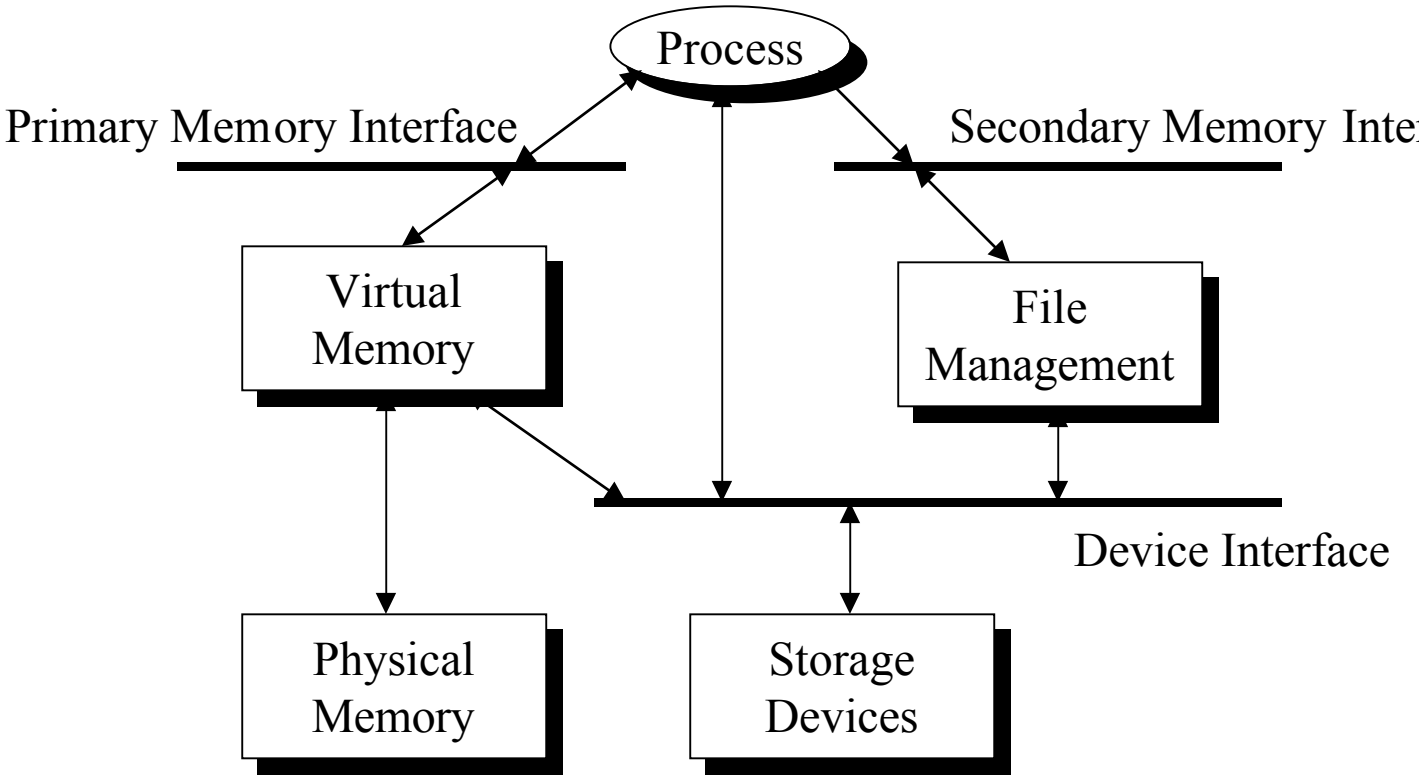
Remote Files

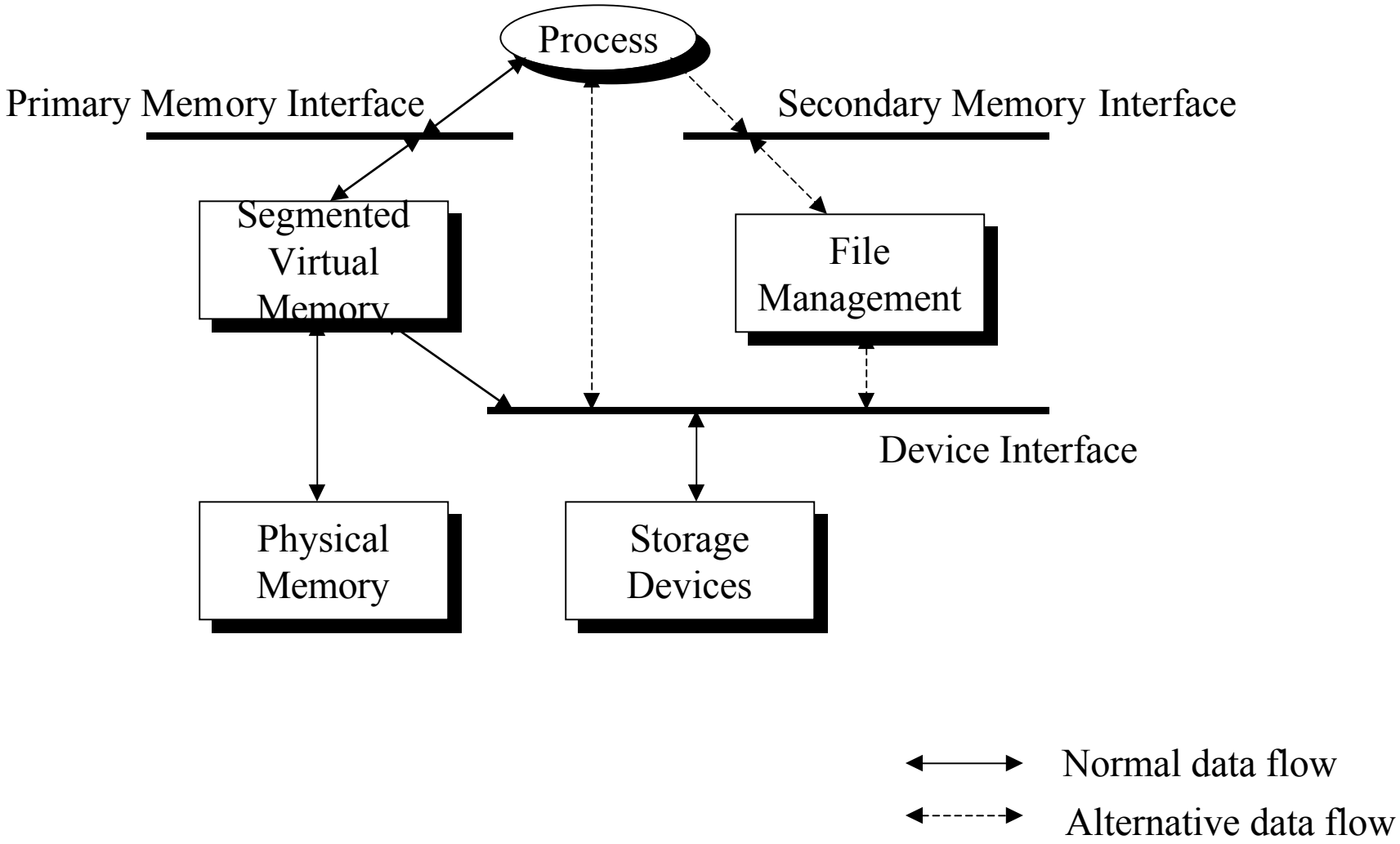
Traditional Memory Interfaces



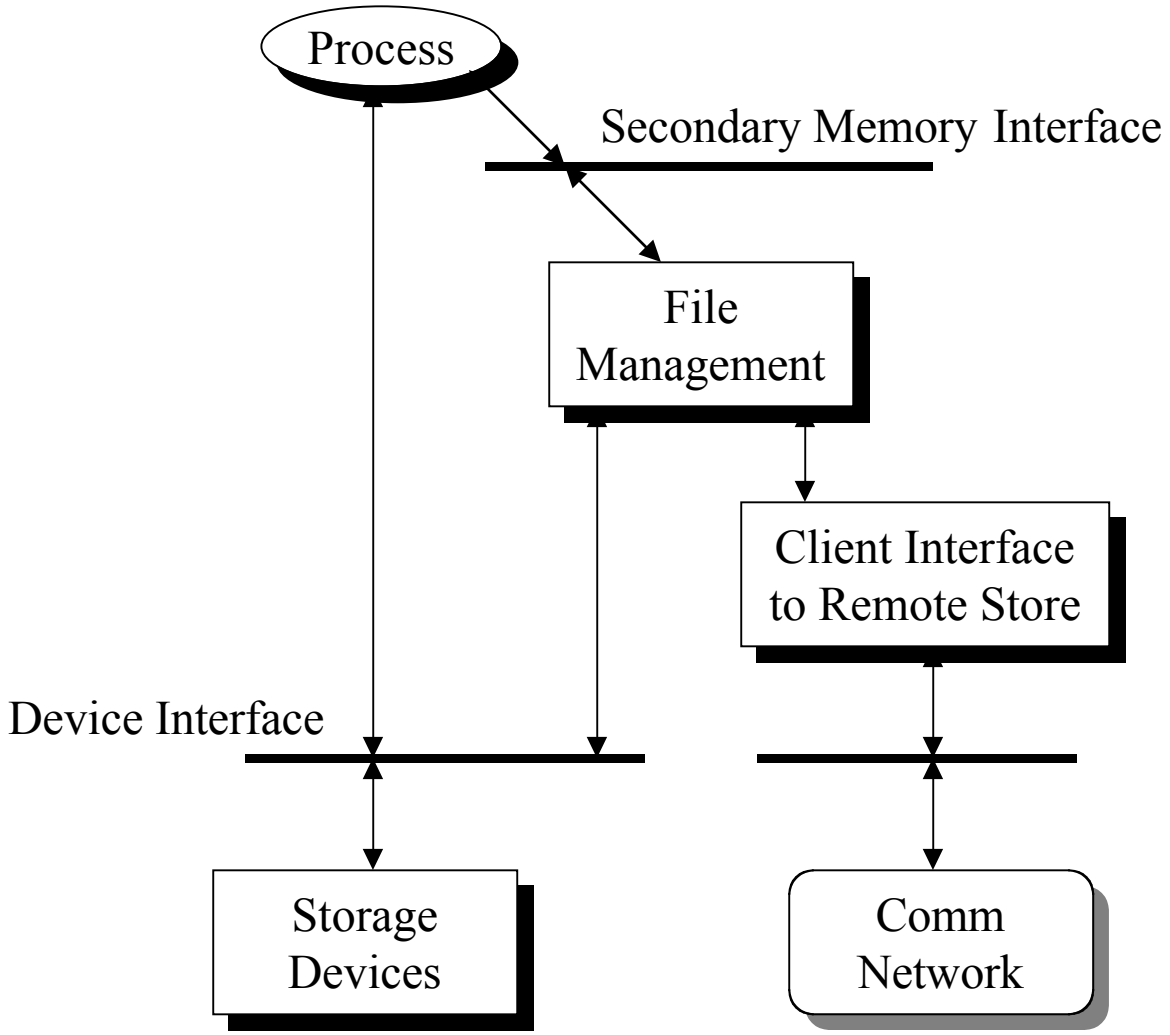
Explicit File Copying

- Need a way for a process on one machine to pass info to a process on another machine
- Technique
  - Sender writes a file
  - User manually copies file to a remote machine
  - Receiver opens the file and reads it
- Very coarse grained
- Very high latency

