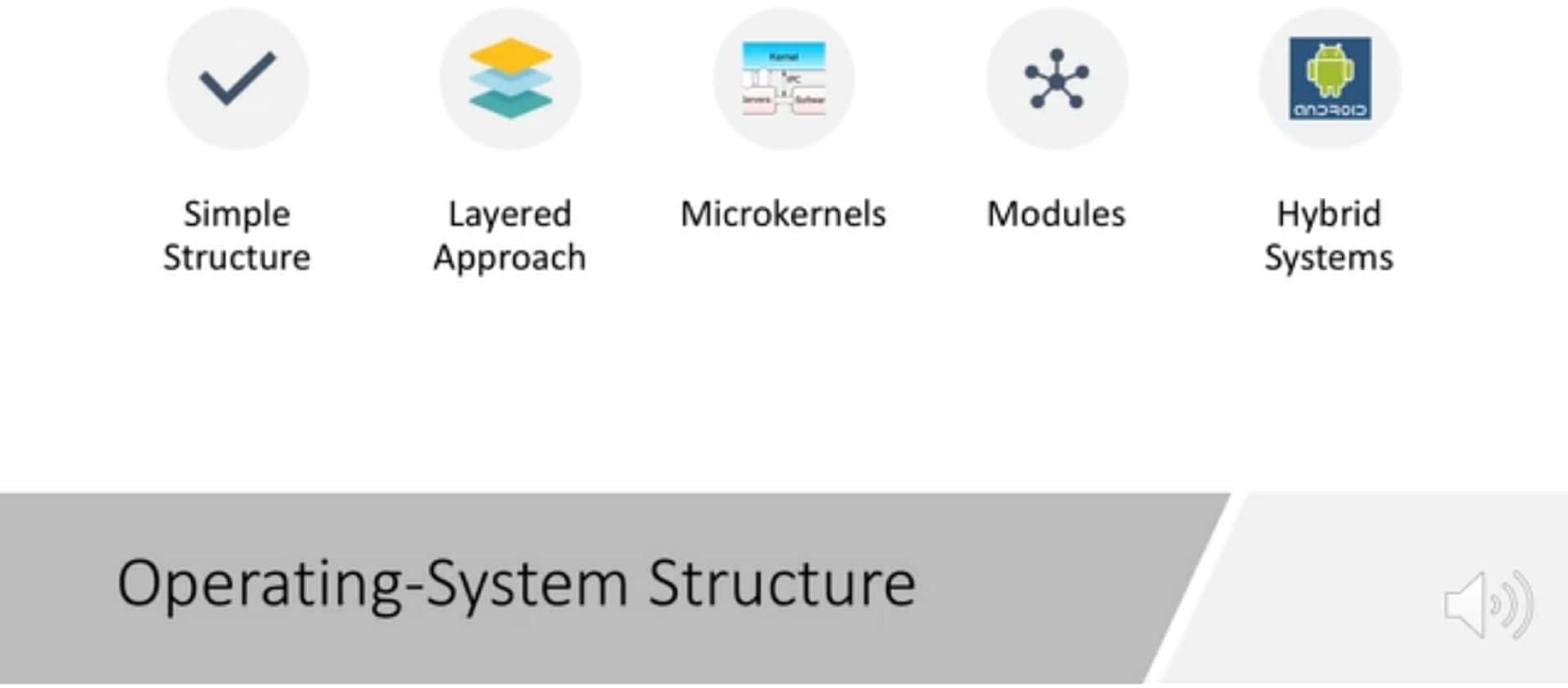
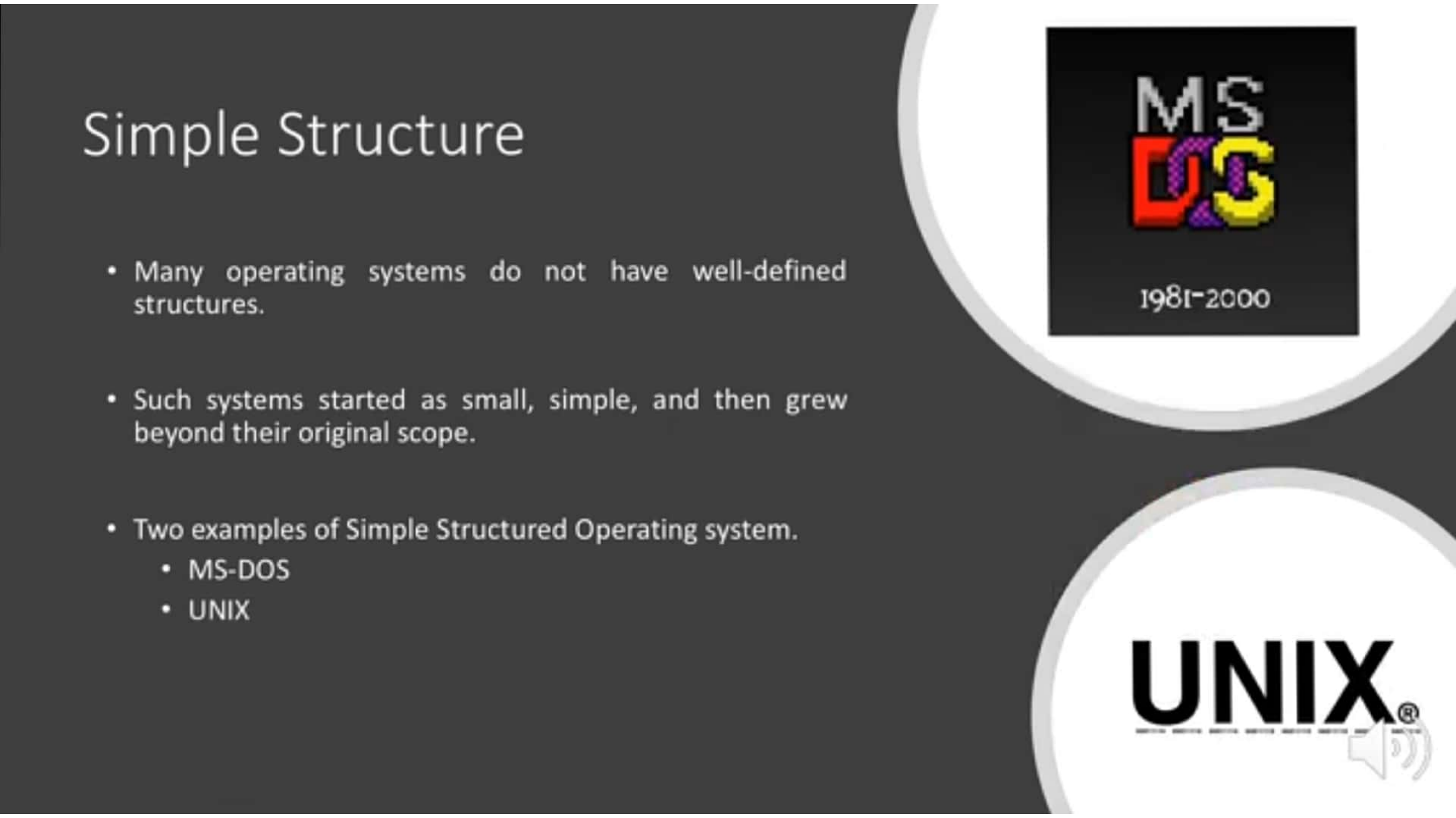


