

## RePlex — Development Overview

### What RePlex is

RePlex is a **private, local-only, high-fidelity Plex client for Fire TV** that prioritizes:

- **Lossless playback**
- **Home-theatre correctness**
- **Custom UX**
- **No transcoding**
- **No cloud dependency**

RePlex **does not replace Plex Media Server**.

It replaces **only the client experience**.

---

### Core Problem RePlex Solves

Official Plex clients are:

- Generic
- UI-constrained
- Sometimes force transcoding
- Not tailored for serious home-theatre setups

RePlex gives you:

- Full UX control
  - Guaranteed Direct Play
  - TrueHD / DTS-HD / Atmos passthrough
  - A “this belongs on my TV” feel
- 

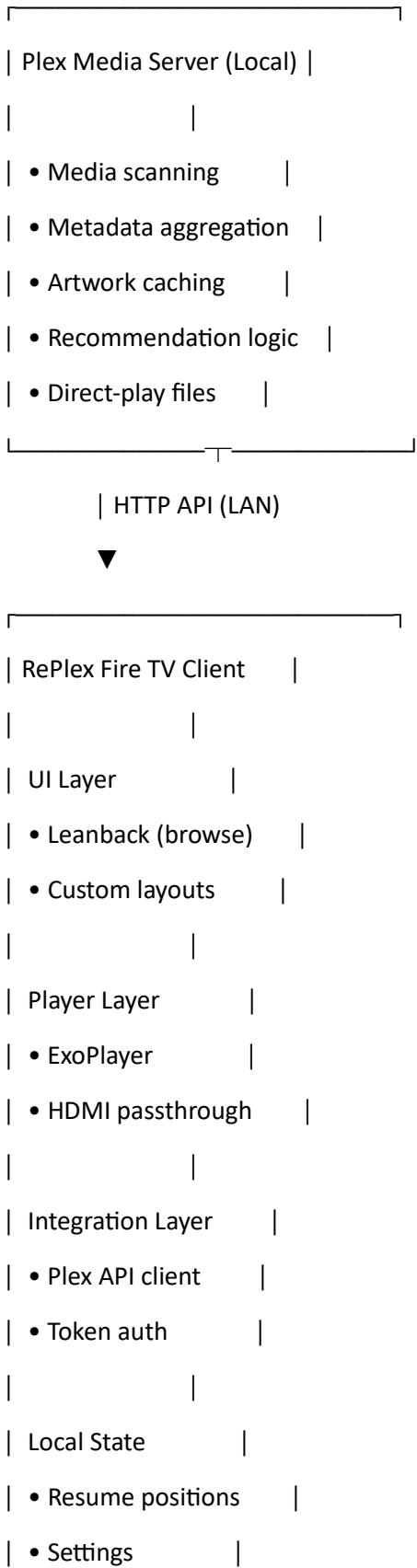
### Non-Negotiable Constraints (by design)

- **✗** No browser playback
- **✗** No Flutter for TV playback
- **✗** No transcoding
- **✗** No DRM bypass
- **✗** No public distribution
- **✗** No Plex server modification

Everything is **local, legal, and interoperable**.

---

## High-Level Architecture



---

## Technology Stack (Final)

### Server

- **Plex Media Server** (existing)
- Uses TMDB, fanart.tv, etc. internally
- Acts as API + media host

### Client (RePlex)

- **Platform:** Fire TV / Android TV
- **Language:** Kotlin
- **UI:** Android TV Leanback + custom views
- **Playback:** ExoPlayer
- **Networking:** OkHttp
- **Storage:**
  - SharedPreferences (token, settings)
  - SQLite / Room (resume, history)

### Dev Environment

- VS Code
- Gradle (gradlew)
- ADB over Wi-Fi
- adb logcat for debugging

---

## Why Kotlin + Leanback (and not Flutter)

### Flutter:

- Amazing animations
- Poor TV focus handling
- Weak passthrough integration
- Extra rendering overhead

### Kotlin + Leanback:

- Native TV focus system
- Correct D-pad behavior

- Stable playback lifecycle
- Used by Plex, Netflix, Prime

**For a Fire TV media player: correctness > flash.**

---

## **Playback Design (Most Critical Part)**

### **Playback Mode**

- **Direct Play ONLY**
- MKV remuxes
- No video conversion
- No audio conversion

### **Audio**

- Dolby TrueHD
- DTS-HD MA
- Dolby Atmos
- AVR handles decoding (passthrough)

### **Player**

- ExoPlayer with:
  - Default controls disabled
  - Custom overlay UI
  - SurfaceView rendering
  - HDMI passthrough enabled

If the AVR shows:

Dolby Atmos / TrueHD / DTS-HD MA

→ RePlex is working correctly.

---

## **UI Architecture**

### **Browsing**

- Android TV Leanback:
  - Rows
  - Grids
  - Focus management

- Netflix-style navigation

## Player UI

- Fully custom overlay:
  - Play / pause
  - Timeline scrubber
  - Audio & subtitle selector
  - Codec / bitrate display
  - Minimal cinema mode

Leanback is **not used for playback UI** — only browsing.

---

## Plex API Usage

RePlex talks to Plex **exactly like an official client**.

### Key Endpoints

- /identity
- /library/sections
- /library/sections/{id}/all
- /library/metadata/{ratingKey}
- /library/parts/{id}/file
- /hubs/home
- /:/timeline
- /search

### Client Headers (critical)

RePlex identifies itself as:

Android + FireTV

to prevent forced transcoding.

---

## Recommendations & Home Screen

RePlex uses Plex's **Hubs system**:

- Continue Watching
- Recently Added
- Because You Watched X

- Recommended

You can:

- Filter hubs
- Reorder hubs
- Rename hubs
- Mix in your own logic later

UX freedom without re-implementing recommendation logic.

---

### **Metadata & Artwork**

- Posters, backdrops, logos are all served by Plex
- Logos may originate from fanart.tv or TMDB
- RePlex never calls TMDB directly

Fallback logic:

- Logo exists → show logo
  - No logo → styled text title
- 

### **Development Workflow (Fast, Practical)**

1. Fire TV in developer mode
2. ADB over Wi-Fi
3. Build via Gradle
4. Install with `adb install -r`
5. Launch instantly
6. Debug with logcat + AVR display

Iteration time: **seconds, not minutes**

---

### **Legal & Safety Posture**

- Private, personal use
- No redistribution
- No monetization
- No Plex branding
- No DRM circumvention

- Uses Plex's intended client-server model

This is **interoperability**, not exploitation.

---

### **What RePlex Ultimately Is**

RePlex is:

- A **reference-quality home-theatre client**
- A **custom UX layer over Plex**
- A **clean, disciplined system design**
- Something very few people actually build correctly