SNS COLLEGE OF TECHNOLOGY

An Autonomous Institution

Coimbatore-35

Department of Computer Science and Engineering

23CST206–OPERATING SYSTEMS AND VIRTUALIZATION

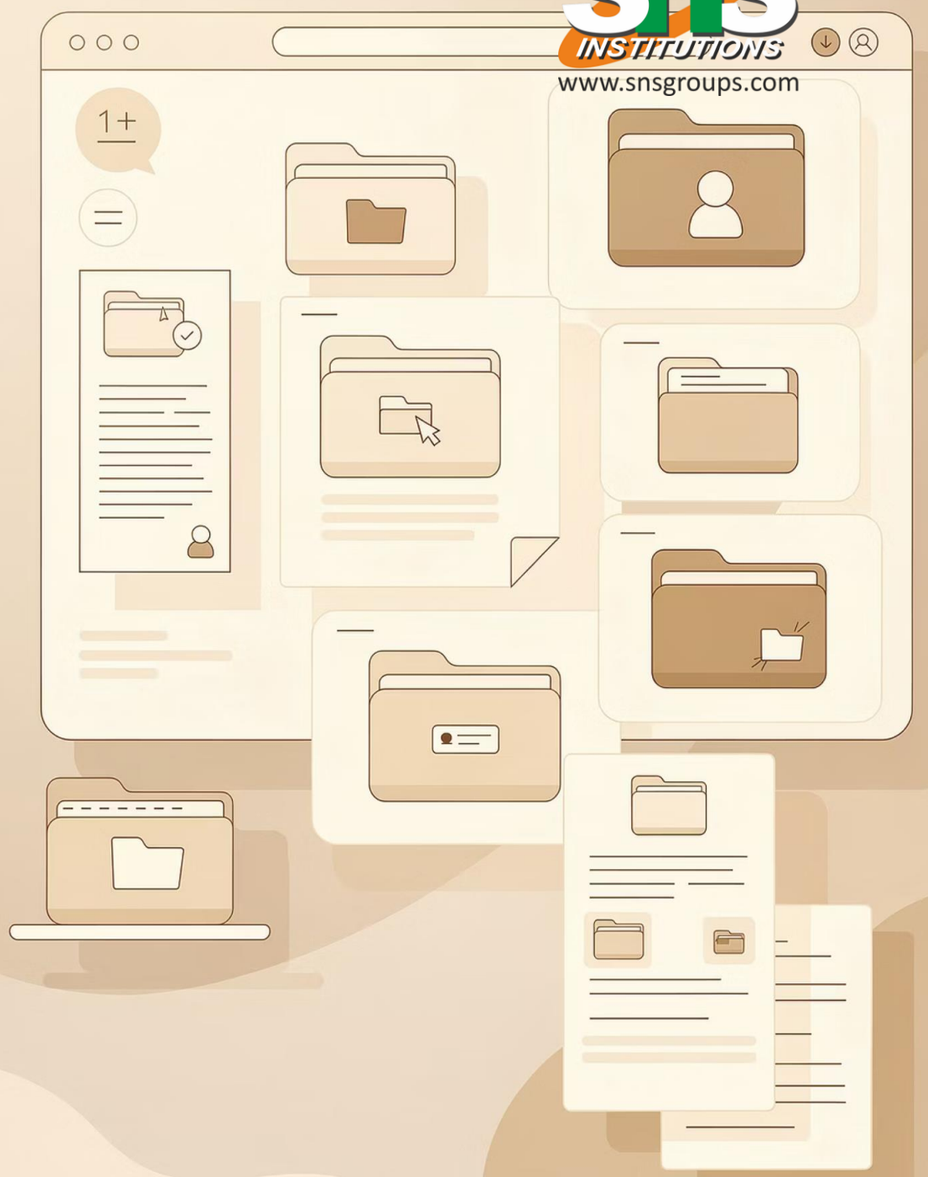
B.E- CSE /IV SEMESTER

UNIT – III MEMORY MANAGEMENT

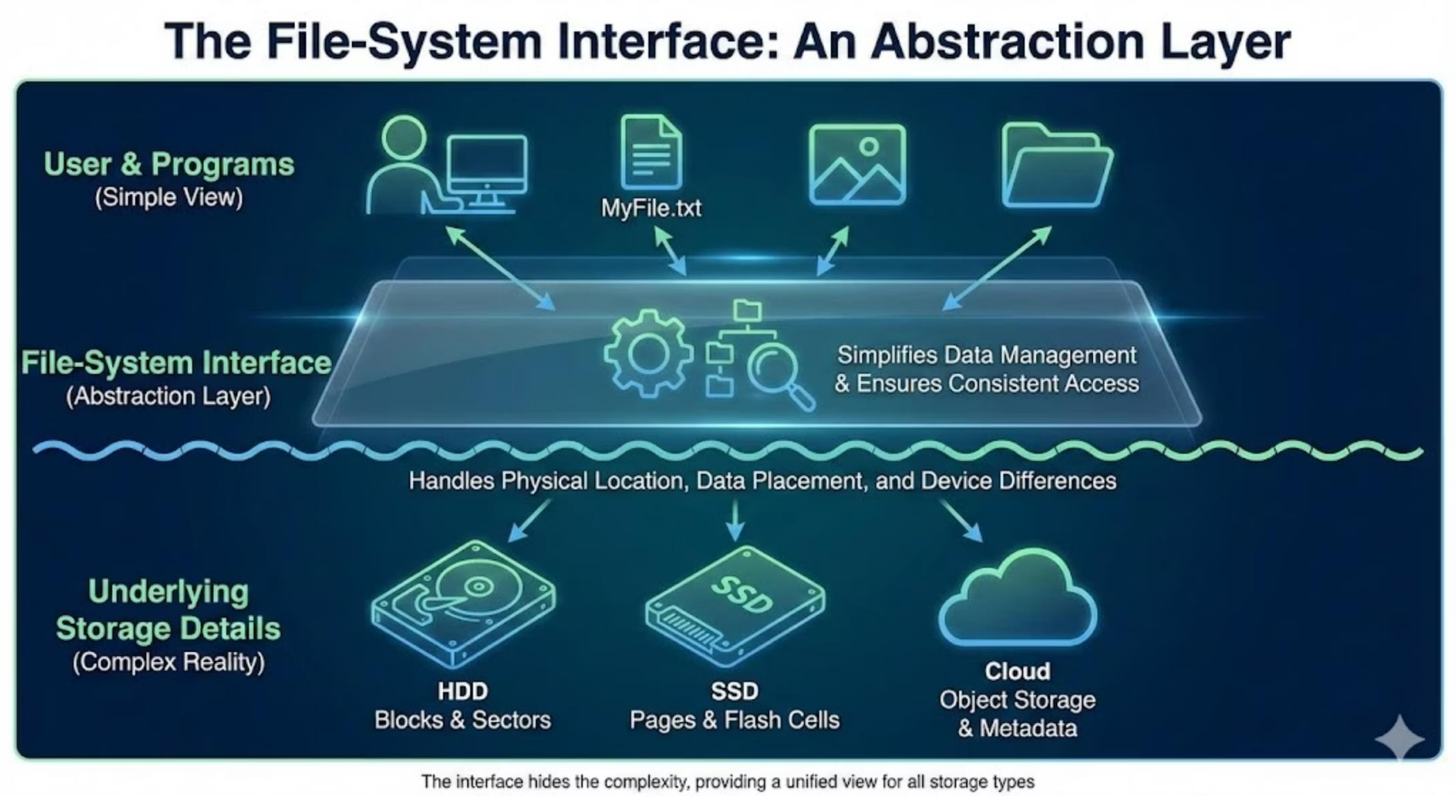
Topic 6:File system

File-System Interface in OS

Understanding how operating systems provide user-level access to files and data storage.



What is the File-System Interface?