- Simple Structure
- Layered Approach
- Microkernels
- Modules
- Hybrid Systems

Operating-System Structure

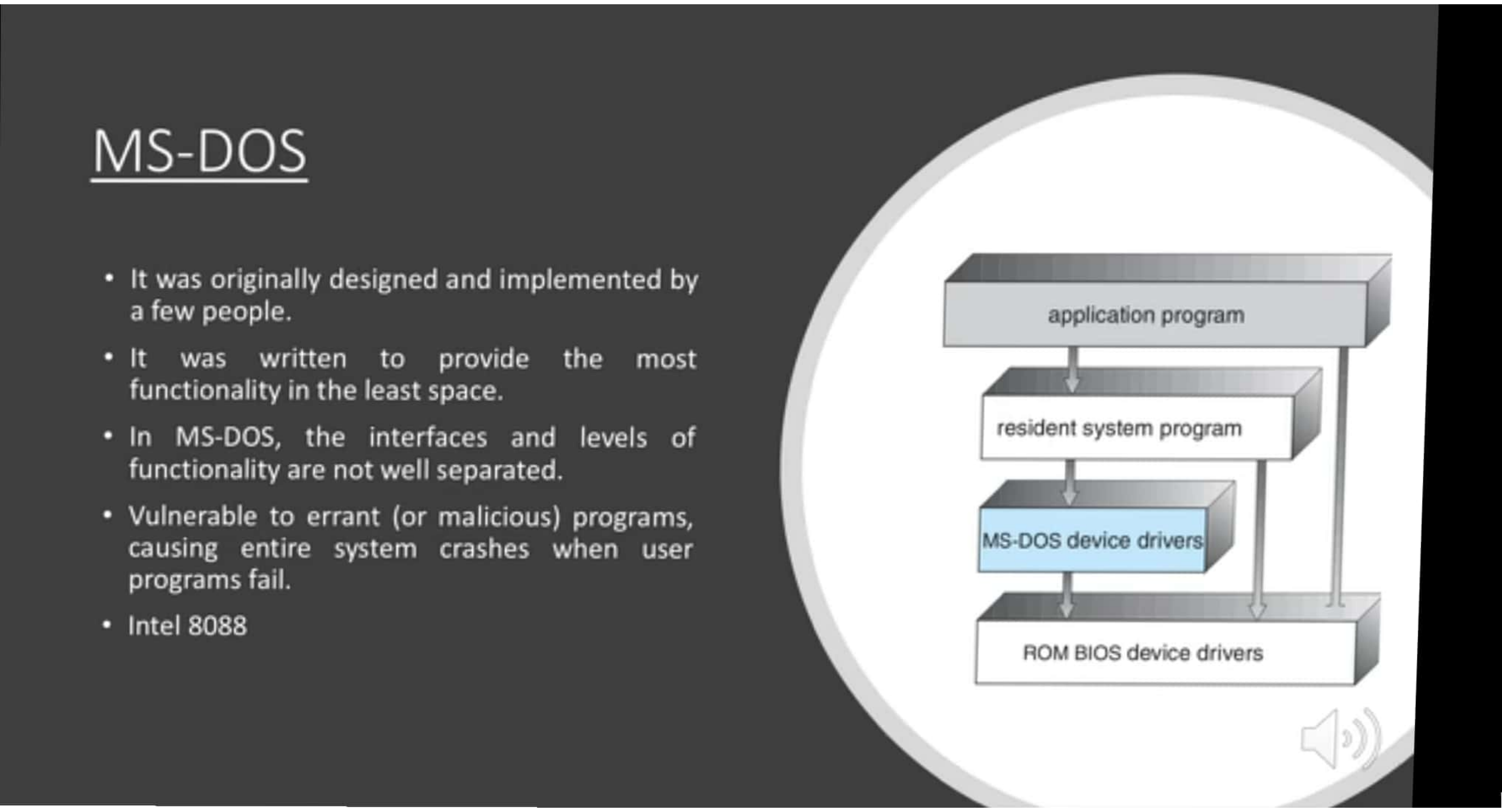
Scanned with CamScanner



Simple Structure

- Many operating systems do not have well-defined structures.
- Such systems started as small, simple, and then grew beyond their original scope.
- Two examples of Simple Structured Operating system.
  - MS-DOS
  - UNIX

