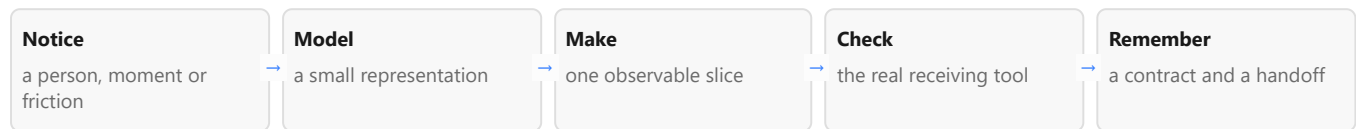


What I built. How I built it.

Not just a collection of apps: a workshop of instruments, games, design bays, film tools and ways to keep their promises true.

The first 18-page edition stays intact. This long edition adds MOLT plates and Auto DJ, connected TALON documents, Vantara models, product specs and paths from experiment to reusable system, without treating experiments as finished products.



The recurring shape; a failed check can return the work to any earlier step.

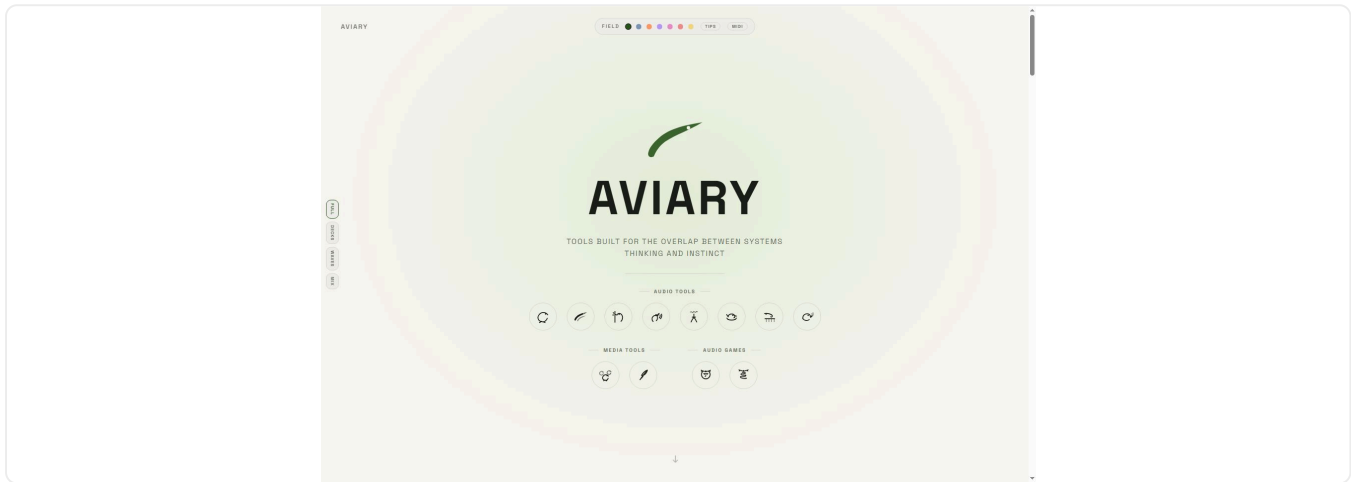
An editorial explanation of Arvind's project records, not a verbatim memoir. App screenshots are dated evidence of the visible surfaces, not proof of the entire product workflow.

I did not just make apps. I made ways to think.

A browser instrument, a game, a hero and a film might look like unrelated projects. For me, they share a question: **can I use AI to make something I care about, understand why it works, and leave enough behind to keep improving it?**

Chirp's [Summary of Approach](#) records the early nine-day AVIARY build, June 29–July 7. A single-deck player became a two-deck DJ instrument, then a family of musical tools. The motivations were wider than coding: practice planning, return to DJ culture and Carnatic percussion, explore synthesis, and reconnect with birds and nature photography.

The design principles were not invented after the screens. “Computers and the forest” paired systems rigor with living resonance. Stewardship meant preserving source material and making the next session easier. Those principles shaped the names, controls, themes and documentation.



AVIARY · live DJ booth/home captured October 13, 2026; the screenshot records a landing state, not playback.

This guide follows the work between the idea and the result

The interesting part is often the thing that is *not* the finished product: a signal graph, a data model, a design bay, a rough body, a reference sheet, a beat sheet, a simulation or a checking tool. Each lets me see something I could not see before.