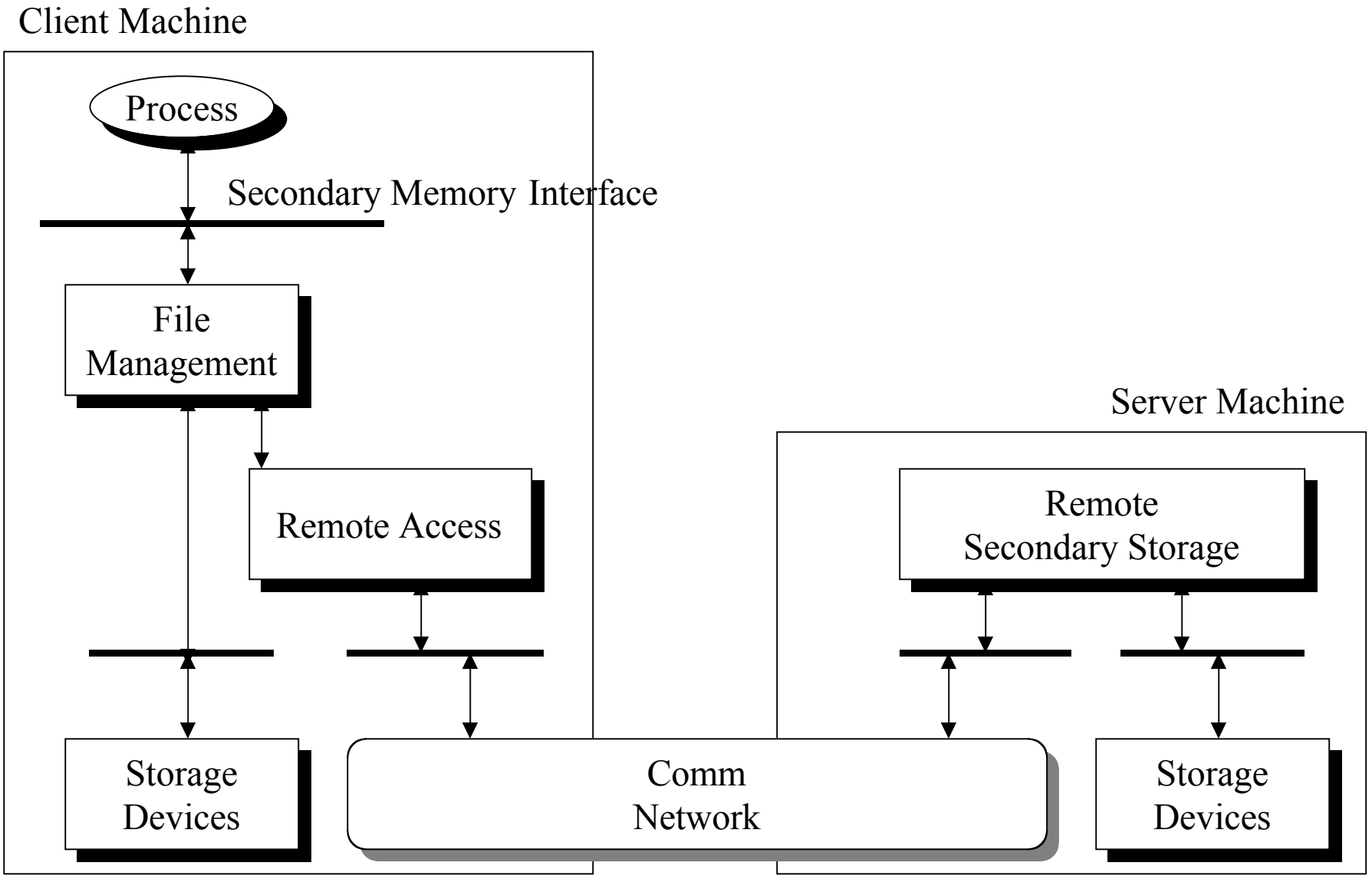
Multics Segmented Memory



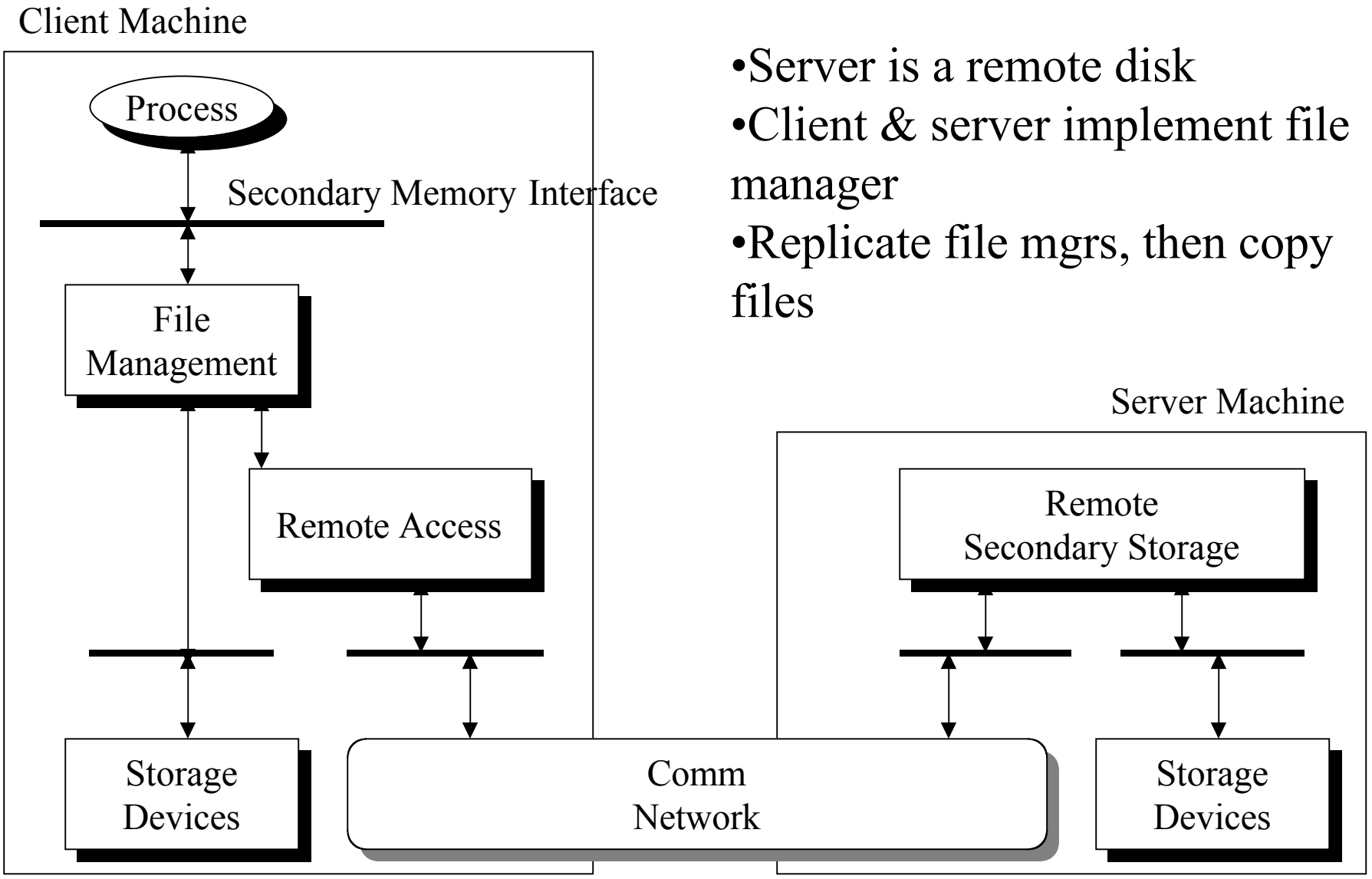
