



Complete Technical Report High-Fidelity Audio Player for Android

Lyra Bitperfect is an Android audio player designed from scratch with a single goal: delivering music to the listener with maximum possible integrity. Every design decision — visual and technical — is oriented toward an uncompromising playback experience. The application combines an analog-inspired interface — physical knobs, textured display, toggle switches, spinning CD — with a native Kotlin audio engine that eliminates all intermediate layers between the file and the DAC.

1. General Architecture

Lyra Bitperfect is developed in Flutter/Dart with a native Kotlin layer for low-level access to Android's audio subsystem. The playback engine runs entirely in Kotlin using ExoPlayer/Media3, with Flutter acting exclusively as the UI layer. Communication between both layers uses dedicated native channels:

1.1 Flutter ↔ Kotlin Communication Channels

Channel	Type	Function
lyra_app/playback	MethodChannel	Play, pause, next, prev, seek, shuffle, repeat
lyra_app/state_events	EventChannel	Real-time playback state, position and track index
lyra_app/metadata	MethodChannel	Metadata and artwork extraction (MediaMetadataRetriever)
lyra_app/bitperfect	MethodChannel	Bitperfect mode management and USB DAC detection
lyra_app/usb_events	EventChannel	Real-time USB connection/disconnection event streaming
lyra_app/rms_events	EventChannel	Full waveform + RMS streaming for instruments and VU meter
lyra_app/volume	MethodChannel	Native volume control — player and AudioManager

1.2 Playback Engine

The engine is built entirely in Kotlin using ExoPlayer/Media3. This eliminates the Flutter audio plugin layer entirely — no just_audio, no plugin bridges, no intermediate abstractions. ExoPlayer handles playlist management, track transitions, audio session, and hardware interaction directly. Fades on play/pause/next/prev are implemented as technical gain ramps on the ExoPlayer volume, never touching the signal.

2. Playback System and Library

2.1 Library Management

The library supports locally stored tracks organized across five views: Songs, Artists, Albums, Folders and Playlists. Metadata is extracted at import time via MediaMetadataRetriever and persisted in SharedPreferences as JSON.

2.2 Non-Interruption Principle

Browsing the library never interrupts playback. The track only changes when the user explicitly selects one. Returning from the library without selecting anything resumes playback exactly where it was.

2.3 Automatic Track Advance

When a track ends, ExoPlayer fires STATE_ENDED. The engine detects this and automatically advances to the next media item. If the last track of the playlist ends, playback returns to the beginning without auto-playing. Shuffle and repeat modes are handled natively by ExoPlayer.

3. Bitperfect System — The Core of Lyra

The Bitperfect system is Lyra's defining feature. It operates in three clearly differentiated states controlled exclusively by the BITPERFECT button. Each state is sonically distinct and indicated by the LED color.

3.1 The Three States

LED	State	Description
White/Pearl	Direct Playback	Pure Android audio chain — no EQ, no DSP, no intervention. Signal delivered as-is.
Green	Direct Mode	No EQ, hifi_mode=on via AudioManager. Maximum internal hardware transparency.
Blue	Bitperfect USB	USB DAC detected. Signal routed bit-to-bit to external hardware DAC.

3.2 Direct Playback — White LED

Standard playback with no signal processing of any kind. The Android audio chain runs completely unmodified — no equalization, no HiFi mode, no AudioAttributes optimization. The device delivers exactly what its standard audio stack produces.

3.3 Direct Mode — Green LED

Activated by pressing BITPERFECT. Lyra configures AudioAttributes to minimize system processing and activates hardware HiFi mode: AudioManager.setParameters("hifi_mode=on"). This eliminates digital attenuation and maximizes internal chain transparency.

3.4 Real Bitperfect USB — Blue LED

Activated automatically when a compatible USB DAC is detected. The signal bypasses the internal DAC entirely and is sent intact to the external hardware — no rounding, no mixing, no processing. Compatible with any device implementing USB Audio Class 1 or 2 (UAC1/UAC2). Detection performed at hardware interface level: `interfaceClass == 1`.

4. Reference Monitor — HEADROOM/PEAK and DYNAMIC/CREST

Accessible via the MONITOR switch, the Reference Monitor screen presents two precision analog-style instruments that provide real-time signal analysis. The screen features the full Lyra Reference Audio certificate and is designed as a physical instrument panel inspired by classic HiFi equipment.

4.1 HEADROOM / PEAK Instrument

Measures instantaneous peak level in dBFS. Scale: -30 to 0 dBFS. The needle uses fast attack (factor 0.60) and slow release (factor 0.12) to emulate the ballistic behavior of a real peak meter. The digital display shows headroom remaining to 0 dBFS. When no music is playing, the needle rests at the left stop (-30 dBFS).

