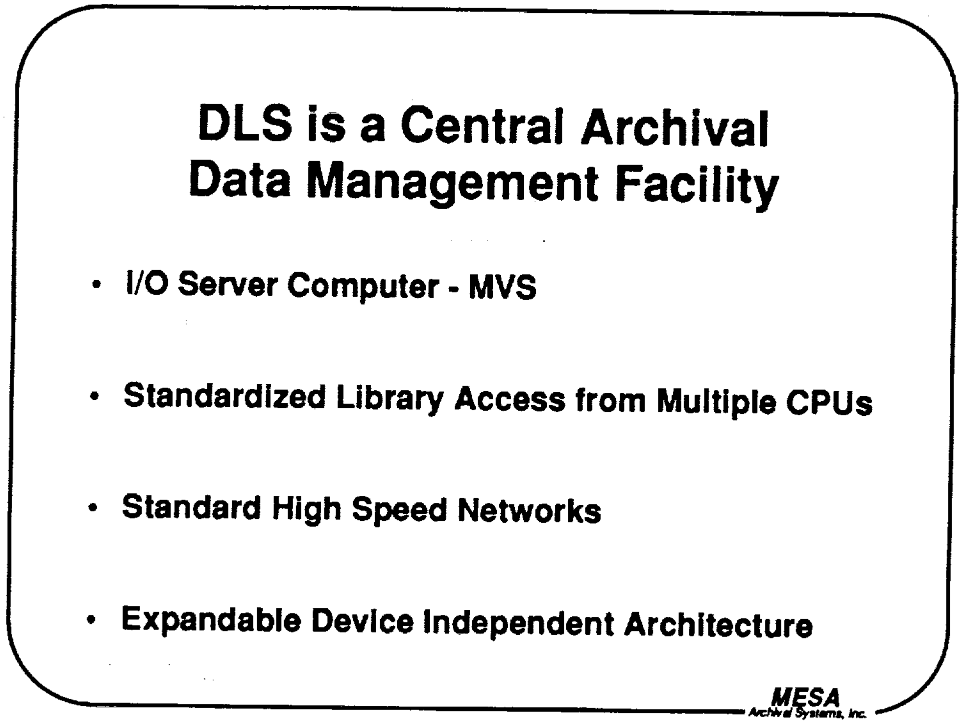
•Hardware Configuration Planning

•System Customization

•Installation Planning

•Training and Support

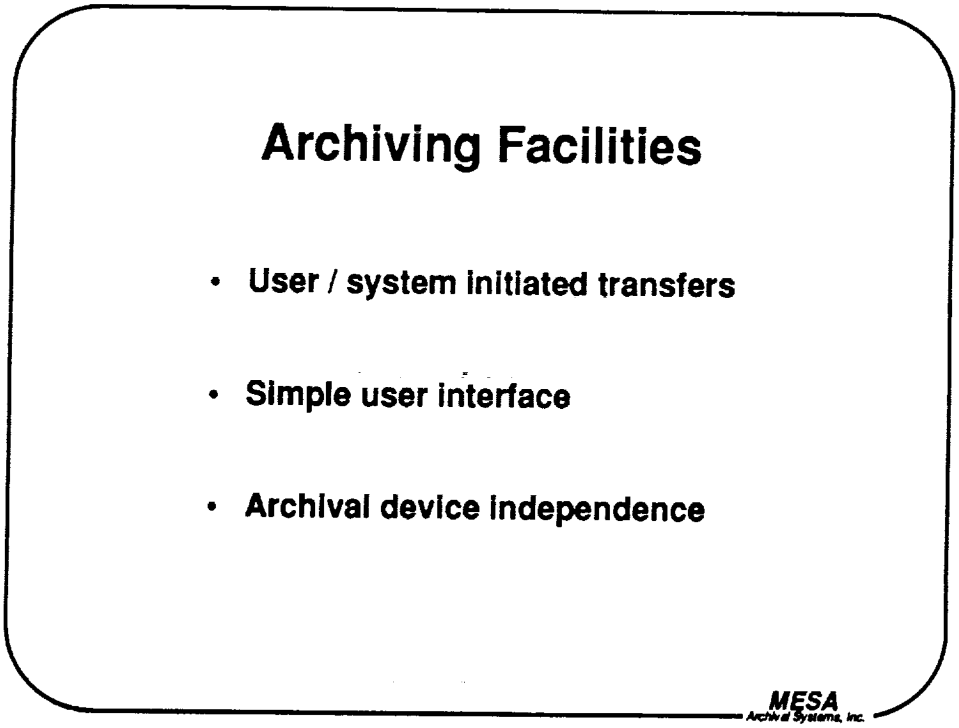
MESA Archival Systems, Inc.