Remote Secondary Memory



Refined View



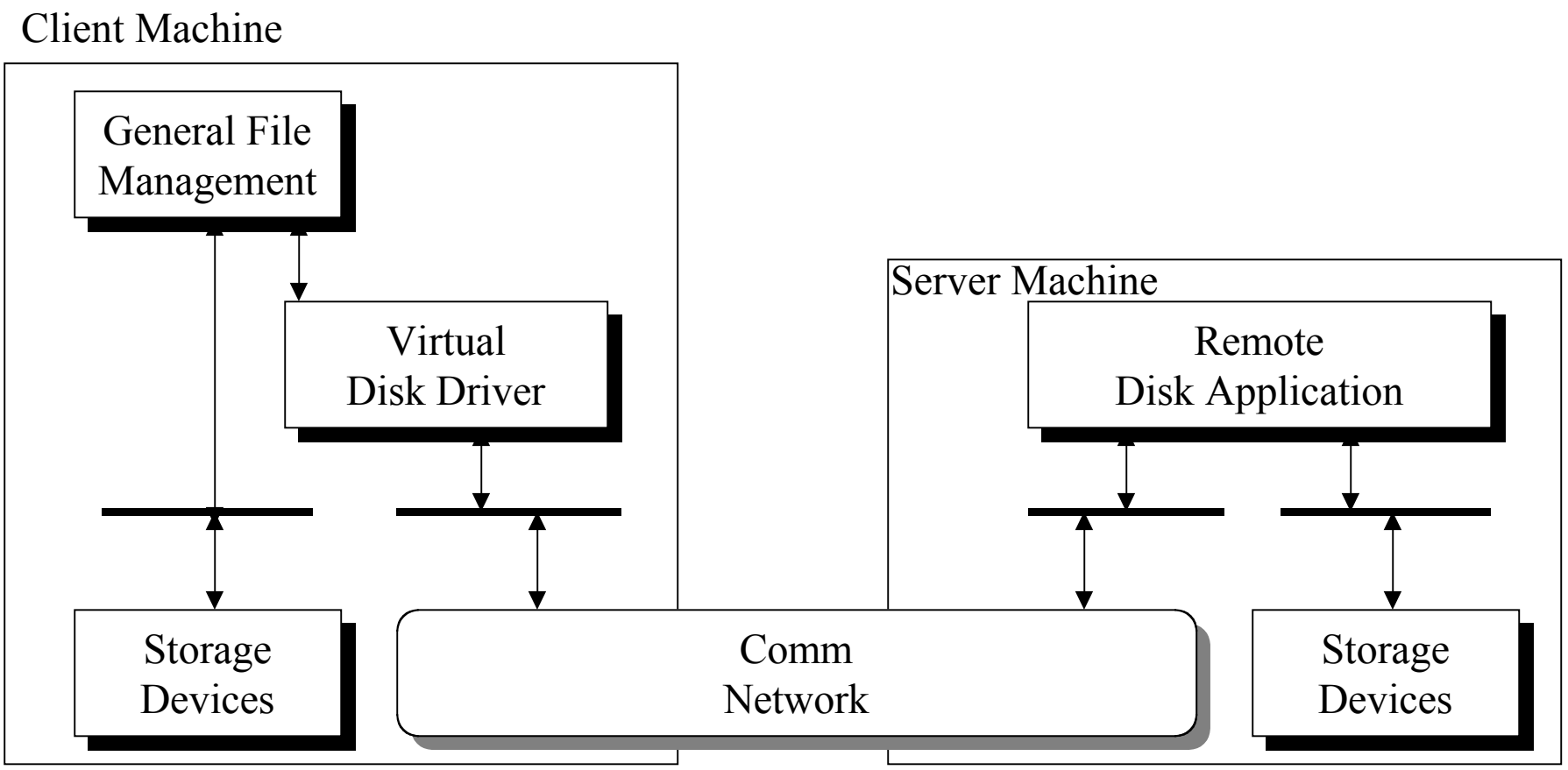
Refined View



- Server is a remote disk
- Client & server implement file manager
- Replicate file mgrs, then copy files