Scanned with CamScanner



MS-DOS

- It was originally designed and implemented by a few people.
- It was written to provide the most functionality in the least space.
- In MS-DOS, the interfaces and levels of functionality are not well separated.
- Vulnerable to errant (or malicious) programs, causing entire system crashes when user programs fail.
- Intel 8088

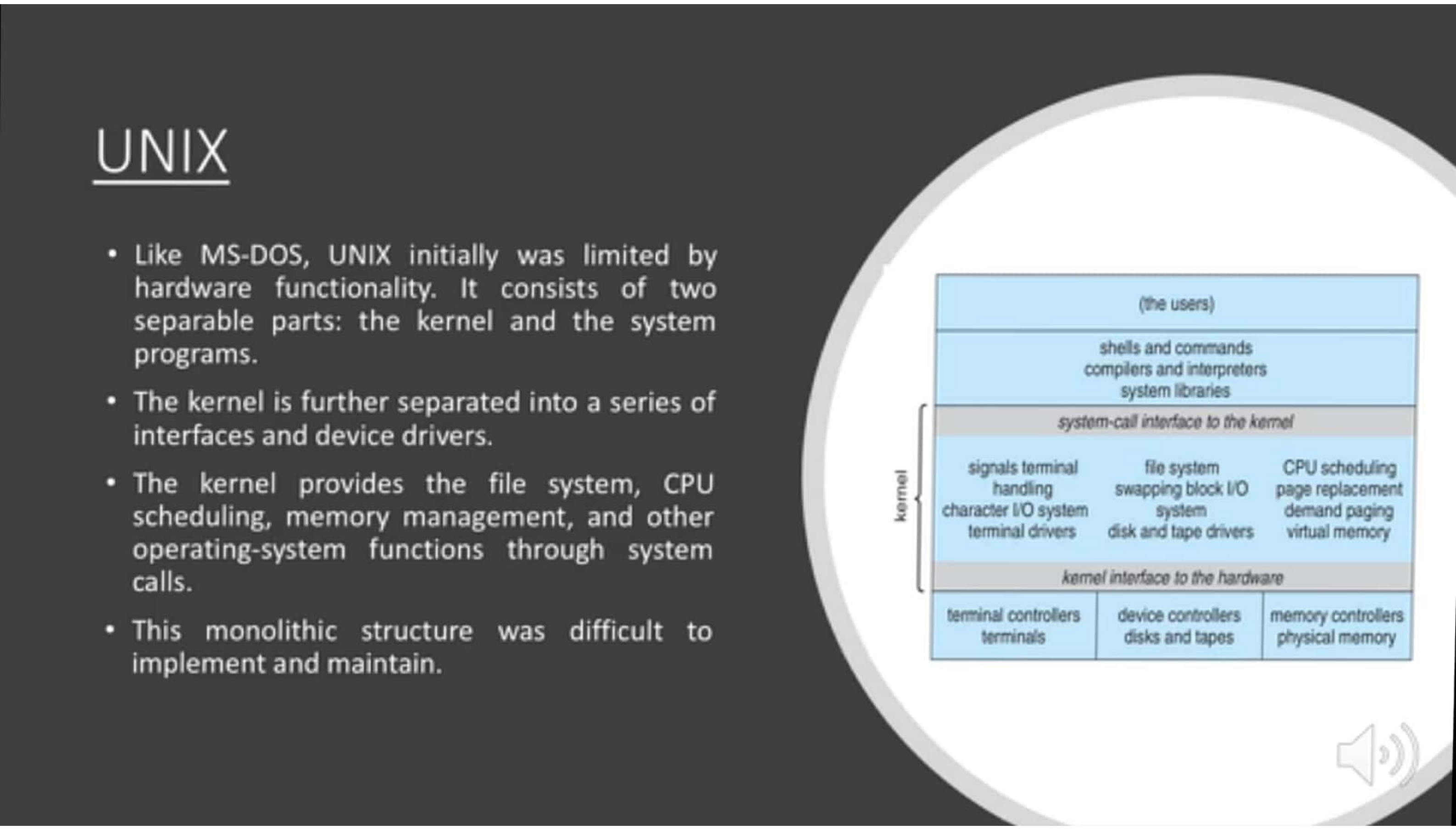
application program

resident system program

MS-DOS device drivers

ROM BIOS device drivers

Scanned with CamScanner



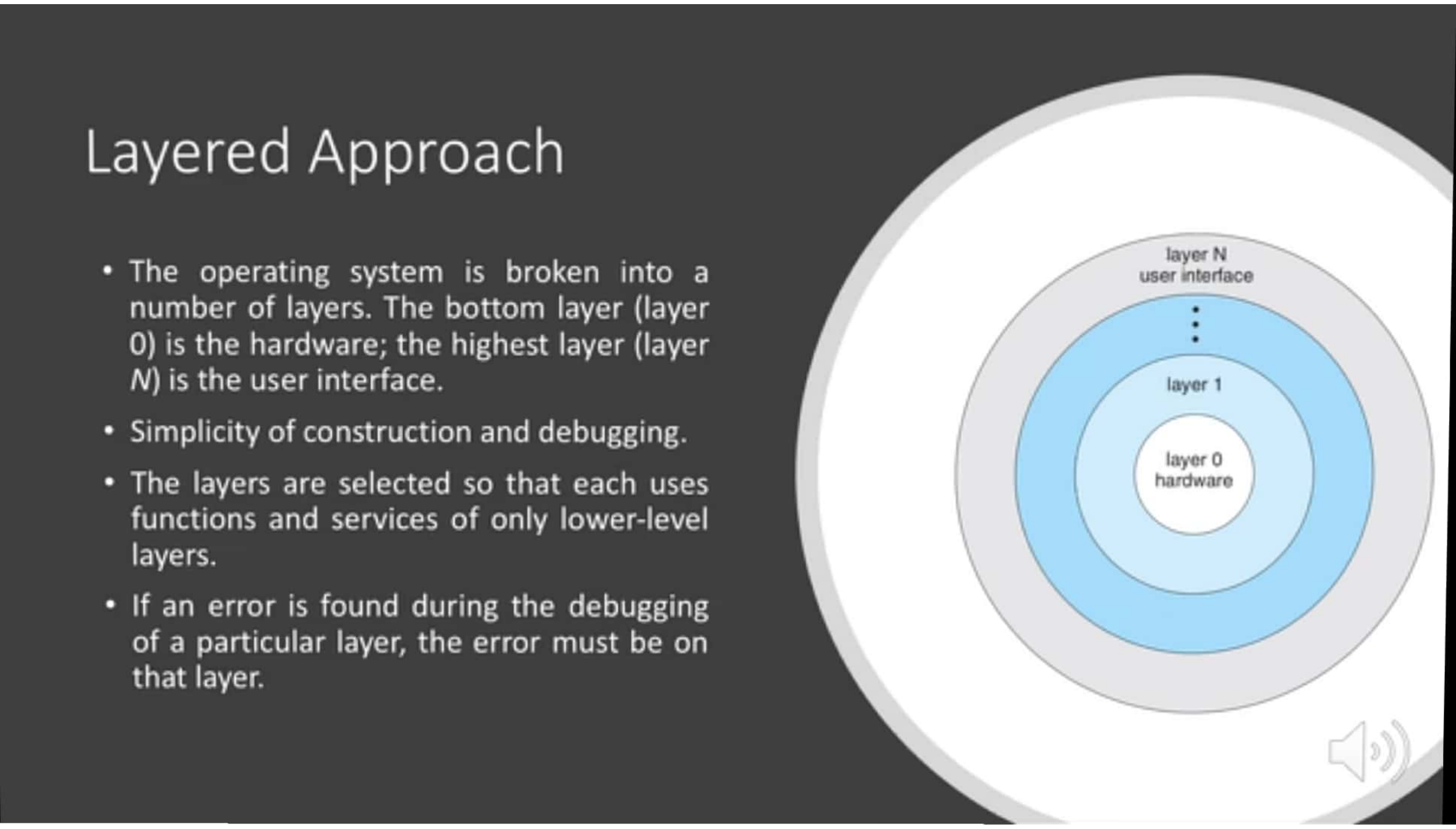
UNIX

- Like MS-DOS, UNIX initially was limited by hardware functionality. It consists of two separable parts: the kernel and the system programs.
- The kernel is further separated into a series of interfaces and device drivers.
- The kernel provides the file system, CPU scheduling, memory management, and other operating-system functions through system calls.
- This monolithic structure was difficult to implement and maintain.

kernel

(the users)		
shells and commands compilers and interpreters system libraries		
system-call interface to the kernel		
signals terminal handling character I/O system terminal drivers	file system swapping block I/O system disk and tape drivers	CPU scheduling page replacement demand paging virtual memory
kernel interface to the hardware		
terminal controllers terminals	device controllers disks and tapes	memory controllers physical memory