How to read it: follow a case study and its visual evidence; the optional practice pages are at the end. The exercises and example briefs are suggestions for the reader, not a claim that every project followed them.

A first-person editorial retelling, not a verbatim memoir. Project results are attributed to their records, not rerun here. “Models” means working representations and prototypes—not AI models I trained. Basis: C1, B1.

A workshop of experiences, engines and working tools

A repo is a container, not a useful description of a creation. The collection contains at least three layers: things to experience, machinery that makes them possible, and tools that help me develop them.

The kinds of things in the workshop		
Kind	Concrete examples	What it lets me do
Playable instruments	Chirp's DJ decks, synths and sequencers; Warble, the bird synth	Hear an idea immediately; learn by touching controls.
Listening and reactive surfaces	Rookery's listening colony and meters; HALLIATUS's cockpit instruments	Explore music and translate signal into readable movement.
Games and interactive worlds	TALON's mecha tactics; SaffronRider's rhythm-fighting prototype; Phrasebloom's music puzzle	Test rules, timing, choices and the feeling of play.
Visual engines	Pixelscapes' seeded landscapes; shared musical timing and pitch helpers	Make a family of outputs from a repeatable foundation.
Creative IP and production assets	Saffron Rider's character rules, reference sheets, proxy, rigs, poses, sets and review cuts	Keep identity coherent while changing pose, medium or shot.
Design and inspection tools	Hero bays, Sprite Lab, portrait/icon factories, Scene Factory, Model and Print benches	Inspect the thing that matters before committing to a full production.
Memory and coordination surfaces	Tool Sheds, decision pages, project records, Bestiary and Watchtower	Find the current work, make choices and continue across sessions.

These are not all finished products. A working instrument, a useful factory, a prototype game and an unfinished film can all be real accomplishments while being at different stages.

The distinction matters for reuse. A game's hero is its IP; the sprite factory may help another game. A landscape is an output; a seeded painter is an engine. A review page is not the product, but it can make the next product easier to judge.

Basis: C1–C4, H1–H3, S1–S5, B2–B3. The wider portfolio is context, not a new priority ranking.

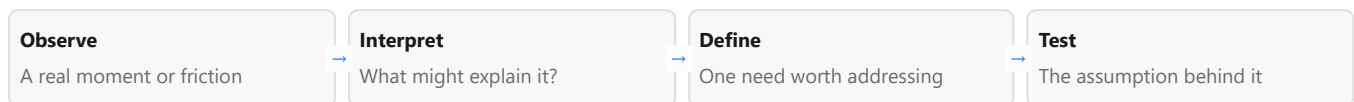
Start with a moment, not a feature list

“Make a music app” leaves almost every important decision open. Who is using it? What are they trying to hear or do? What makes the current way awkward? A clear moment gives AI something more useful than a broad category.

Bestiary’s ways of working already connect the build process to design thinking: start human-centred, explore different directions, build to think and test to learn. It also records a weakness: research can accumulate without becoming a clear problem statement. More material is not the same as a better definition.

Turn observation into a design question

Separate **what I know**, **what I assume** and **what I need to learn**. Spend time with the old way: watch a player without coaching, listen to a track through the current controls, or use your own tool for a week. A solo creator can be the first user, but that is not evidence of wider demand.



Return to observation if the explanation does not survive the test.

A worked example—not a claim of completed user research

Observation from the records: Rookery’s meters could be mathematically active but unreadable, or use only the bottom of their scale on real music. **Interpretation:** drawing a signal is not enough; the display must fit the material and the viewing size.

Design question: how might a listener see meaningful changes without learning the raw analysis machinery?

That question suggests a focused test: play the same material through the old and revised meter, compare usable range and label readability at a small size, and ask whether the listener can notice the change. It does not immediately justify a whole new dashboard.