Remote Disk Server

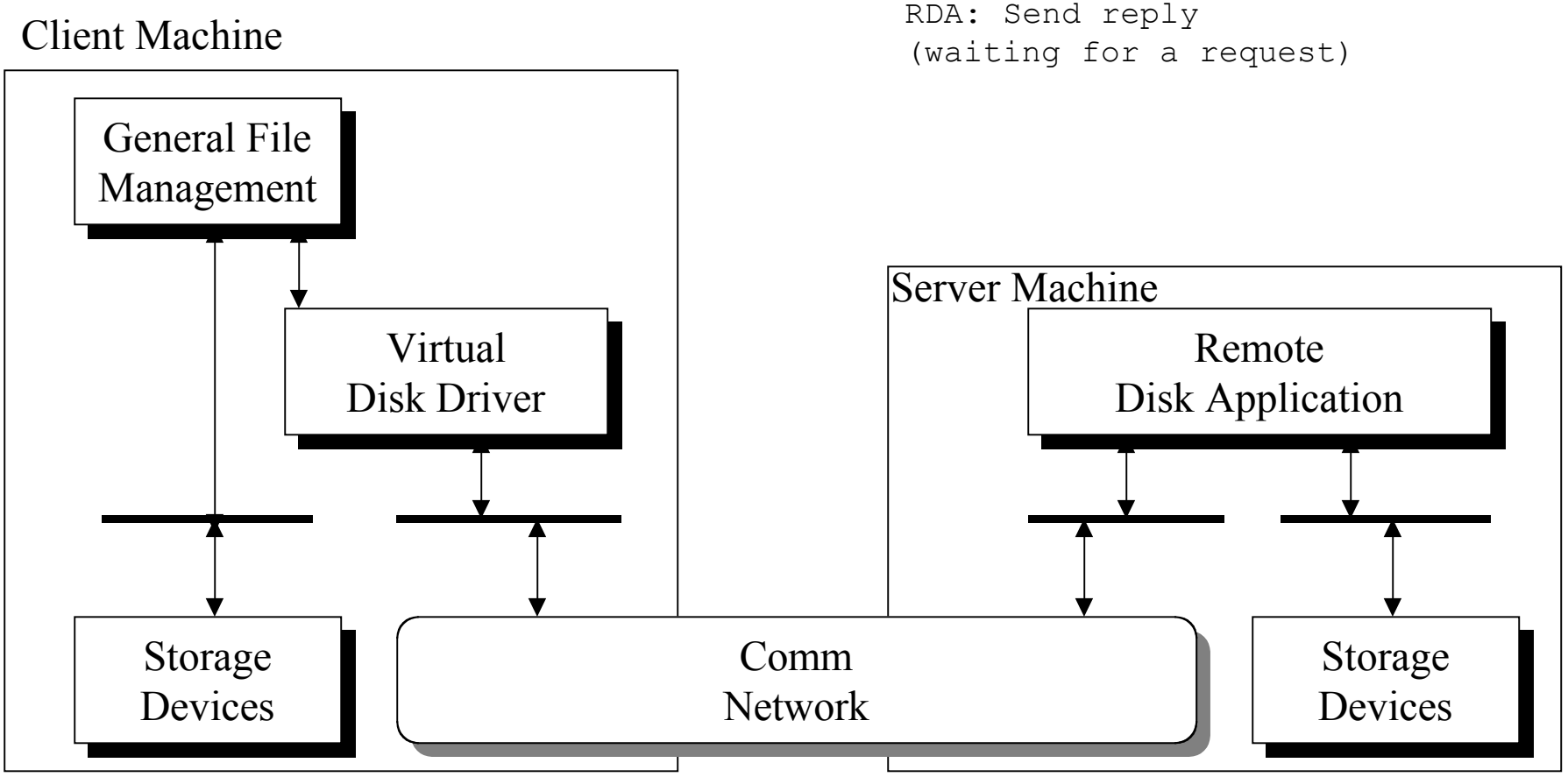
```
file mgr: diskRequest(details);
VDD:  Pack parameters;
VDD:  Send request;
(wait for response)
```