The file-system interface is the part of the operating system that provides user-level access to files.

It defines how users and programs interact with files without worrying about the underlying storage details.

Objectives of File-System Interface

File Operations

Create, read, write, and delete files efficiently

Protection & Sharing

Enable access control, sharing, and data protection

Logical View

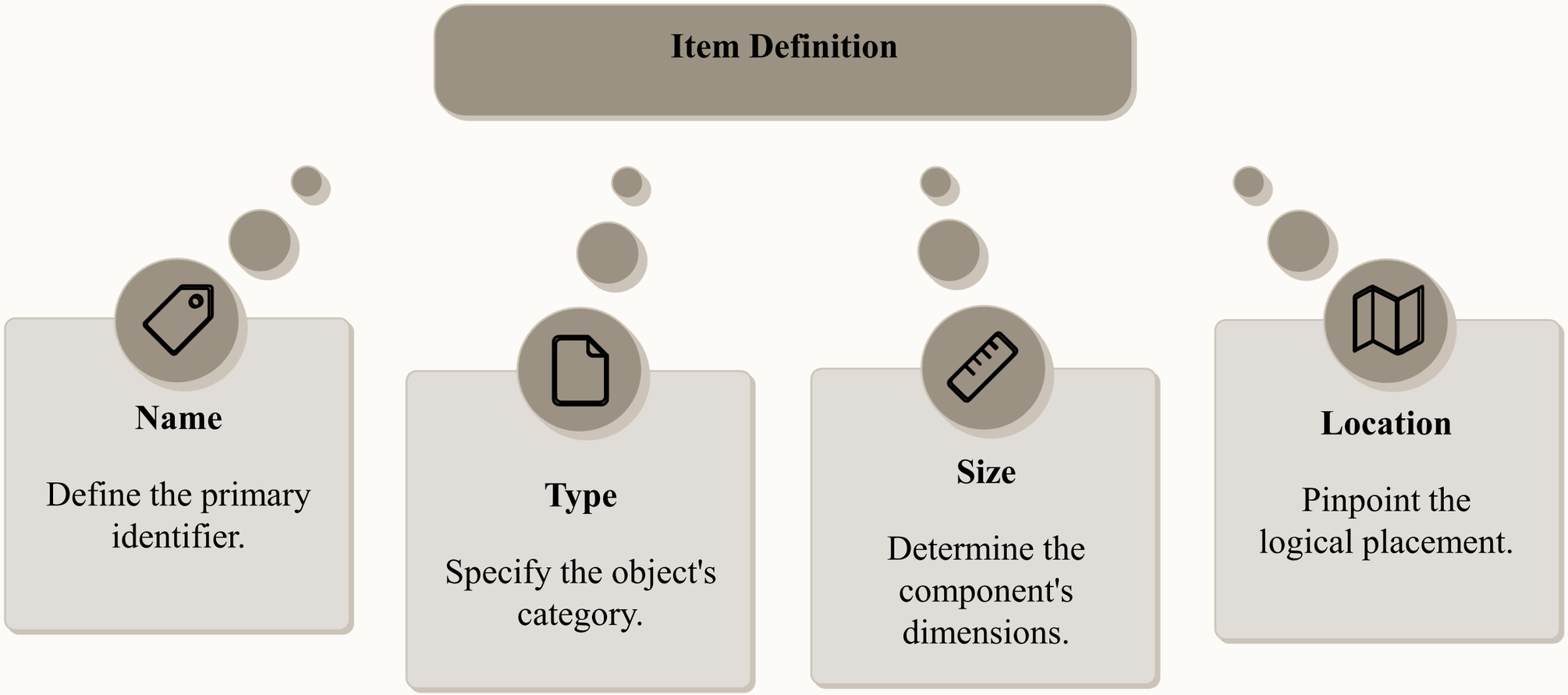
Expose logical data view without storage complexity