DLS is a Central Archival Data Management Facility

- I/O Server Computer - MVS
- Standardized Library Access from Multiple CPUs
- Standard High Speed Networks
- Expandable Device Independent Architecture

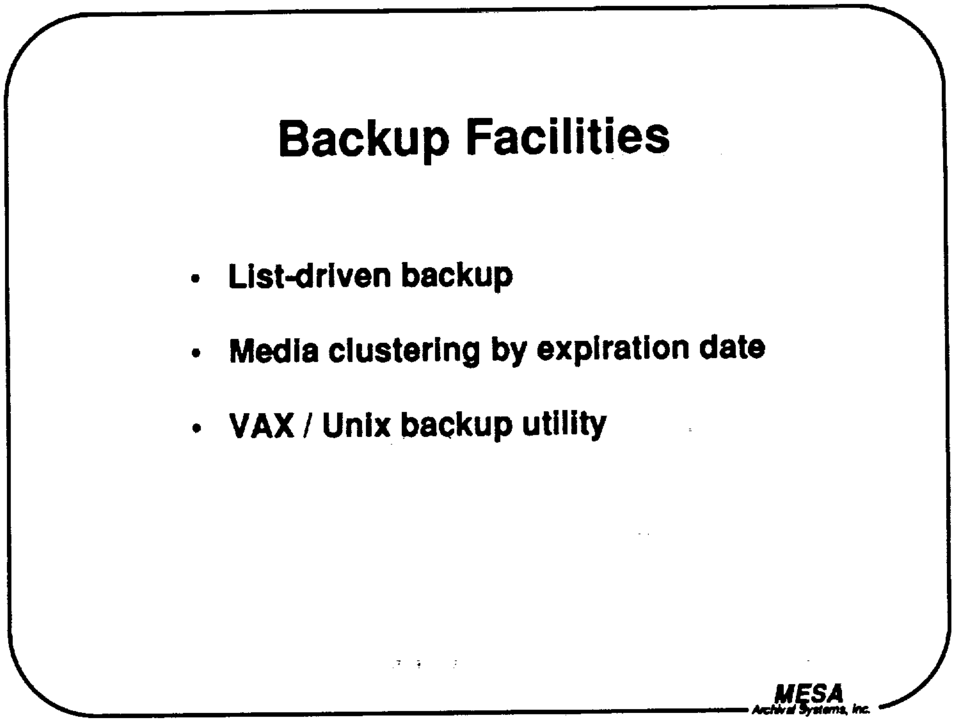
MESA Archival Systems, Inc.



Archiving Facilities

- User / system initiated transfers
- Simple user interface
- Archival device independence

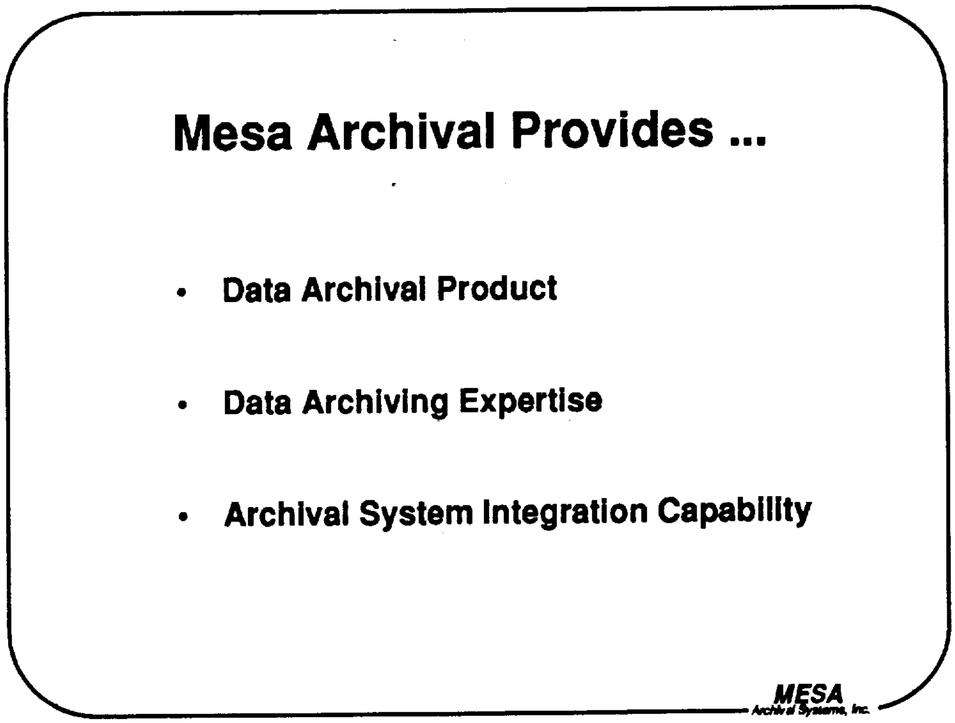
MESA Archival Systems, Inc.



Backup Facilities

- List-driven backup
- Media clustering by expiration date
- VAX / Unix backup utility

MESA Archival Systems, Inc.



Mesa Archival Provides ...

- Data Archival Product
- Data Archiving Expertise
- Archival System Integration Capability

MESA Archival Systems, Inc.