Three lenses before investing

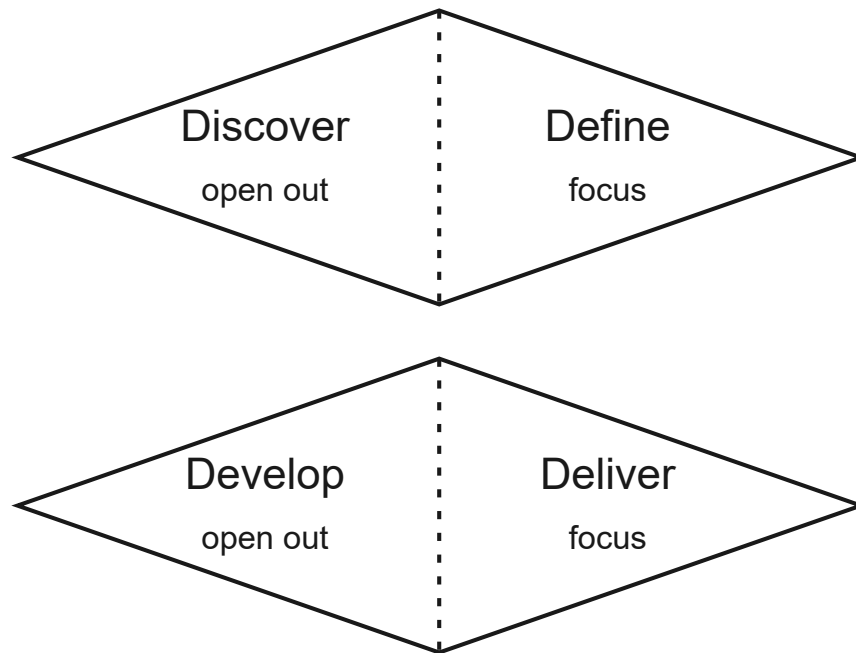
Wanted: whose need does this serve? **Possible:** what small test shows it can run? **Sustainable:** can I maintain it, afford its tools and use its assets legitimately? A blank answer is an uncertainty to investigate, not a box to quietly tick.

Basis: B1, C3. Discover/Define framing: Design Council; prototype/test framing: Stanford d.school. This is an iterative lens, not a certification or a claim of fleet-wide adoption.

Envision the whole; choose the next cut

Think like a sculptor: hold the intended shape in your head before chiselling. Spend time envisioning the experience and sketching a roadmap, then chip away in useful slices. The image guides the next cut; evidence from the emerging work can change the plan.

AI can make variations quickly, but five colourways are not five ideas. SaffronRider calls for genuinely different directions before locking one. Explore the whole; commit only the next bounded, reviewable result. A roadmap is a guide to choices and dependencies, not approval to build every possibility now.



Adapted from the [Design Council Double Diamond](#), [CC BY 4.0](#). Layout and labels simplified here. Explore broadly, then focus; repeat when evidence changes the problem.

Give the exploration a boundary

A brief names the receiving person, intended feeling, mechanical job, identity constraints and NOT-list. Ask directions to differ in interaction, rhythm, composition or production method, not just colour. Judge what each actually demonstrates; an attractive picture does not prove a control scheme.

Use a comparison the reader can actually decide on

HALLIATUS's decision-page learnings came from a practical failure: the planning document existed, but the owner could not conveniently open and review it. The response was a readable page with the finding, before/after diagrams, recommendations, choices and an explicit way to return the answers.

That page is itself an intermediate artifact. It turns a vague discussion into a decision without pretending the choice has already been made. A recommendation should say why it wins and what would be lost or costly to reverse.

Choose the first test before choosing the expensive tool

The recorded 2D-animation trial proposed the same tasks in Moho, Character Animator and Blender: build the silhouette, turn the head, perform a namaste-to-point gesture and make an idle loop. The rule favoured speed only if the visor and mustache still read. The document is a **trial plan**; it does not establish a winner.

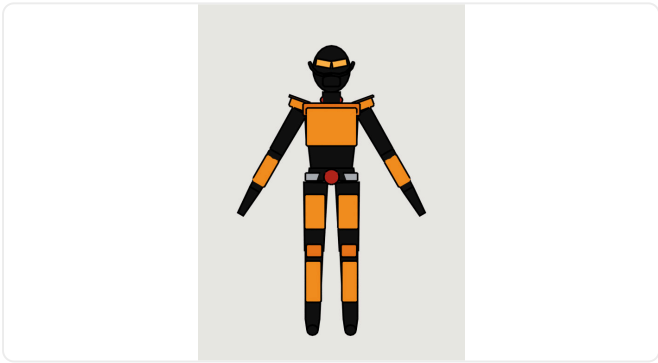
This is a useful habit: choose the common question and the pass condition before the tool. Otherwise, every tool will demonstrate its favourite trick and the comparison will tell you very little.