Efficiency

Ensure fast, reliable access across devices

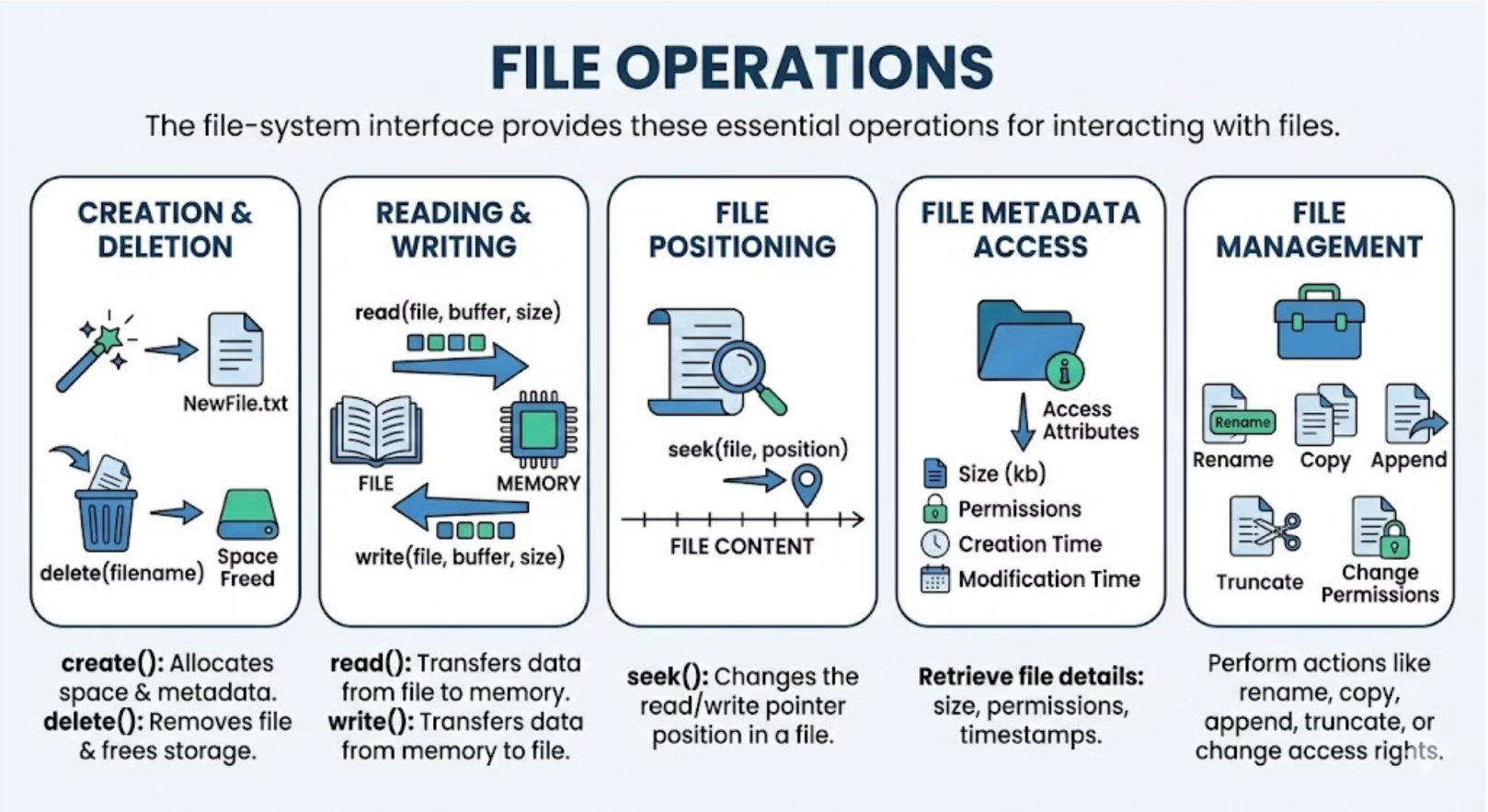
Abstraction of a File

A file is a collection of related information stored on a secondary storage device.

