The thinking behind the desk

A lighting console is a machine for remembering. This one is built on one idea — *Masterpiece simplicity*: the power of a professional desk, but the screen does the remembering, so the operator can think about the show instead of the software.

01 The idea, and the problem it solves

MASTERPIECE SIMPLICITY

A full-featured lighting desk carries a lot of power, and the temptation is to put all of it on one screen at once. That's fast for the person who built it and a wall of switches to everyone else. This desk keeps the capability but organises it around how an operator actually thinks during a show: **define the rig, build looks, run the show** — three moments, kept apart, each showing only the controls that moment needs.

> **"Masterpiece simplicity — clear structures, but with the flexibility to do things your own way. The screen does the remembering."**

Three commitments follow from that, and every part of the desk is a consequence of one of them:

- **Discoverable over dense.** Nothing important hides two menus deep. A control says what it does; a "?" is always at the point of confusion, never in a manual.
- **One consistent grammar.** The same gestures mean the same thing on every screen, so learning one corner teaches the whole desk.
- **Safe by construction.** The states where you could damage a live show are kept structurally apart from the state where you run it, and one gesture blacks everything out, always.

Everything a professional desk does is here and software-verified. The care went into the surface a human touches — not into piling more onto it, but into making the power legible.

02 How light is actually made

THE SIGNAL MODEL — THE ONE DIAGRAM WORTH INTERNALISING

Everything the desk does ends as one number per channel, per universe, sent over the network many times a second. Understanding *how* those numbers are computed explains almost every design decision that follows. Several independent **sources** each have an opinion about a channel; a **merge** resolves them into one value; two **masters** scale it; and it goes out as Art-Net or sACN.

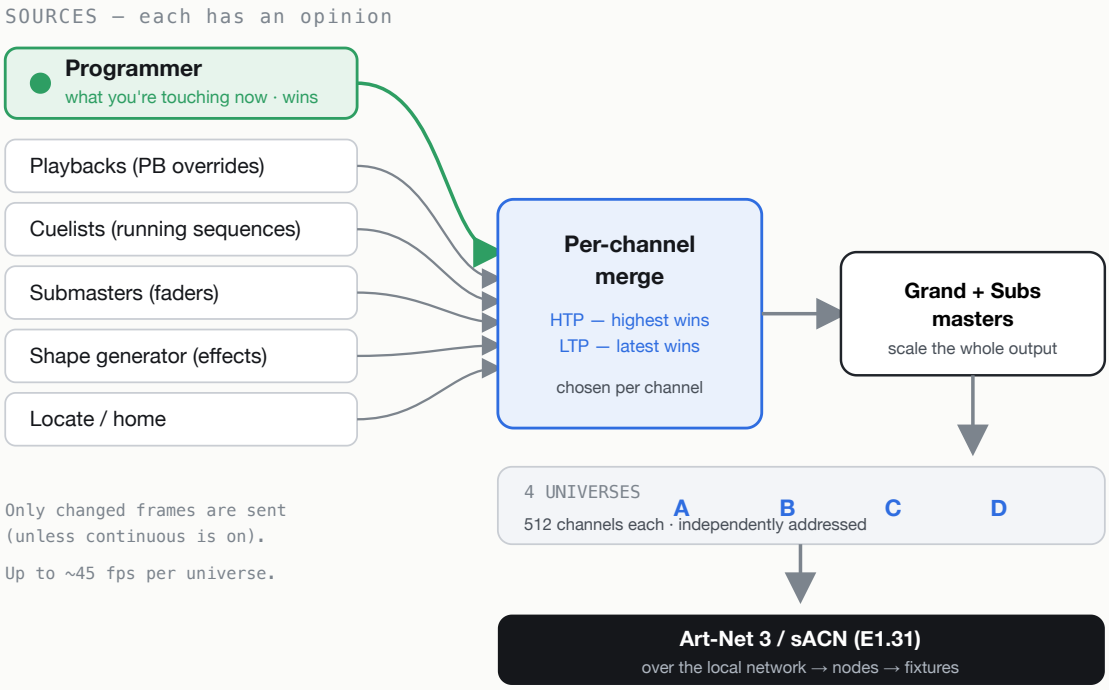


Fig. 1 — sources → per-channel merge → masters → four universes → the wire. The programmer always sits on top.

Programmer wins. Whatever you are touching right now overrides everything beneath it. That single rule is why the app feels direct: grab a channel and it moves, no matter what a cue or submaster is doing to it. **HTP** ("highest takes precedence") lets a submaster and a cue both contribute and the brighter one shows — the natural behaviour for intensity. **LTP** ("latest takes precedence") picks exactly one source — the natural behaviour for position and colour. Each channel is tagged with which rule it follows, so a mover's pan/tilt behave sensibly while its dimmer still piles up HTP.

