

Title: Saving Application Data in Mobile Devices

Introduction

- Mobile applications often need to store and retrieve data efficiently.
- Data storage is crucial for maintaining user preferences, app settings, and database records.
- **Android provides multiple storage options**, including:
 - Key-Value storage
 - File storage
 - Database storage
 - Cloud storage
- Choosing the right storage method depends on **data size, security, and accessibility**.

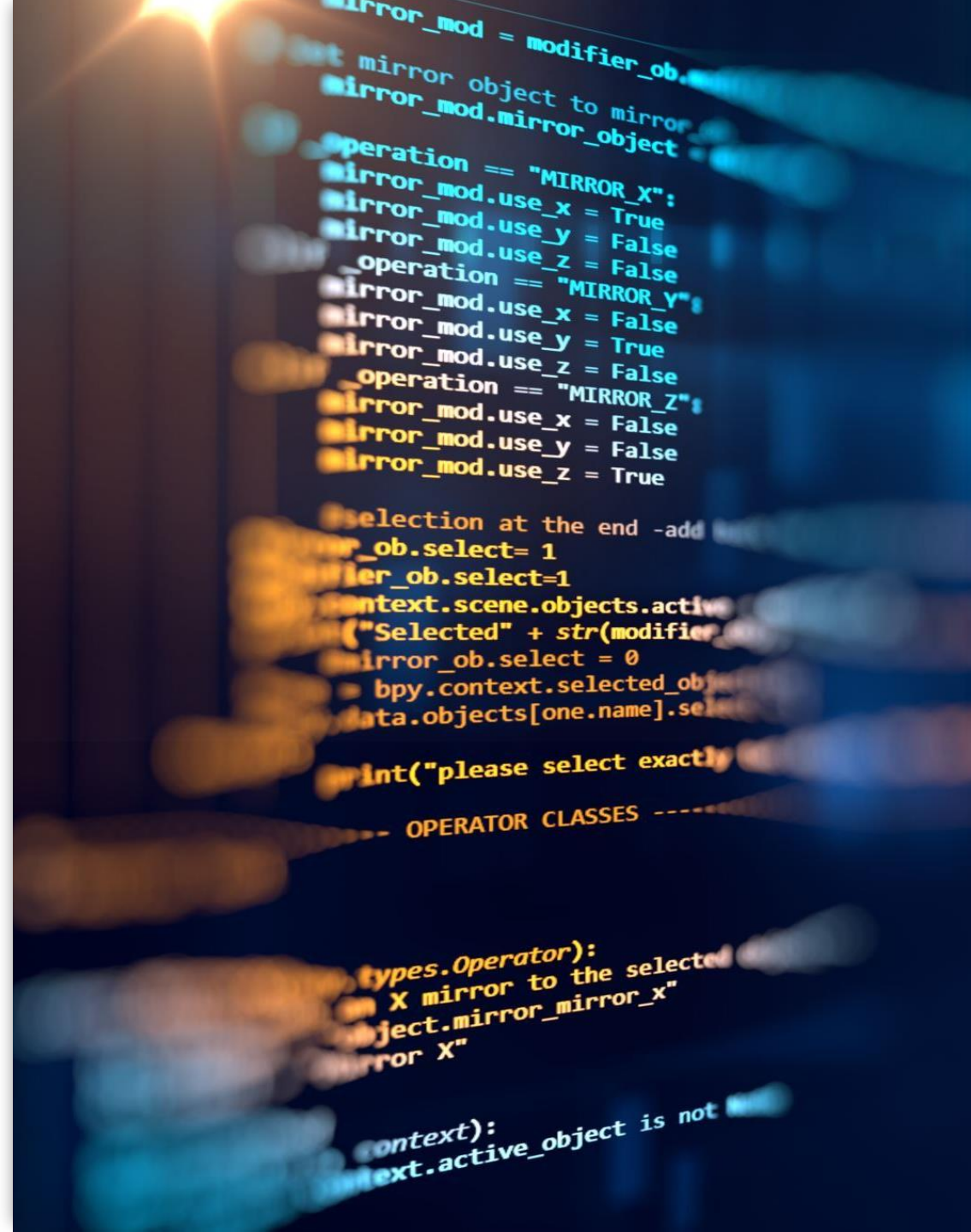
Types of Data Storage in Android

- Android offers different ways to store data:
- **Shared Preferences** – Store small key-value data (e.g., user settings).
- **Internal Storage** – Store private files within the app.
- **External Storage** – Store larger files accessible by other apps.
- **SQLite Database** – Store structured relational data.
- **Room Database (Jetpack)** – A modern SQLite wrapper with easy APIs.
- **Cloud Storage** – Store data on remote servers for synchronization and backup.



Shared Preferences (Key-Value Storage)

- Used for storing small, lightweight data as **key-value pairs**.
- Best for saving **user preferences, settings, login status**.
- Stored in an XML file within the app's private storage.
- **Example Usage:** Saving the dark mode setting in an app.