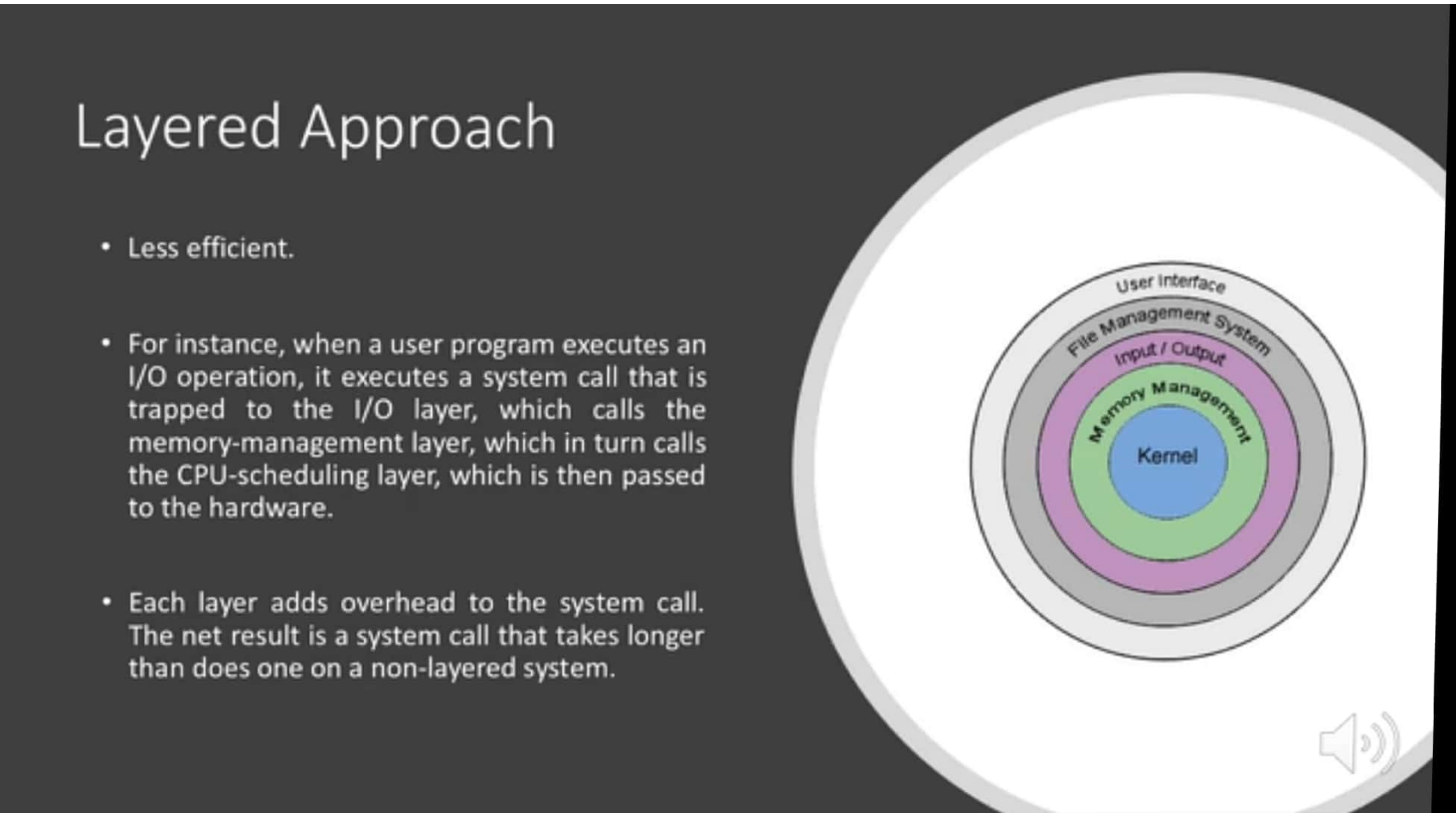
Scanned with CamScanner



Layered Approach

- The operating system is broken into a number of layers. The bottom layer (layer 0) is the hardware; the highest layer (layer *N*) is the user interface.
- Simplicity of construction and debugging.
- The layers are selected so that each uses functions and services of only lower-level layers.
- If an error is found during the debugging of a particular layer, the error must be on that layer.

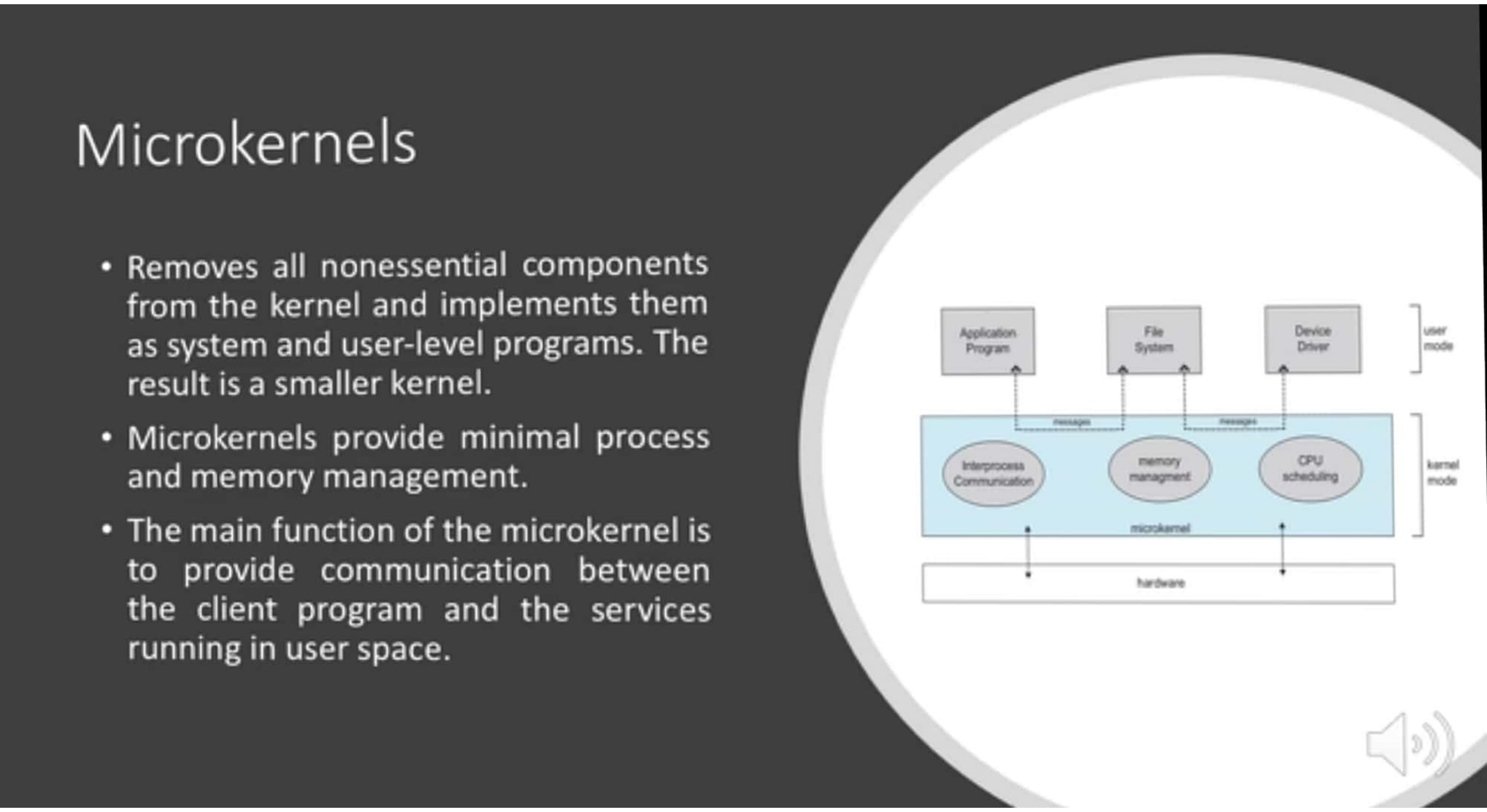
Scanned with CamScanner



Layered Approach

- Less efficient.
- For instance, when a user program executes an I/O operation, it executes a system call that is trapped to the I/O layer, which calls the memory-management layer, which in turn calls the CPU-scheduling layer, which is then passed to the hardware.
- Each layer adds overhead to the system call. The net result is a system call that takes longer than does one on a non-layered system.