Remote Disk Server

```
file mgr: diskRequest(details);
VDD:  Pack parameters;
VDD:  Send request;
(wait for response)
```

```
(waiting for a request)
RDA: Unpack parameters;
RDA: Generate local disk request;
(waiting)
RDA: Generate reply;
RDA: Send reply
(waiting for a request)
```

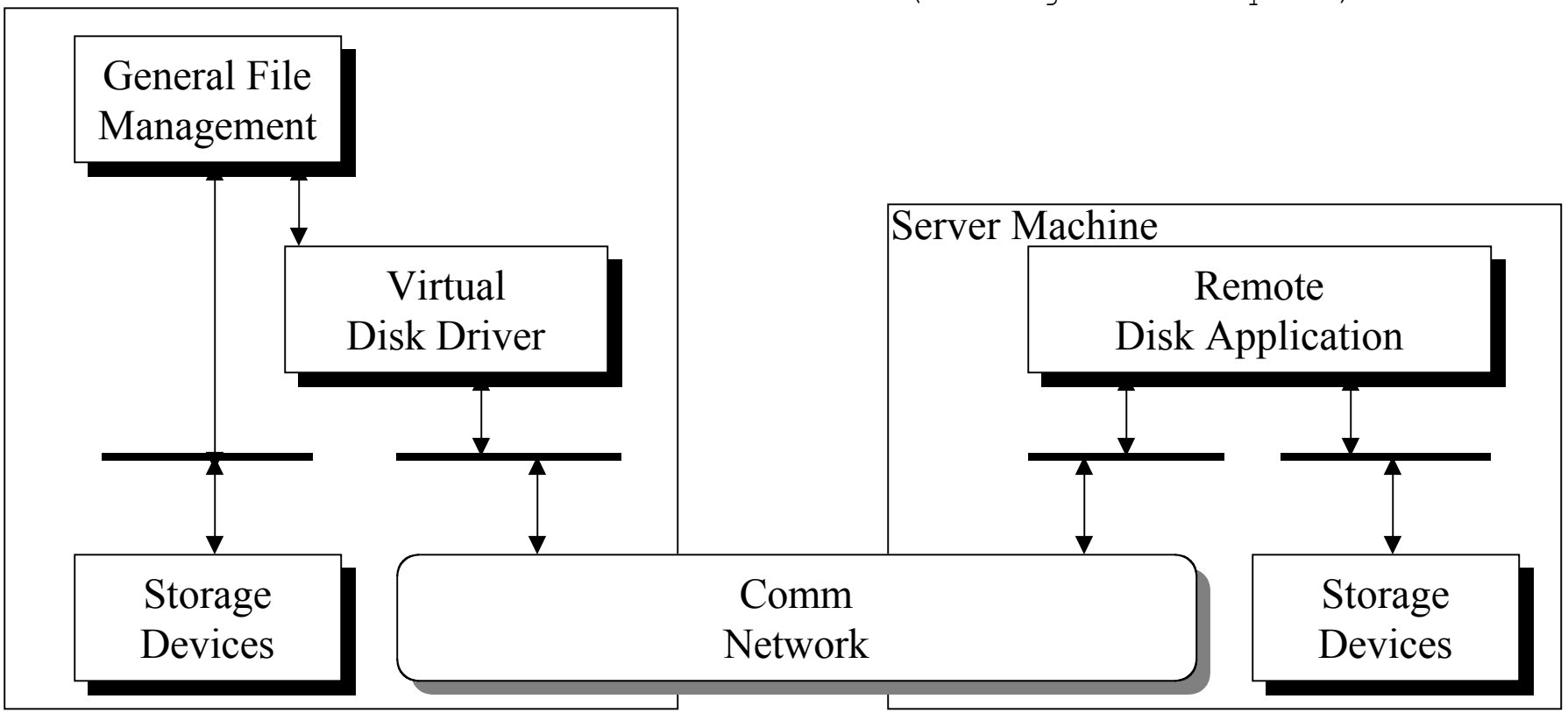


Remote Disk Server

```
file mgr: diskRequest(details);
VDD:  Pack parameters;
VDD:  Send request;
(wait for response)

VDD:  Receive reply;
VDD:  Unpack parameters;
VDD:  Return to file mgr
```

```
(waiting for a request)
RDA: Unpack parameters;
RDA: Generate local disk request;
(waiting)
RDA: Generate reply;
RDA: Send reply
(waiting for a request)
```