Choose a model for the question it can answer

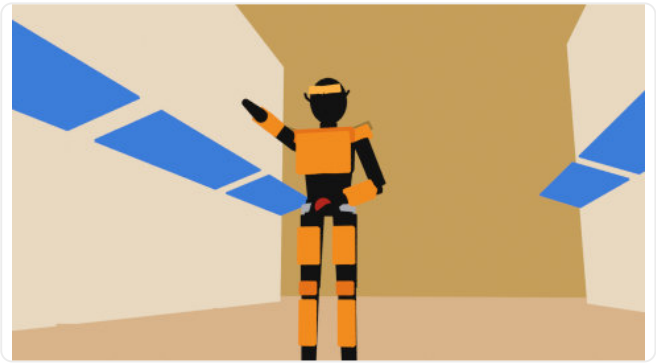
An intermediate model is a useful simplification. It leaves some things out so I can understand something else. A rough body is not a bad final character; a fixture is not fake evidence if it is honestly labelled and used for the right question.

Working representations I use

Representation	Question it answers	What it cannot approve
Design principles / codex	What must stay true?	Whether an interaction feels good.
Signal graph / schema	How does information move; what does a field mean?	Whether the screen is readable.
Design bay / clickable study	Does the form, control or visual vocabulary work?	All game behaviour or production integration.
Proxy / reference sheet	Do proportion, view and identity agree?	Rig weights or finished motion.
Rig / pose library	Can the asset move and be reused?	Scene pacing or mix quality.
Beat sheet / animatic	Do action, camera and sound read in time?	Final surfaces and full-size finish.
Fixture / simulation / frozen reference	Does the mechanism meet a repeatable condition?	Audience response or artistic approval.



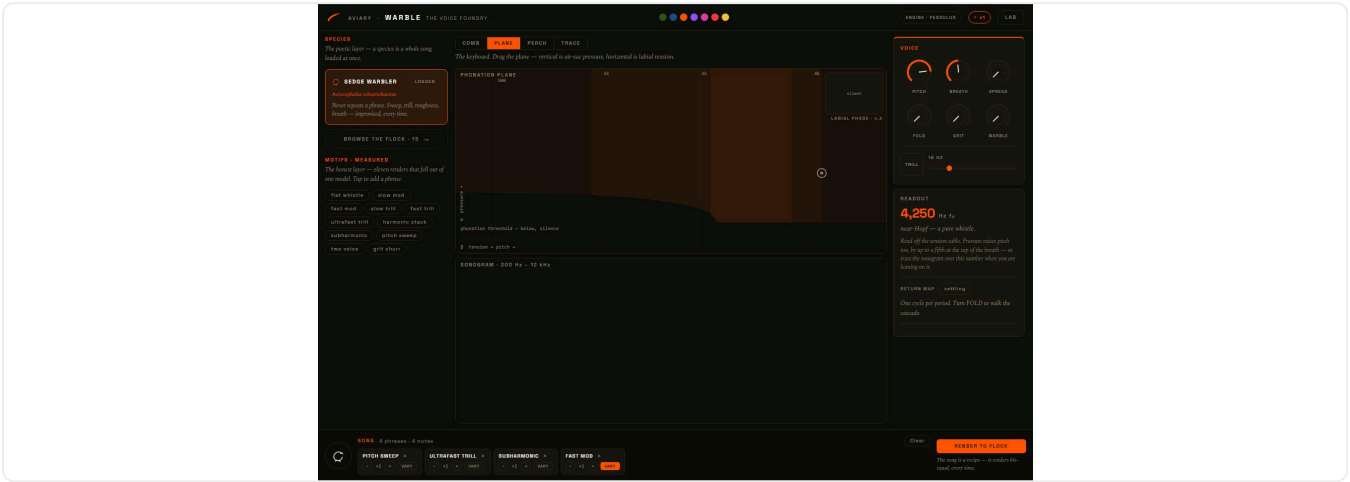
The proxy tests proportions, pose and camera. Detail is intentionally missing.



A pose and camera turn the same proxy into a shot question. These are early artifacts, not final looks.

Basis: C2–C4, H2–H3, S1–S5, B1–B3. Images: original SaffronRider proxy-front and K6-flat artifacts, resized for reading.

Give the tool ears before giving it a face

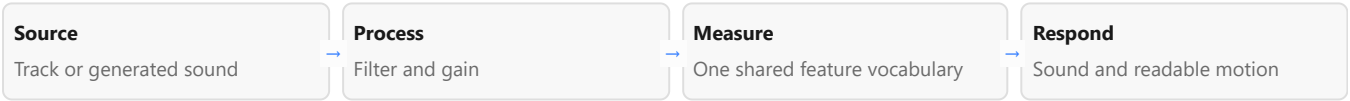


Warble · the voice foundry. The visible plane and sonogram express the signal model; audio was not tested here.