03 One grammar for everything

LEARN ONE CORNER, YOU'VE LEARNED THE DESK

Every screen obeys the same four verbs and the same colour language. There are no per-screen exceptions to memorise — the point is that a gesture you learn patching a fixture works identically recording a look or running a cue.

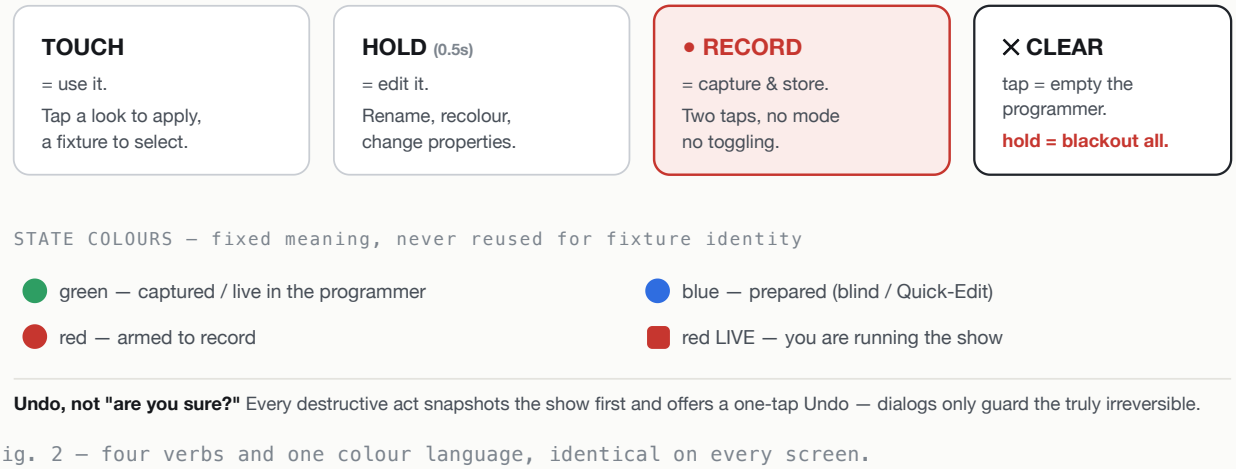


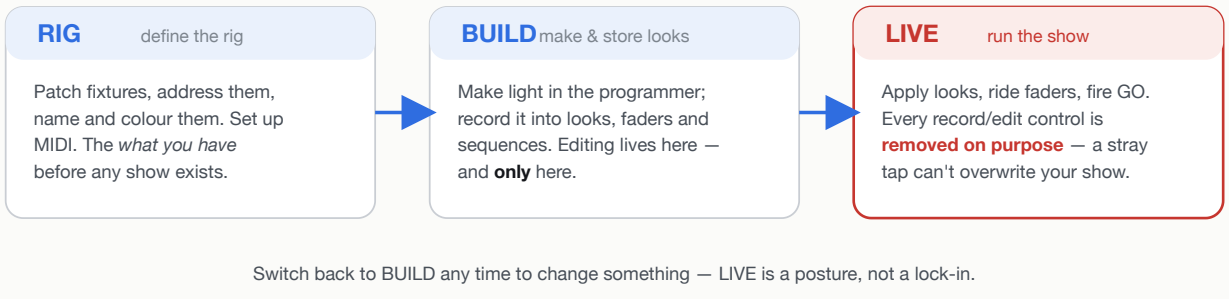
Fig. 2 — four verbs and one colour language, identical on every screen.

Two consequences are worth calling out. First, **CLEAR exists exactly once**, top-right, in every room — a tap empties the programmer, a *hold* is the panic that instantly blacks out faders, sequences, playbacks and anything applied. There is never a second Clear button to hunt for. Second, because destructive actions are undoable, the desk rarely interrupts you with a confirmation — it lets you work fast and take it back, which is how a real console feels.

04 Three rooms, and why they're separate

DEFINE → BUILD → RUN

It would be easy to show everything at once. Instead the desk splits the work into three rooms you switch with one tap, top-left. The split isn't cosmetic — it's the main safety mechanism.



Switch back to BUILD any time to change something — LIVE is a posture, not a lock-in.

Fig. 3 — the three rooms. LIVE deliberately has no editing surface, so the show can't be damaged mid-run.

This is the same instinct behind a physical desk's "blind" and "run" modes, made structural: in LIVE the app genuinely cannot expose a record or edit action, because the run surface doesn't implement them. The safety isn't a warning you can ignore — it's the absence of the dangerous control. When you need to change a look, you step back to BUILD; the show keeps running underneath.

05 The programmer, and the record workflow

ONE STATE, THREE WAYS TO PLAY IT BACK

The **programmer** is the heart of the desk: a scratchpad that holds whatever you're currently making. Touch a channel, dial a colour, position a mover — it lands in the programmer, sits on top of everything, and shows as a green bar. Nothing about it is permanent until you *record* it somewhere.