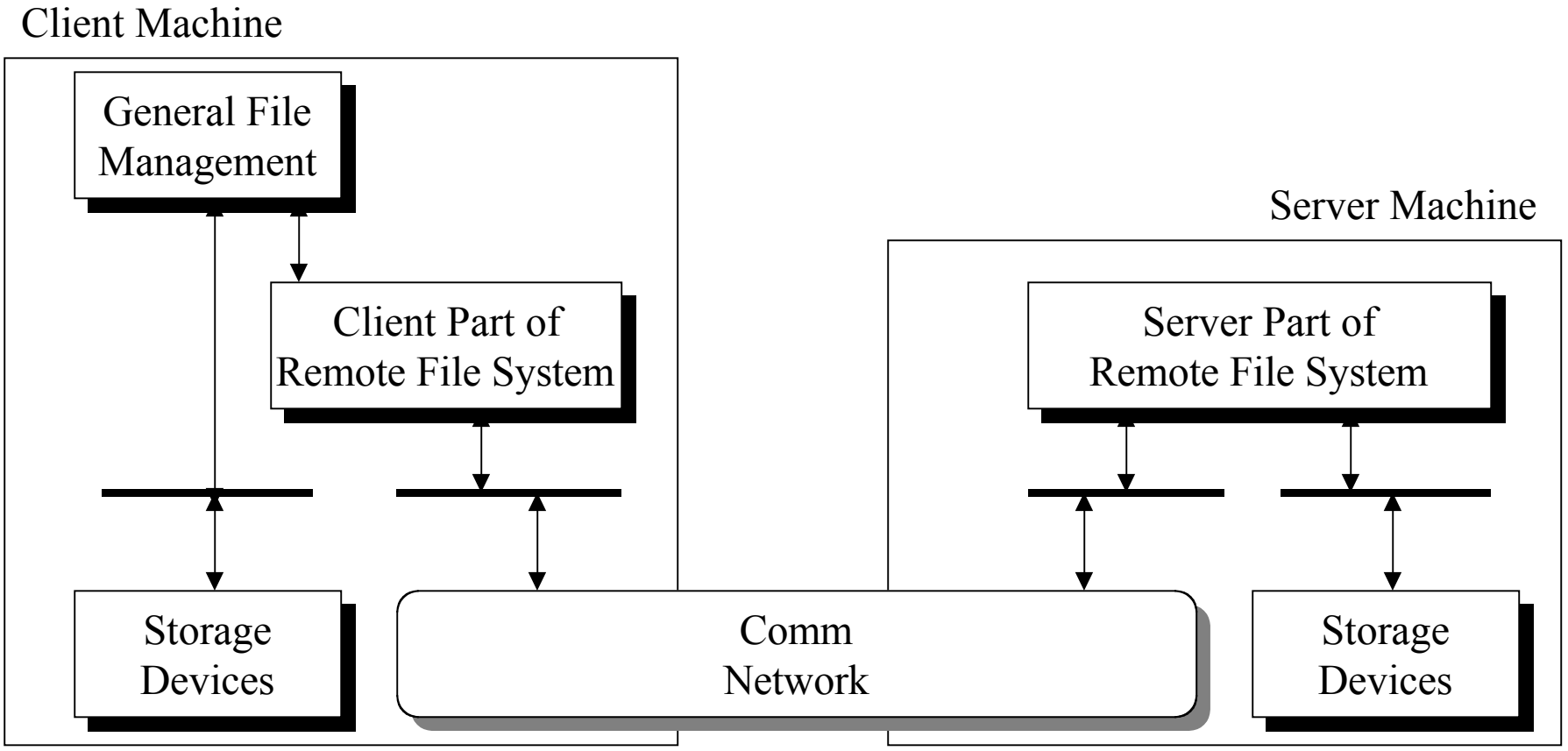
Performance & Reliability

- Became commercially feasible in about 1986
- Biggest concern was reliability
  - Reliable command execution
    - Time-outs
    - Idempotent disk operations
  - Crash recovery
    - Stateless servers
- Forerunner of the “network computer”

Remote File Server

- Read/write management
- Pack/unpack byte stream
- Buffering
- ...

- Block management
- Buffering
- Device management
- ...



Block Caching

- Widely used in all file systems
- In RFS can buffer at:
  - Server
    - Doesn't avoid network latency
  - Client
    - Consistency
    - Sometimes use sequential write consistency (no sharing if there are multiple writers)

Crash Recovery

- Client has a file open and server crashes
  - Distributed state makes recovery difficult
  - Can counteract with a *stateless server*
  - But it requires that state be transmitted with every service request
- Recovery-oriented file service, e.g., Sun NFS
- Performance-oriented file service

Directories

- Names
  - Superpath names
  - Remote mount
- Opening a file