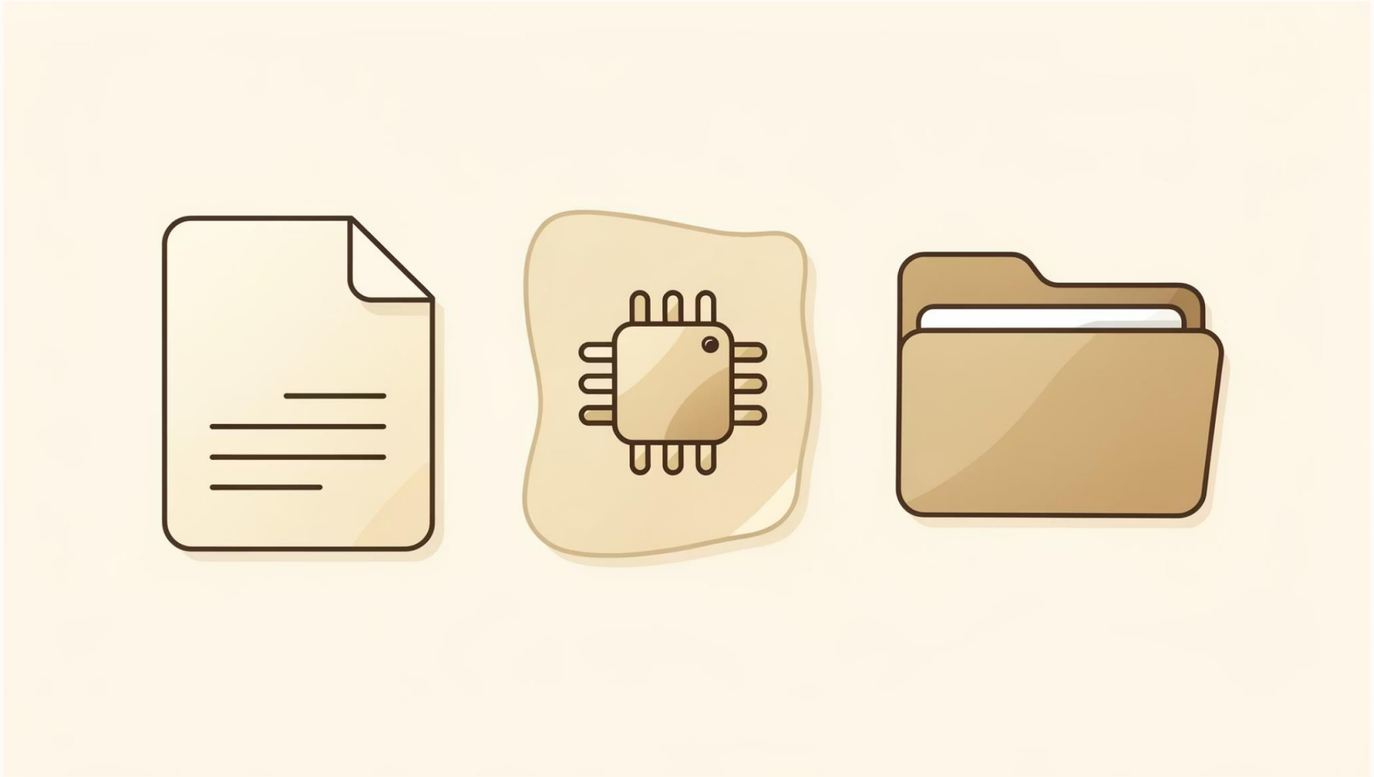


File Operations

The file-system interface allows the following basic operations:



File Types



Text Files

Contains human-readable characters.

Example: .txt, .c, .java

Binary Files

Contains non-text data (compiled programs, images, videos).

Example: .exe, .jpg, .mp4

Directory Files

Special files that store information about other files. Acts as a map of file names to file metadata.

