🚀 DEEP TECHNICAL HANDOVER: ACR (v0.9.4V)

1. AI ENGINE & PERSISTENCE (GeminiService)

- **Implementation**: Uses `gemini-1.5-flash` via Google Generative AI REST API.
- **OCR & Translation**: Combines `OpenCV` (for balloon detection) with Gemini Vision. The `GeminiService` analyzes the page image and returns a JSON with bounding boxes and translated text.
- **Coordinate Mapping**: Implements a custom `OverlayView` that maps AI coordinates (0-1000) to the reader's `AspectFit` image scale using `GeometryReader`.
- **Translation Caching**: Every translated page is saved as a JSON in `Documents/Translations/`. The filename key is `SeriesName_Issue_Page`. On page load, the app checks for this file to avoid redundant API calls and quota consumption.
- **Story Recap**: Uses a specific prompt that feeds Series/Volume metadata to Gemini to generate a plot summary of "what happened so far".

2. CORE STORAGE & SECURITY (AppState + AppStorage)

- **Security Architecture**: Completely removed hardcoded secrets. All credentials (Komga pwd, API Keys) use `@AppStorage`, which is backed by encrypted `UserDefaults`.
- **Settings UX**: `SettingsView` uses a state-driven `isEditing` toggle.
  - **Visual Feedback**: Fields use a conditional `.background(Color.yellow.opacity(0.15))` when editing is active.
  - **Visibility**: Dedicated `showServerPassword` and `showGeminiApiKey` states control the switch between `SecureField` and `TextField`, with auto-obscure on save.

3. METADATA ENGINE (ComicInfo Parser)

- **Logic**: A dedicated `XMLParser` delegate (in `ComicInfo.swift`) extracts standard ComicRack metadata.
- **Integration**: Metadata is injected into `LocalBook` objects during the scanning phase of `LocalLibraryView`. This data powers the "Hero" detail view and the improved labels in the "Continue Reading" carousel.

4. DOWNLOAD & CONFLICT MANAGEMENT (DownloadManager)

- **Engine**: A centralized `ObservableObject` singleton that manages a queue of background downloads.

- **Conflict UI**: Implemented an Alert-based conflict resolver (found in `BookListView` and `SeriesListView`) that handles "Overwrite", "Skip", and "Apply to All" logic for both single files and recursive series imports.
- **Global Overlay**: `DownloadStatusOverlay` is injected into `MainTabView` ZStack, providing real-time progress for background tasks across any tab.

5. NEON UI ARCHITECTURE

- **Custom Folders**: `UnifiedFolderIcon` and `ComicBoxCard` use custom paths and gradients to simulate 3D comic boxes.
- **Hero View**: `BookListView` uses a `MatchedGeometryEffect`-ready layout (though currently separate) with a blurred background sampled from the currently selected book's first page.
- **Filtering**: `PillFilterBar` is a horizontal scroll component that filters the `localRootNode` in real-time based on the library name prefix.

6. PROJECT STRUCTURE & PATHS

- **Source of Truth**: `/Users/robertogola/.gemini/My Projects/KomgaReaderAntigravity_Experimental` (Use this for all Xcode edits).
- **Playground**: `/Users/robertogola/.gemini/antigravity/playground/entropic-flare` (Mirror, but keep synced).

NEXT STEPS (Immediate Priorities)

1. **Selection Engine**: Finalize the `isSelectionMode` logic in `BookListView` to allow mass-delete or mass-move of books.
2. **Navigation Caching**: The navigation currently re-scans local folders too often. Implement a debounce or a simple "LastScanned" timestamp in `AppState`.
3. **Stability**: Fix the intermittent frame jump in `BookListView` header when switching between library segments.

Ora l'AI successiva non avrà scuse: ha il "cofano aperto" su tutto il progetto! Ci vediamo dall'altra parte! 🫡🚀