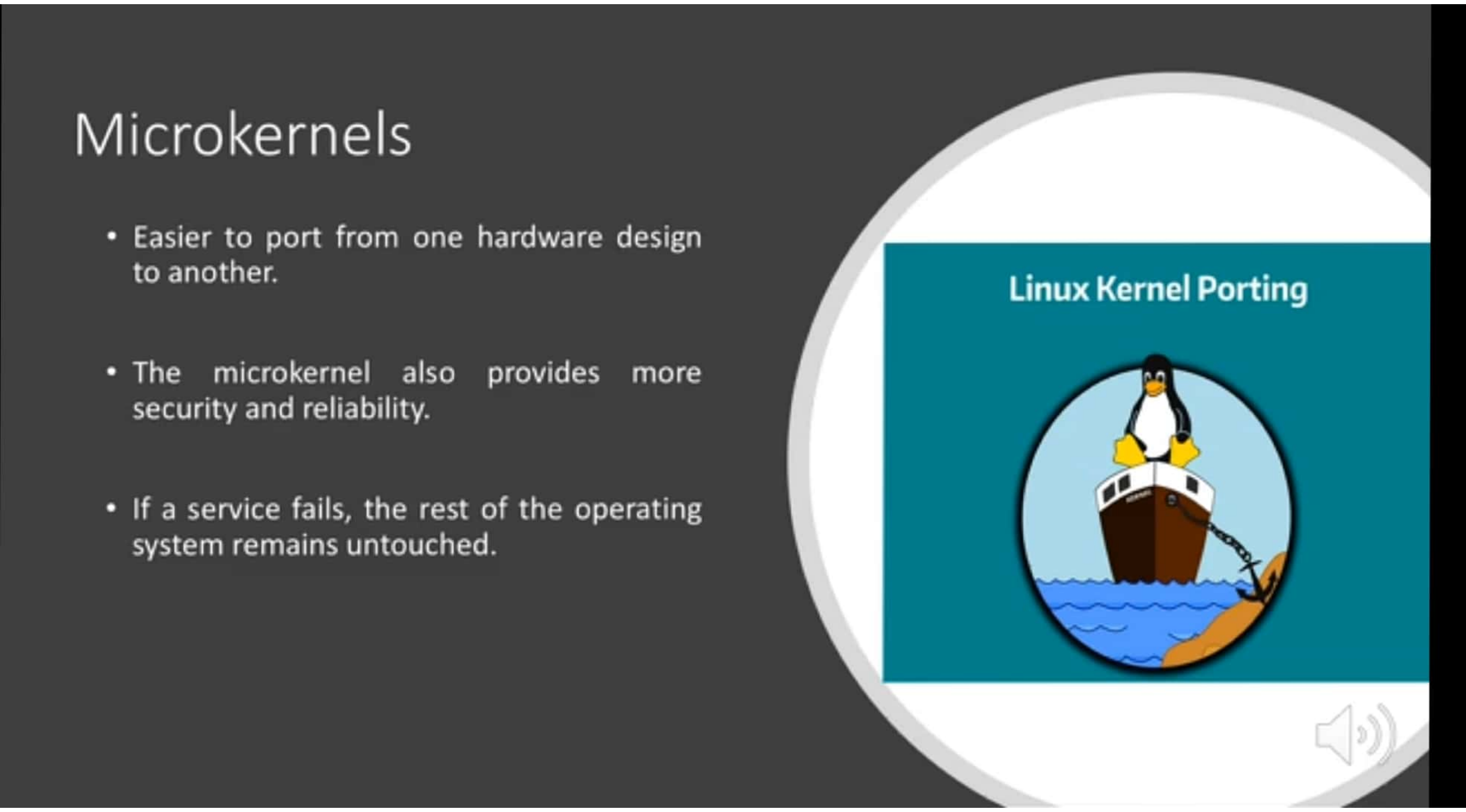
Scanned with CamScanner



Microkernels

- Removes all nonessential components from the kernel and implements them as system and user-level programs. The result is a smaller kernel.
- Microkernels provide minimal process and memory management.
- The main function of the microkernel is to provide communication between the client program and the services running in user space.