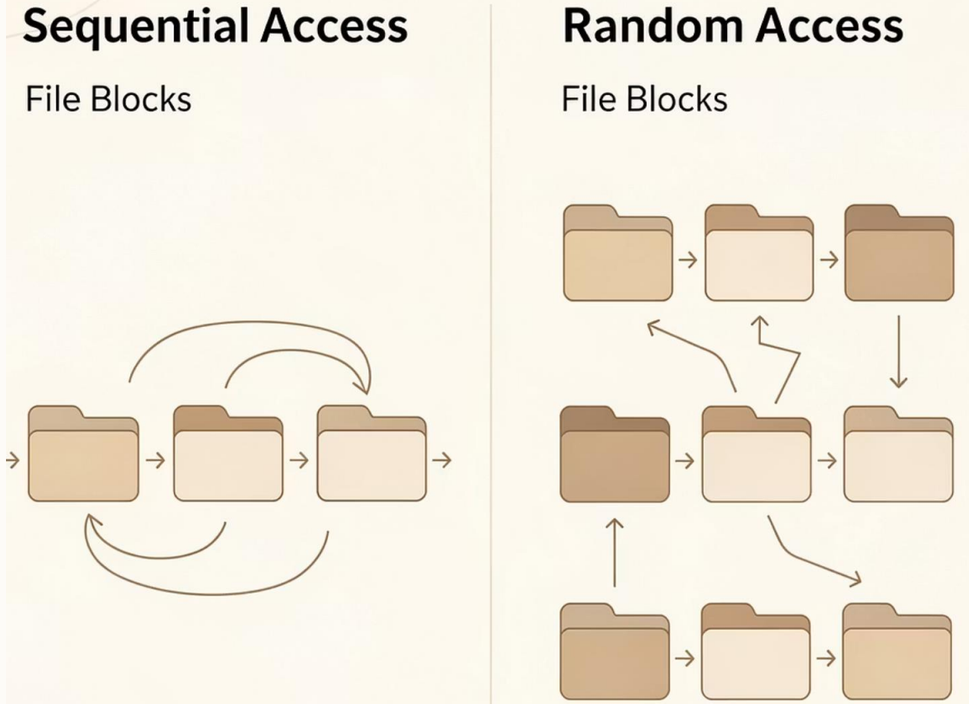
File Access Methods

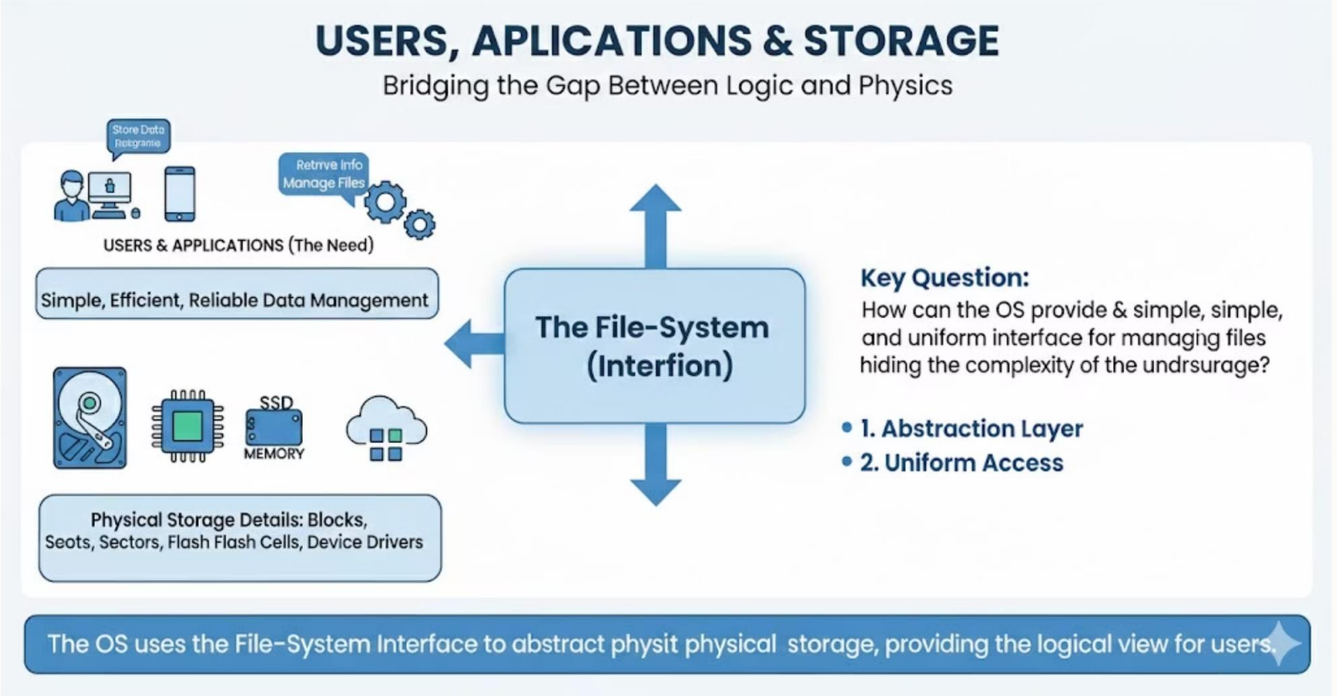
Sequential Access

Data is accessed in order, from beginning to end. Example: Reading a text file line by line.

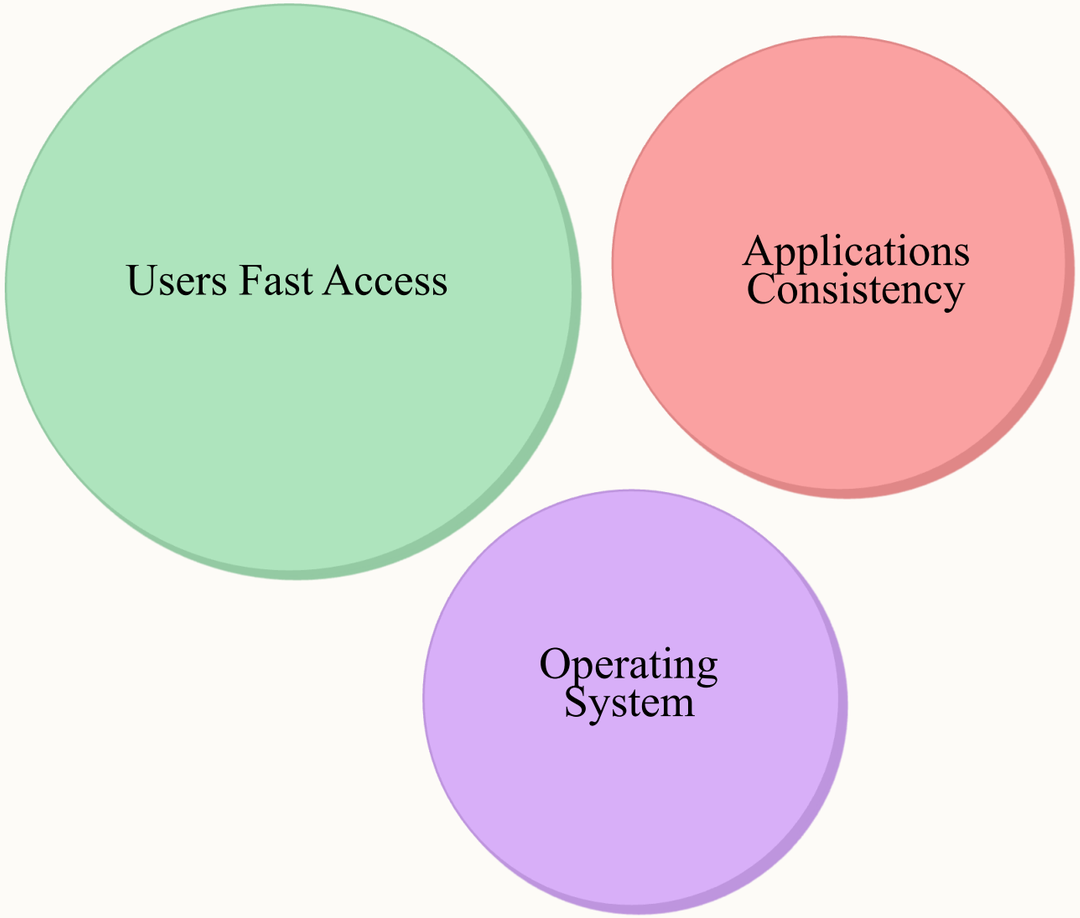
Direct / Random Access

Data can be accessed at any position directly using an offset. Example: Database files, video playback.




USERS, APLICATIONS & STORAGE
Bridging the Gap Between Logic and Physics
Store Data Retrgrame
Retrrve Info Manage Files
USERS & APPLICATIONS (The Need)
Simple, Efficient, Reliable Data Management
SSD
MEMORY
Physical Storage Details: Blocks, Seots, Sectors, Flash Flash Cells, Device Drivers
The File-System (Interfion)
Key Question:
How can the OS provide & simple, simple, and uniform interface for managing files hiding the complexity of the undrsurage?
1. Abstraction Layer
2. Uniform Access
The OS uses the File-System Interface to abstract physit physical storage, providing the logical view for users.

