# Document on Cross-Platform Storage Management Approaches in Flutter

I have chosen Flutter to develop the project. Flutter supports several data storage options, suitable for different types of applications and data needs. My application is cross-platform and works across the iOS, macOS and Android platforms. Accordingly, here are the approaches for cross-platform storage management in Flutter:

## Local Storage Options

1. Shared Preferences
   * Description: This plugin wraps platform-specific persistent storage for simple data (key-value pairs). It is ideal for saving user preferences or settings.
   * **Pros**: Easy to implement; automatically abstracts platform differences; synchronizes data across app instances.
   * **Cons**: Limited to simple data types; not suited for storing large amounts of data or complex objects.
2. SQLite (via the sqflite package)
   * Description: A robust option for structured data storage that creates a local database on the device.
   * **Pros**: Supports complex data models and queries; works consistently across all target platforms.
   * **Cons**: Requires more setup and maintenance; database schema migrations need careful handling as your app evolves.
3. Hive
   * Description: A lightweight, NoSQL database written in pure Dart, making it a good fit for Flutter applications.
   * **Pros**: Fast performance; does not require bridge to native code hence faster on all platforms; simple to use with a flexible schema.
   * **Cons**: Not as widely used or tested in large-scale production as SQLite; might have limitations in highly complex data operations.

## Remote Storage Options

1. Firebase Firestore
   * Description: A cloud-hosted NoSQL database that syncs across all users in real-time.
   * **Pros**: Seamless integration with Flutter; provides offline data access and synchronization; robust security features.
   * **Cons**: Depends on network availability for initial data fetching and updates; costs can scale up with increased usage.
2. Cloud Storage Solutions (e.g., Firebase Storage, AWS S3)
   * Description: For storing files like images, videos, and large datasets.
   * **Pros**: Offloads heavy data handling from devices; scalable storage capacity; robust data management and security.
   * **Cons**: Requires robust internet connection for uploads and downloads; implementation complexity; potential latency in data access.

### 