Internal Storage (Private File Storage)

- Stores data inside the app's **private directory**.
- Data **cannot be accessed by other apps**.
- Best for storing **confidential files, cache, and app-specific data**.



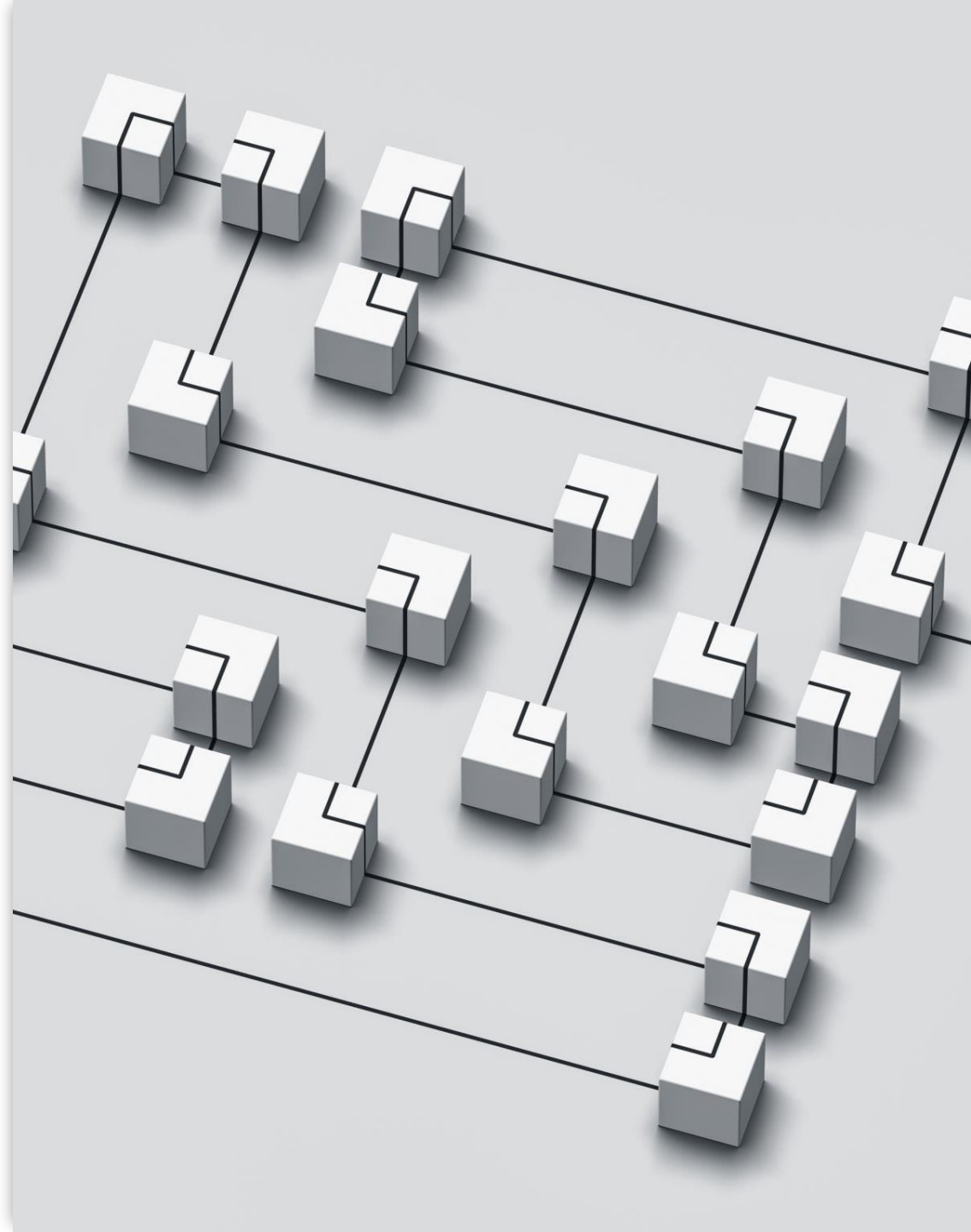
External Storage (Public File Storage)

- Stores files on **SD cards or external storage directories**.
- Can be accessed by other apps if **permissions are granted**.
- Used for **storing media files, documents, and downloads**.
- **Permissions Required:**
 - READ_EXTERNAL_STORAGE
 - WRITE_EXTERNAL_STORAGE
 - **Example Usage:** Saving an image file.