Empathize: Understanding the Need



Pain Points

- Searching large directories manually is slow
- Incorrect access or deletion can corrupt data
- Different storage devices require different handling

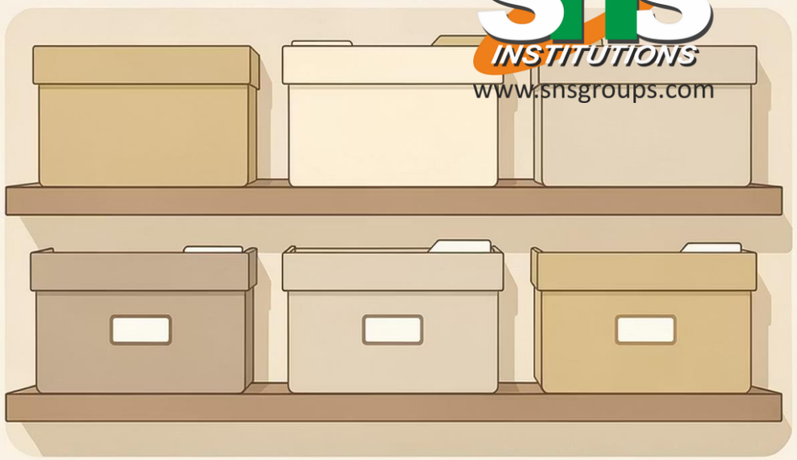
Define: Clear Problem Definition

Uniform Access

Basic Operations

Data Protection

File Management



Ideate: Possible Solutions

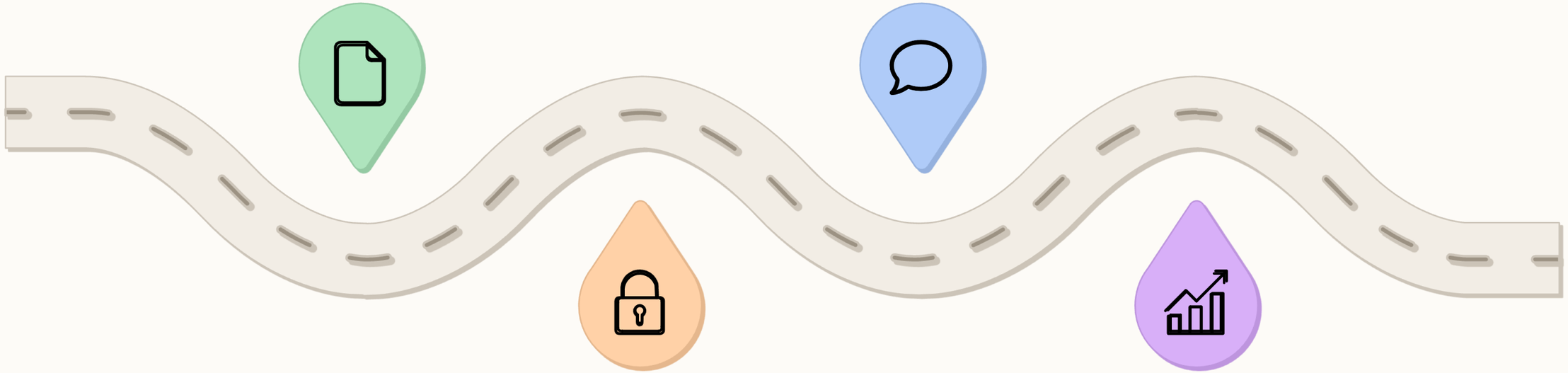
Exploring approaches for the file-system interface:

File API

Defines how applications interact with files.

Metadata

Provides descriptive information about the data.