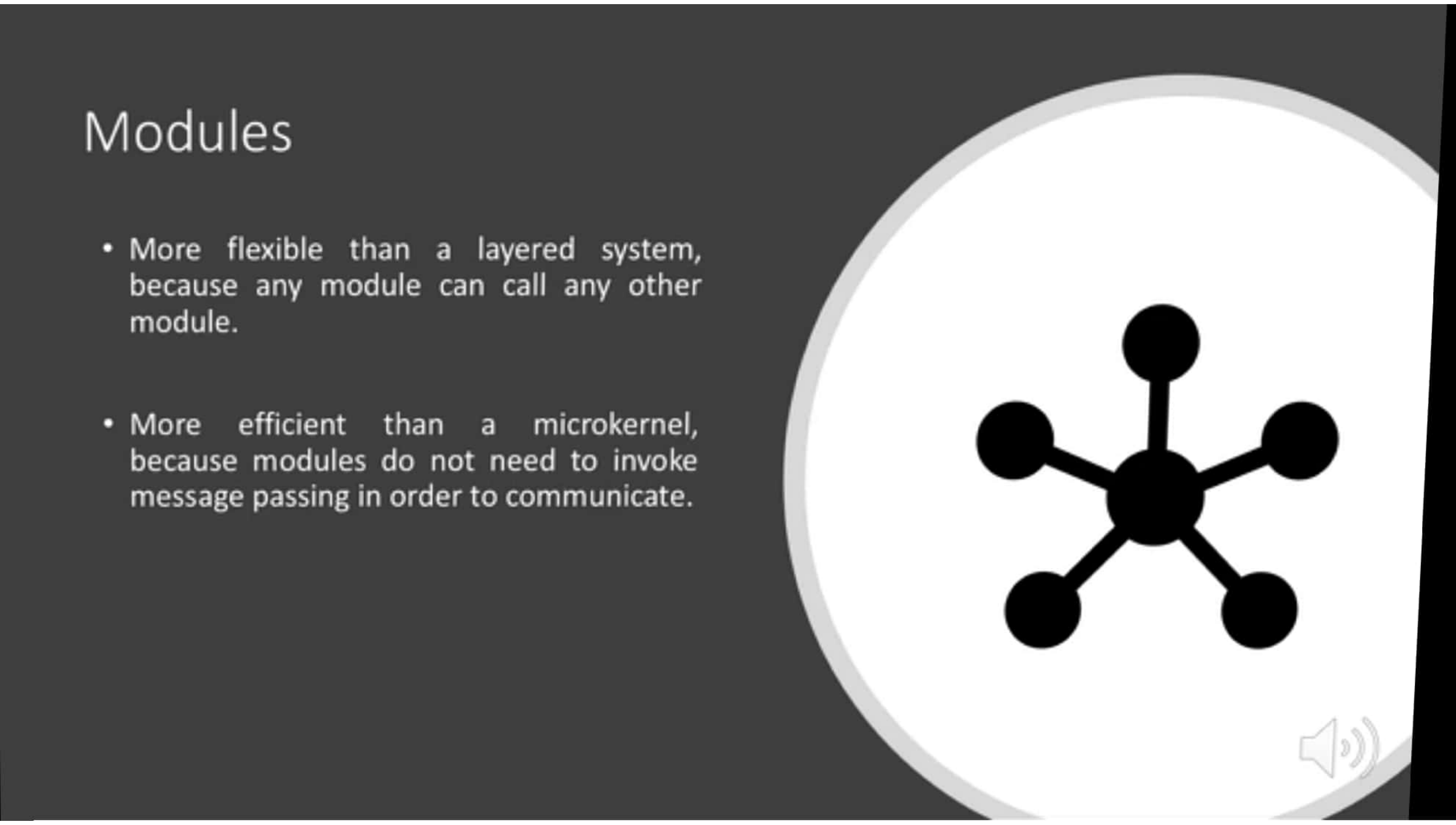
Scanned with CamScanner



Microkernels

- Easier to port from one hardware design to another.
- The microkernel also provides more security and reliability.
- If a service fails, the rest of the operating system remains untouched.