4.2 DYNAMIC / CREST Instrument

Measures the crest factor: the difference between peak level and RMS level in dB. Scale: 0 to 24 dB. High crest factor indicates dynamic, uncompressed music. Low crest factor (loudness war) is immediately visible. The needle uses slow, stable movement (factor 0.08). RMS is calculated over a 500ms window.

4.3 Measurement Architecture

Android's Visualizer captures waveform at 20,000 Hz. Data is transmitted to Flutter as `[sample_0, ..., sample_N, rmsL, rmsR]`. A single broadcast StreamController in the player forwards this data to both the on-screen instruments (VU meter, goniometer) and the Reference Monitor screen simultaneously, avoiding EventChannel subscription conflicts.

4.4 Certificate of Direct Signal

The bottom section of the Reference Monitor displays a brass-style technical certificate — tappable to reveal a full-screen enlargement — declaring the complete signal architecture:

Field	Value
MODEL	LYRA HiFi Mobile Reference Player
ENGINE	Native Audio Core
DRIVER	ExoPlayer / Media3
PLATFORM	Android · Kotlin Native
SIGNAL	Direct Playback
CHANNEL	Stereo
BALANCE	Centered
DSP	None
EQ	None

5. Real-Time Visualization

Three visualization modes on the main display, accessible via double-tap with a 400ms animated transition.

Mode	Activation	Content
Spinning CD	Default state	Animated CD, track info, seekbar, playback controls
VU Meter	1st double tap	Two needles with real galvanometer physics
Goniometer	2nd double tap	Lissajous figure with phosphorescent trail

5.1 VU Meter — Real Analog Galvanometer

Two needles with real galvanometer physics: spring=0.08, damping=0.72, mass=1.0. Pivots calibrated to the millimeter: left at 18% width, right at 82%, height at 83%.

5.2 Goniometer / Vectorscope

Lissajous figure with correct 45° phase rotation: $X=(L-R)/\sqrt{2}$, $Y=(L+R)/\sqrt{2}$. Phosphorescent trail with opacity x0.94 per frame at 60fps. Color: #00FF88.

6. User Interface

6.1 Design Philosophy

Conceived as a high-end physical instrument. Aesthetic references: McIntosh, Marantz, Technics. All graphic assets designed in Photoshop exclusively for Lyra. The interface is not designed to look like an app — it is designed to look like a €5,000 physical object.

6.2 Volume Knob - Direct Touch with Mechanical Stops

Grab offset eliminates touch jumps; unwrapNear() ensures angular continuity; stopLock enforces hard limits. Progress ring is a fixed layer beneath the rotating PNG, visible through the transparent groove.

6.3 Seekbar

Custom seekbar with gold metallic assets. Drag updates position visually in real time. seekTo is throttled during drag (150ms) and sent as a final exact call on release. ExoPlayer confirms the new position immediately via onPositionChanged, eliminating visual bounce.

6.4 Precision Haptics

Interaction	Haptic
Prev / Play / Next buttons	Medium on press, light on release
Library button	Medium on press, light on release
Power button	Light impact
MONITOR switch	Medium impact
BITPERFECT button	Heavy on state change
Volume knob — stops	Heavy at min or max
Display double tap	Light on mode change

7. Trial System and Licensing

21-day trial with total blocking on expiry. Built on EncryptedSharedPreferences (AES256-GCM) with clock rollback anti-cheat.

Component	Description
TrialManager.kt	Encrypted timestamp + clock rollback anti-cheat
trial_gate_screen.dart	Full-screen block with gold certificate plate

Purchase seal	Tap area over wax seal image
launchPurchaseFlow()	Google Play Billing integration

8. Technical Specifications

Parameter	Value
Platform	Android 5.0+ (API 21)
Framework	Flutter / Dart
Native layer	Kotlin
Audio engine	ExoPlayer / Media3
Formats	MP3, FLAC, AAC, M4A, OGG, WAV, OPUS
USB DAC	USB Audio Class 1 and 2 (UAC1 / UAC2)
Audio capture	Android Visualizer — 20,000 Hz — waveform + RMS
Persistence	SharedPreferences (JSON) + EncryptedSharedPreferences (trial)
Fonts	DotMatrix, DSEG7Classic
Permissions	READ_MEDIA_AUDIO, RECORD_AUDIO, MODIFY_AUDIO_SETTINGS
Price	EUR 10.99 one-time payment - no subscriptions, no ads

9. Flutter Dependencies

Package	Version	Function
file_picker	^8.1.2	Audio file selection
permission_handler	^11.3.1	Runtime permissions management
shared_preferences	^2.2.2	Library and configuration persistence

"An interface that doesn't look like an app, but like a €5,000 physical object."

— Independent technical analysis