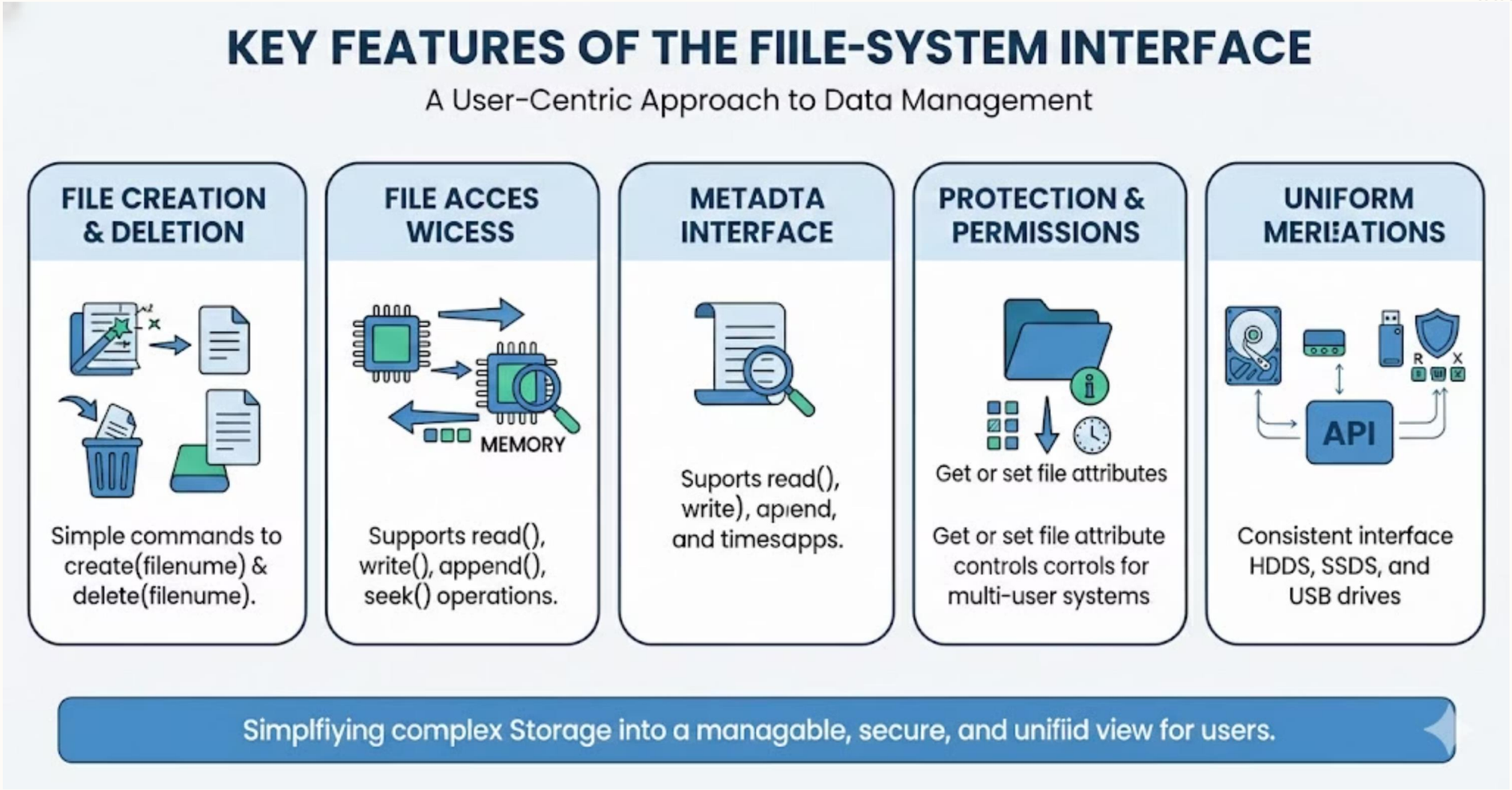
Access Control

Manages permissions and security for data.

Access Methods

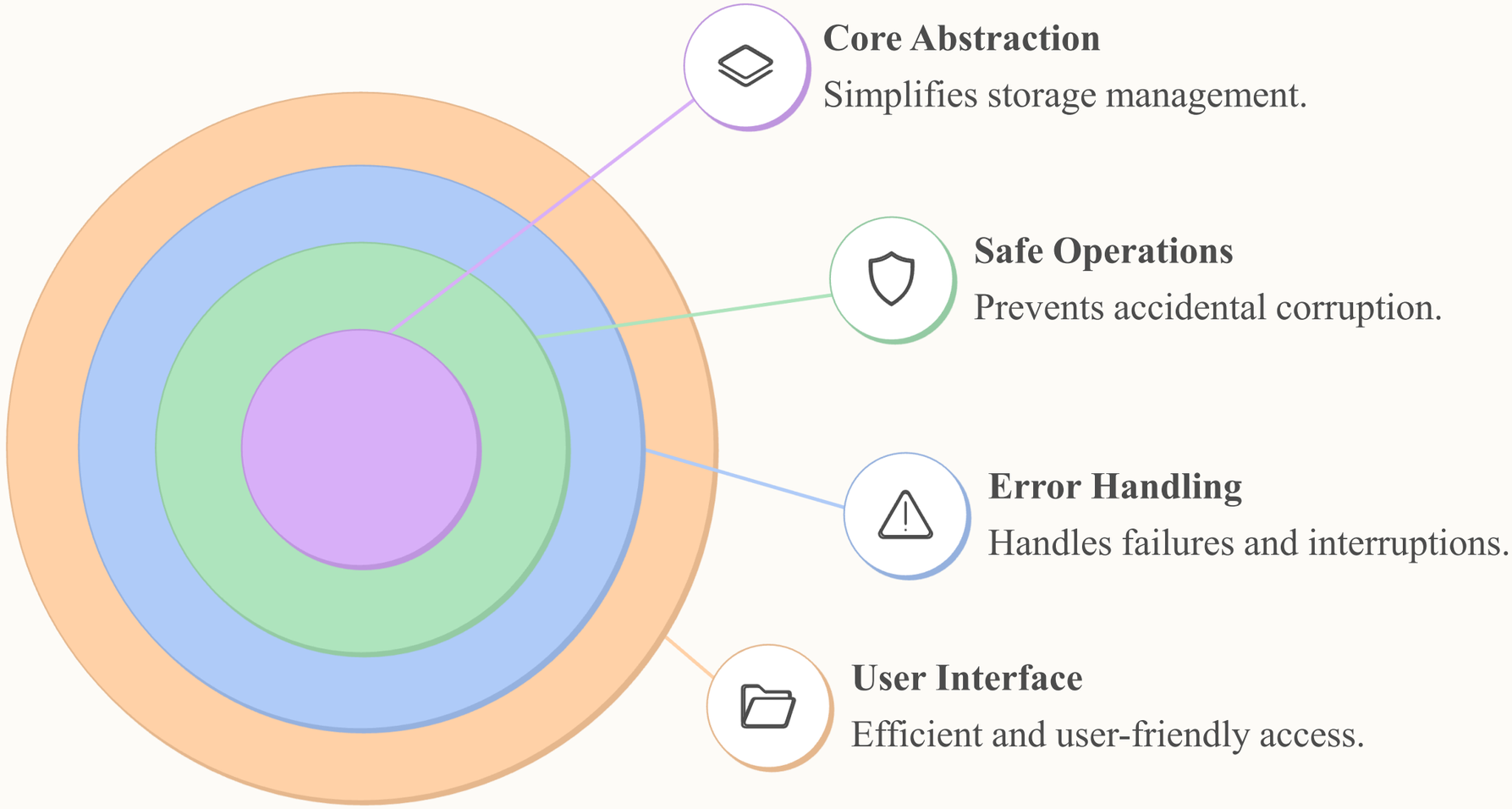
Specifies ways to retrieve and use data.