Recording is deliberately **two taps and no mode**. You don't switch the desk into "record mode" and back; you tap RECORD, the desk shows you every place a look can live, and you tap the one you want.

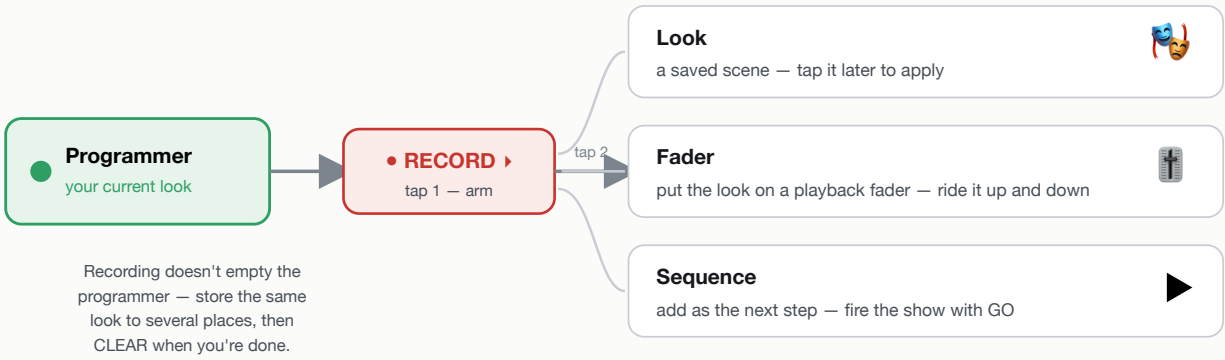


Fig. 4 — the same captured state records into a Look, a Fader, or a Sequence. One idea, three playback shapes.

That's the whole conceptual model of building a show: **make light → record it → it becomes a thing you can play back**. A *look* is a button you tap; a *fader* is that look on a handle you can ride and blend; a *sequence* is looks in order with fade and wait times, advanced with GO. They are three ways to replay the same kind of captured state, which is why the record gesture is identical for all three.

06 Playback: looks, faders, sequences

WHY THREE KINDS, AND HOW THEY COMBINE

Because intensity merges highest-takes-precedence, the three playback types **stack** rather than fight. Riding a submaster up while a cue is running raises only the channels the submaster is brighter on; the rest of the cue stays put. A look applied on top asserts its channels and holds them until CLEAR. This is what lets an operator layer a busking fader over a running sequence without either cancelling the other.

The two **masters** sit above all of it. The Grand master proportionally scales the whole output — the universal "bring it down" — and the Subs master scales the submaster layer alone. Both are always live, in every room, including LIVE.

- **Look** — an instant scene. Tap to apply over whatever's running; it latches until CLEAR. Best for states you jump to.
- **Fader** — a look on a handle. Ride it in and out, blend it with others. Best for elements you want manual control over live.
- **Sequence** — ordered steps with per-step fade and wait, plus Auto-GO follow and jumps/loops. Best for a show that advances on cue.

The panic is part of playback. No matter how many looks, faders and sequences are live, holding CLEAR releases every one of them instantly, with no fade. It's the one action guaranteed to get you back to black.

07 What's a show, what's the desk

TWO KINDS OF MEMORY

A clean split runs through the app's storage, and it matters the moment an operator loads a different show or moves to another iPad. Some things belong to the *show*; some belong to the *desk*.

Fig. 5 — the show travels; the desk setup stays. Your APC mapping survives a show swap; your patch doesn't.

This is why the app can **autosave from the first edit** without ever surprising you: a new show is given a working name immediately, so your programming has a home the instant you start. Save-with-a-name is how you keep separate shows and produce a portable file you can hand off or back up.

08 Honest edges

THE COMPROMISES, STATED PLAINLY

Running on iPad inherits a few of the platform's constraints. Rather than hide them, the design names them at the point they matter:

- **Output is unicast for now.** iOS gates default broadcast and multicast behind an Apple entitlement still pending, so this build reaches a node by its typed-in IP. It's one extra field, and it works exactly as well — but it's the single thing a first-time operator must do, so it leads the tester guide and has its own "?" in Settings.
- **No unattended running.** iOS won't let the app run in the background or start on boot; scheduling works only while it's foreground. Guided Access plus screen-always-on is the honest workaround for an installation.
- **One MIDI port in, one out.** Several APC minis need a hardware merger into a single port — the app talks to one surface, so the merger does the combining.
- **RDM is real but unproven on hardware.** The full controller is implemented and software-verified; it hasn't yet met a real gateway. It lives under RIG › Advanced, framed as a diagnostic tool, not a showtime control.

None of these change the model in 02–07 — they're boundaries around it. The desk you operate is the one described in this document; these are the edges of the room it runs in.

> **The whole design reduces to one sentence: make light, record it, play it back — safely, with one grammar, and let the screen do the remembering.**