SQLite Database (Structured Data Storage)

- Lightweight **relational database** for storing structured data.
- Used for **storing large sets of data**, like **contacts, orders, or notes**.
- **Requires writing SQL queries** for CRUD operations.
- **Example: Creating a Database Table**



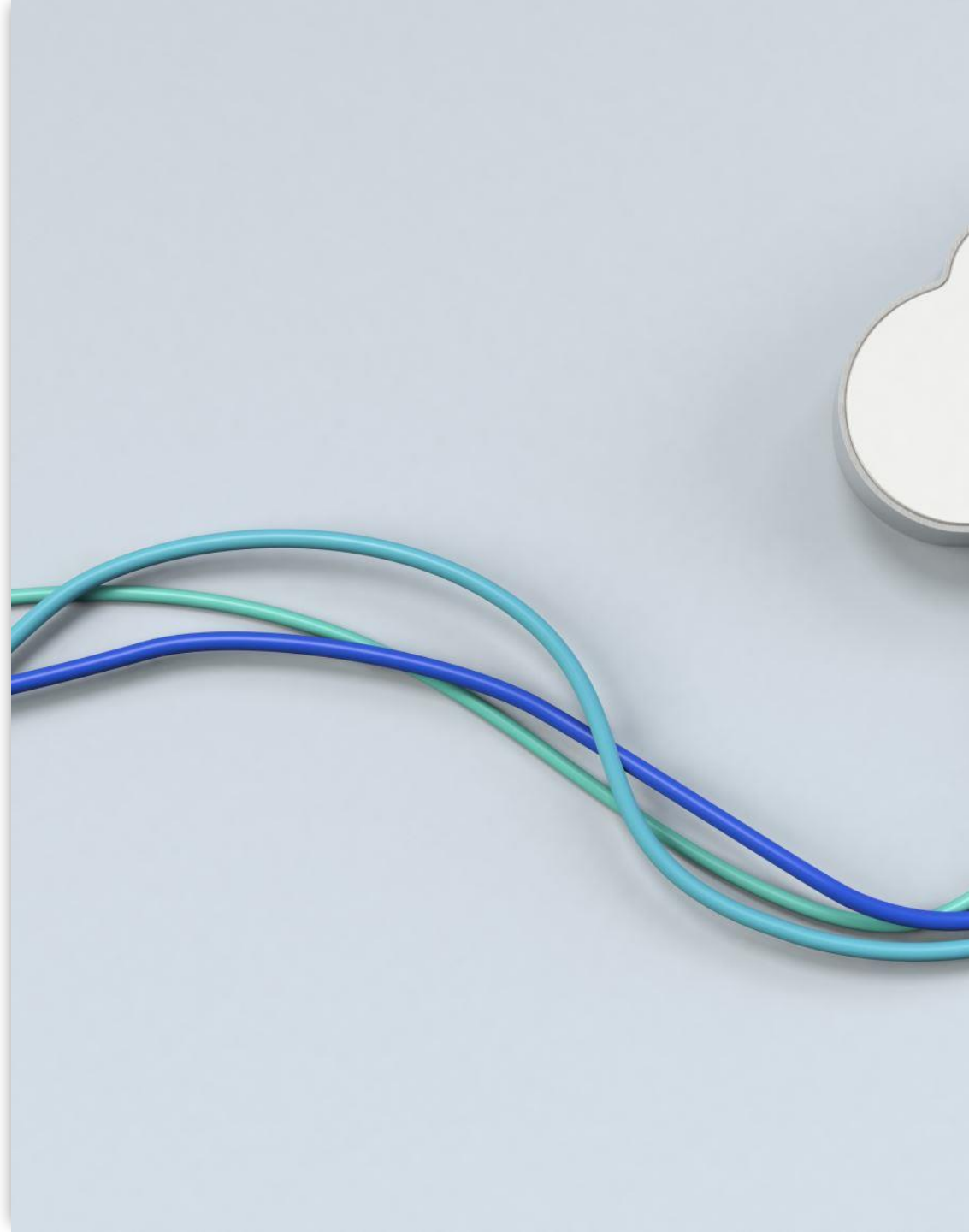
Slide 8: Room Database (Modern SQLite with Jetpack)

- Room is a **wrapper around SQLite** that simplifies database operations.
- Provides **easy-to-use APIs** and avoids raw SQL queries.
- Uses **DAO (Data Access Object)** for database operations.
- **Example: Defining a Room Entity (Table)**



Cloud Storage (Remote Data Synchronization)

- Cloud storage allows apps to store data remotely, making it accessible across devices.
- **Popular Cloud Storage Services:**
- **Firebase Realtime Database** – Stores JSON data and syncs in real-time.
- **Google Drive API** – Stores and retrieves files from the user's Google Drive.
- **AWS S3 / Firestore** – Scalable cloud storage for enterprise apps.



Conclusion

Storage Type	Best Use Case
Shared Preferences	User settings, small key-value data
Internal Storage	Private app-specific files
External Storage	Media files, downloads
SQLite	Structured relational data
Room Database	Modern SQLite with easy APIs
Cloud Storage	Remote storage, multi-device sync