Prototype: Chosen Solutions



Test: Evaluation of Solutions

The prototype was evaluated based on key criteria:

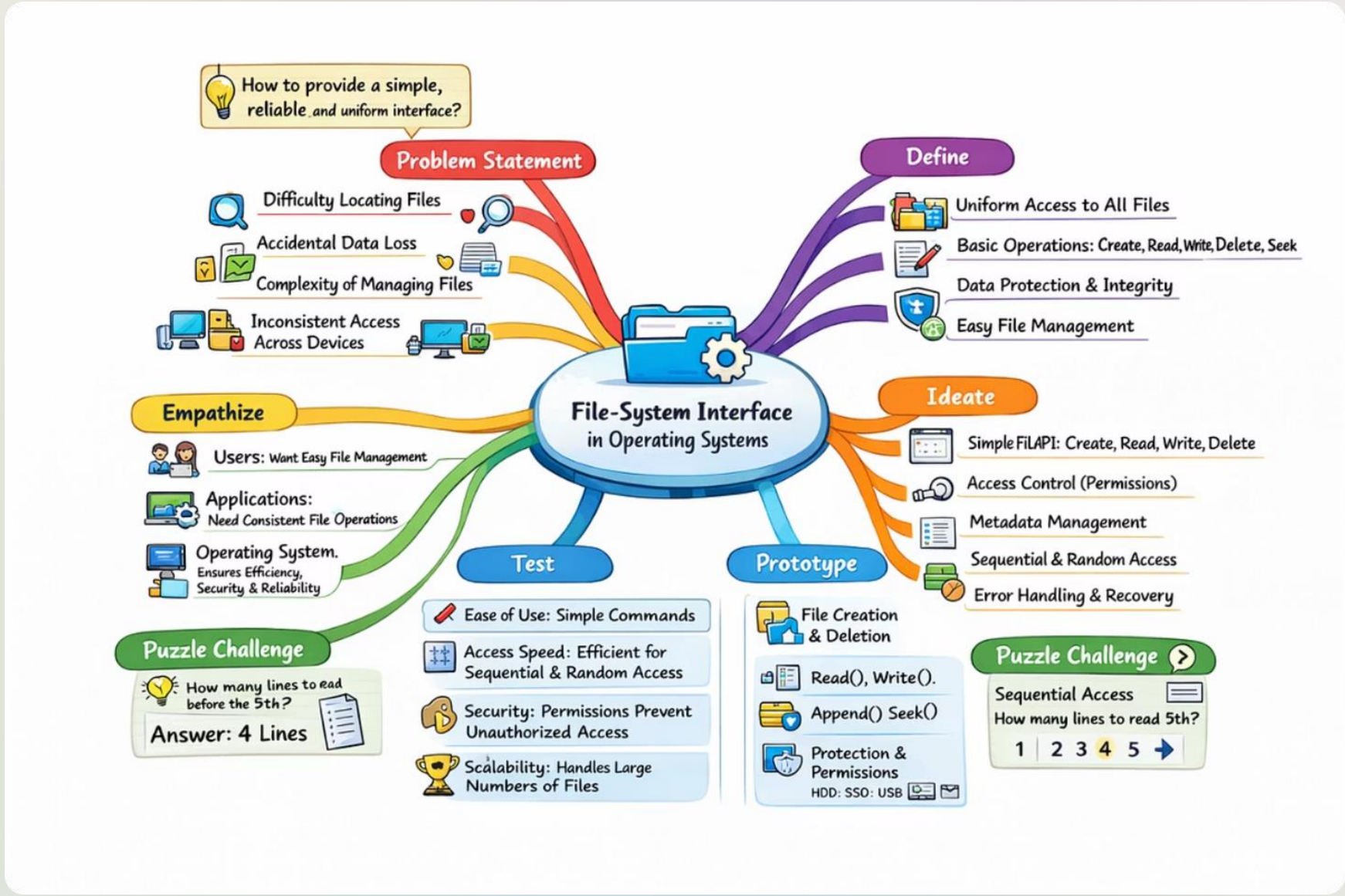


Core Abstraction
Simplifies storage management.

Safe Operations
Prevents accidental corruption.

Error Handling
Handles failures and interruptions.

User Interface
Efficient and user-friendly access.


How to provide a simple, reliable and uniform interface?
File-System Interface in Operating Systems
Problem Statement
Difficulty Locating Files
Accidental Data Loss
Complexity of Managing Files
Inconsistent Access Across Devices
Define
Uniform Access to All Files
Basic Operations: Create, Read, Write, Delete, Seek
Data Protection & Integrity
Easy File Management
Empathize
Users: Want Easy File Management
Applications: Need Consistent File Operations
Operating System. Ensures Efficiency, Security & Reliability
Ideate
Simple FILAPI: Create, Read, Write, Delete
Access Control (Permissions)
Metadata Management
Sequential & Random Access
Error Handling & Recovery
Puzzle Challenge
How many lines to read before the 5th?
Answer: 4 Lines
Test
Ease of Use: Simple Commands
Access Speed: Efficient for Sequential & Random Access
Security: Permissions Prevent Unauthorized Access
Scalability: Handles Large Numbers of Files
Prototype
File Creation & Deletion
Read(), Write().
Append() Seek()
Protection & Permissions
HDD: SSO: USB
Puzzle Challenge
Sequential Access
How many lines to read 5th?
1 2 3 4 5