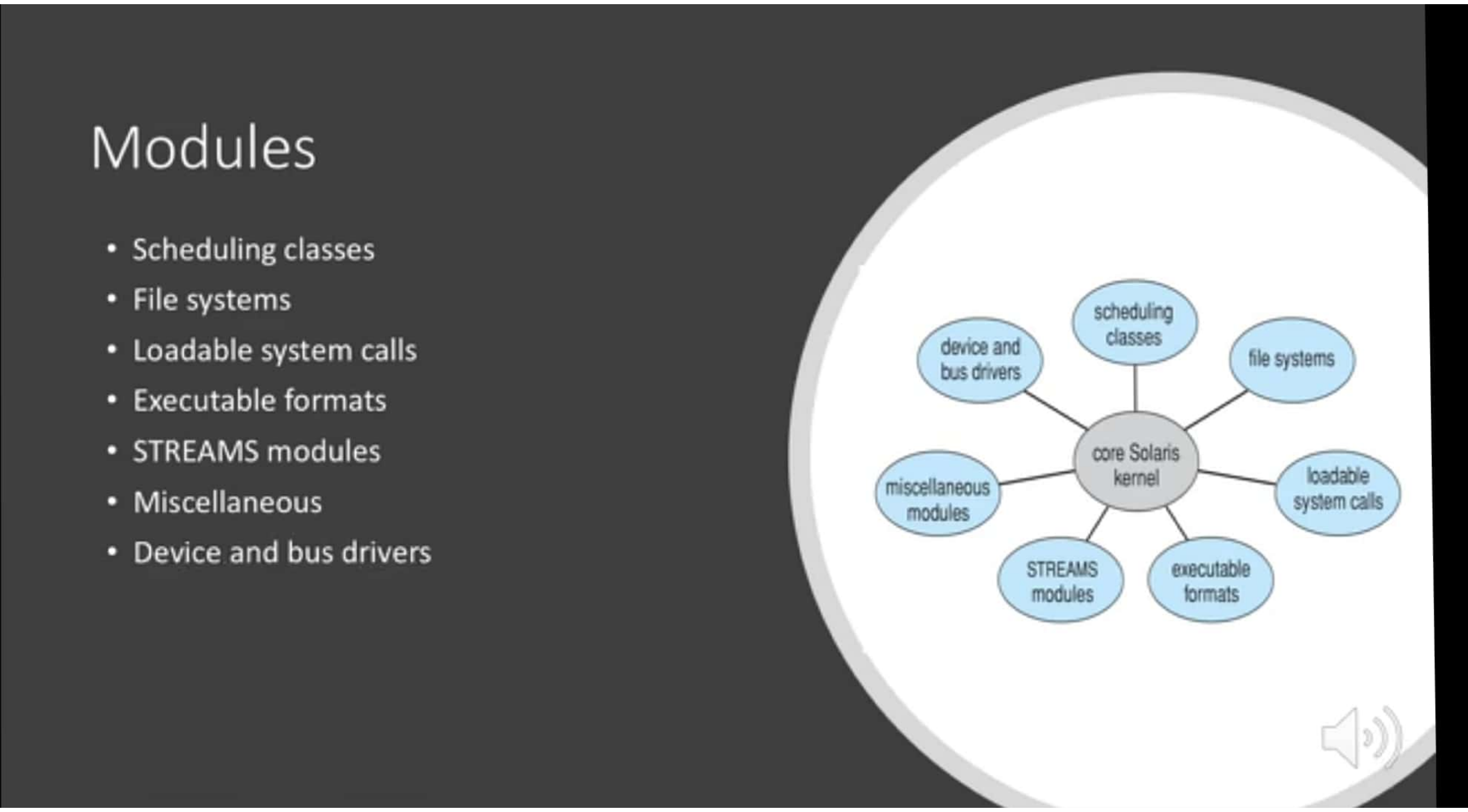
Scanned with CamScanner



Modules

- More flexible than a layered system, because any module can call any other module.
- More efficient than a microkernel, because modules do not need to invoke message passing in order to communicate.

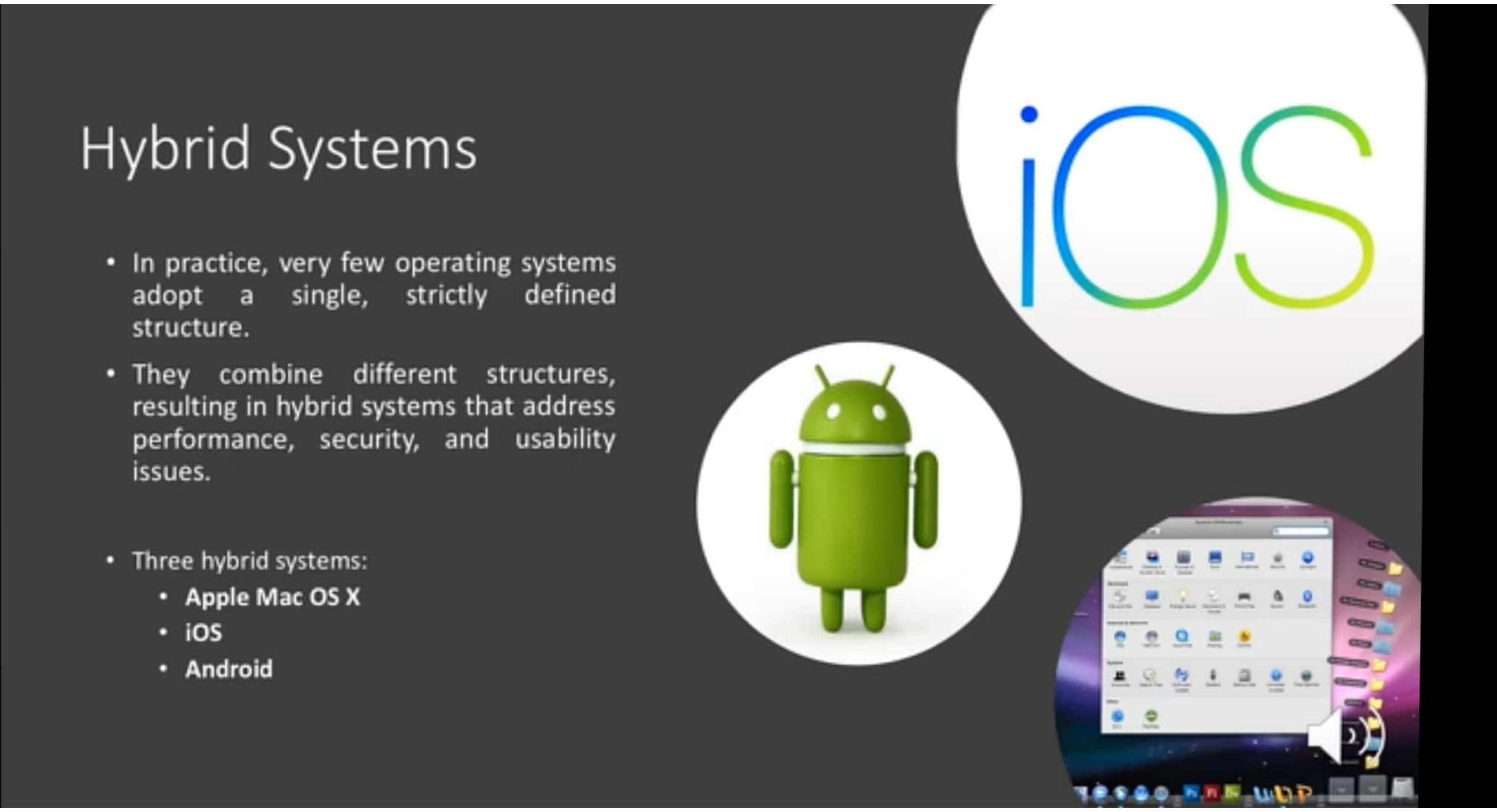
Scanned with CamScanner



Modules

- Scheduling classes
- File systems
- Loadable system calls
- Executable formats
- STREAMS modules
- Miscellaneous
- Device and bus drivers

Scanned with CamScanner



Hybrid Systems

- In practice, very few operating systems adopt a single, strictly defined structure.
- They combine different structures, resulting in hybrid systems that address performance, security, and usability issues.
- Three hybrid systems:
  - **Apple Mac OS X**
  - **iOS**
  - **Android**

Scanned with CamScanner