



Case Study: Reeldemo

AI-first music production — agentic Ableton sessions on an in-house sound-design platform

Julian M. Kleber | Founder | reeldemo.github.io

julian.kleber@sail.black | Last updated: July 19, 2026

Outcome snapshot

- Built an **agentic Ableton workflow**: reference track + intent in, full production session out — driven by highly innovative AI algorithms for compose, remix, and quality review.
- Shipped an **in-house sound-design and engineering platform**: commercial Studio plus MIT ReelSynth (own DSP engine) and Kaleidoscope (promo visuals).
- Shipped **ReelSynth v0.1.0** as a standalone app and export tools for macOS, Linux, and Windows.
- Designed Ableton delivery so AI accelerates production while the session stays producer-owned (clips, MIDI, and devices).

CONTEXT

Indie producers want AI speed without giving up authorship. Most music-AI tools either export a finished audio file you cannot reopen as a project, or hide every decision in a black box. Ableton Live is still where producers arrange and finish tracks — but getting from “this reference song + this idea” to structured, revisable session material is slow and hard to trust.

Reeldemo was built for producers who ship under deadline: keep creative control, review what the AI decided, and receive *editable* tracks, MIDI, and devices — not a finished master file.

PROBLEM DEFINITION

In typical AI-music workflows, teams and solo producers face:

- loss of edit authority when tools export only bounced (flattened) audio,
- opaque generation with no record of what the agent decided,
- brittle “reference audio → MIDI” pipelines that miss arrangement intent,
- separate tooling for sound design, session building, and promo visuals.

APPROACH

Reeldemo combines three tools under one production idea — **reference in, editable session out**:

- **Reeldemo Studio** — chat-driven compose, remix, and quality review; splits the reference into stems (separate drums/bass/etc.); hands results into Ableton Live.
- **ReelSynth** — MIT open-source synthesizer. The AI does not invent raw audio in a mystery model: stems are rendered through a readable sound engine you can inspect and export.
- **Kaleidoscope ASCII** — fast visual engine for landing pages, social ads, and CLI promo exports.

Studio tasks (copy, remix, create, grade) run against explicit contracts and automated quality checks before anything is sent to Ableton. The agent loop is deliberate: plan the arrangement → review layers → score quality → prepare the Live handoff.

SYSTEM OUTLINE

- **Studio agent layer:** orchestration for copy/remix/create, instrument plans per layer (drums, bass, chords, melody, effects), rubric scoring, and trace logs.
- **Stem pipeline:** separates a reference mix into remixable layers and steers arrangement.
- **Render engine:** ReelSynth (Rust) — wavetable synthesis with filter and modulation; available as standalone UI, export CLI, and headless renders for Studio.
- **Ableton handoff:** pushes scenes/clips into Live; Max for Live helpers write clips and drum racks; a Live 12 Extension can import a handover bundle (session JSON + audio files).
- **Licensing:** trial and paid keys for commercial Studio features.
- **Promo engine:** Kaleidoscope powers the org landing hero and exportable visual assets.

IMPLEMENTATION STACK

- **Agent orchestration:** Python services, schema contracts, commercial API/export endpoints.
- **Sound / instrument:** Rust ReelSynth engine with Python bindings; cross-platform standalone app and CLI.
- **DAW integration:** Ableton Live bridges (remote control, Max for Live tools, Live 12 Extension import).
- **Visuals:** Rust Kaleidoscope core; npm CLI and browser demo on reeldemo.github.io.
- **Web / waitlist:** static product site (Studio, ReelSynth, Kaleidoscope) with waitlist capture.
- **Licensing & auth:** database-backed Studio keys and waitlist.

OSS VS COMMERCIAL BOUNDARY

A deliberate license split keeps the sound engine reusable while protecting the agent layer:

- **MIT (ReelSynth):** sound engine, preset formats, importers (Vital, Serum, WAV), export CLI, Python render API.
- **Commercial (Studio):** AI compose, text-to-wavetable, session export, Ableton handoff orchestration, licensed API.

RESULTS (JUL 2026)

- **ReelSynth v0.1.0 shipped** — standalone synthesizer app and export CLI for macOS (ARM + Intel), Linux, and Windows; plugin waitlist open.
- **Three-product org landing live** — Studio / ReelSynth / Kaleidoscope story with waitlist forms.
- **Editable handoff design validated** — explicit compose plans, per-layer roles, and quality checks documented; AI does not render audio as an opaque blob.
- **Promo pipeline unified** — Kaleidoscope powers homepage hero and CLI/npm export paths.

Treat Studio conversion and user-count metrics as **early waitlist stage** until refreshed from production analytics.

WHAT THIS CASE STUDY DEMONSTRATES

- AI systems with **reviewable steps** — not one-shot black-box generation.
- Clean open-source / commercial split (MIT engine + licensed orchestration).
- Full-stack indie delivery: Rust audio, Python agents, Ableton extensions, marketing site, and CLI in one org.
- Domain authenticity — built by a producer who uses Ableton Live daily.

WHAT I WOULD IMPROVE NEXT

- Ship ReelSynth as a VST3/AU plugin so it lives inside the DAW like other instruments.
- Grow Studio early access with telemetry on successful Live imports and how much users still edit afterward.
- Harden quality rubrics with more reference-track regression fixtures per genre.

LINKS

- Org landing: reeldemo.github.io
- Studio: reeldemo.github.io/studio
- ReelSynth (MIT): github.com/reeldemo/reelsynth
- Kaleidoscope: reeldemo.github.io/kaleidoscope
- GitHub org: github.com/reeldemo