Chirp's AVIARY family includes decks, synths, sequencers and audiovisual experiments. Many instruments are self-contained browser pages. The family resemblance comes from shared principles; their different jobs come from different interaction and sound designs.

The first model is the sound path

The audio learnings start with a graph: source, processing, gain, analyser, output. Drawing that path makes the intended behaviour inspectable before the control panel exists. If the tool is silent, establish that a signal reaches the master before debugging the decorative end of the chain.



Schematic, not a complete wiring diagram: the analyser measures the signal; it does not replace the audio output.

The visual side receives named features—bass, treble or energy—rather than every raw frequency bin. This small intermediate representation gives several displays a common language. A control can change the sound while a meter helps the listener understand the consequence.

A real platform changed the architecture

Rookery's records describe a background-playback constraint on iOS: decoded-buffer Web Audio is suspended when the page backgrounds. The recorded response was a lifetime media element with a gesture-gated analysis graph, rather than treating lock-screen playback as a simple toggle.

Events such as play, pause and waiting drive the displayed state. A page should report what the player is doing, not what a button last asked it to do. This is a design lesson as much as a coding lesson: a user's expectation can reveal that the underlying model is wrong.

A green test can still hide an unreadable instrument

One Rookery pass drew every pane and measured the signals correctly, yet several scales lacked numerals and a reel looked empty. The lesson was to look at the screenshot, then add the check that would have caught the actual problem. Pixel activity proves drawing; it does not prove interpretation.

Basis: C1, C3–C4. These are recorded instrument learnings, not a fresh device test.

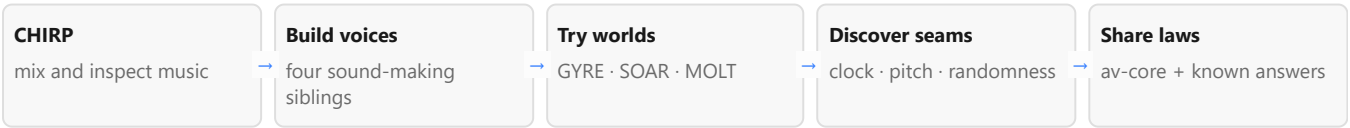
The flock grew different jobs, not copies of one screen

The early nine-day AVIARY retrospective begins with a player and DJ booth and follows the questions it raised: what if I could make the music, fly through it, see it change a world and save what happened? Different instruments became different answers.

What each instrument is for		
Surface	What it lets me test	A boundary worth keeping
CHIRP + PERCH	Mix beats; detect musical sections and loops	A detector can say “not confident”, not invent a section.
TRILL / SYRINX	Arpeggiator, scale-aware melody and a sound-design cabinet	The engine is reusable; the playable face stays musical.
BITTERN / GULAR	Rave bass, acid lines and exposed synthesis controls	A playground by design, not merely a preset picker.
CLUTCH / GIZZARD	808/909 circuit-model drums and an egg-grid sequencer	Synthesis rather than a hidden sample library.
FLAMEBACK / MEMBRANE	Two mridangam heads and a Carnatic tala grid	Physical/modal voice and rhythmic structure remain explicit.
GYRE	Free 3D flight through a musical sky; steer and latch sound	Flight handling, onboarding and musical feedback are one experience.
SOAR	Fly through the existing synth engines as a performance	Recomposes engines; not a new sound-generation family.
MOLT	Develop photographic/ambient worlds that move with music	A scene’s place and a track’s musical origin differ.
ROOKERY · WARBLE	Listen/inspect a colony; model and play bird voice	The listening surface and bird synth are Chirp parts, not standalone repos.
NIGHTJAR	A hidden game inside the AVIARY retrospective	An experiment, not the shared audio core.

The supporting workshop: SHRIKE slices beats; MYNA is an agent sandbox; QUILL is a review bench. These are not all synths. The project record separates instruments, listening tools and maintenance tools so their next steps do not get mixed together.

Why this sequence matters. GYRE and SOAR are not “just another synth”: existing rhythm and voices become a place to fly. MOLT extends the visual question into an atlas and darkroom. NIGHTJAR tests a hidden playful use of the same musical instincts. These roles explain what should be reused and what should stay different.



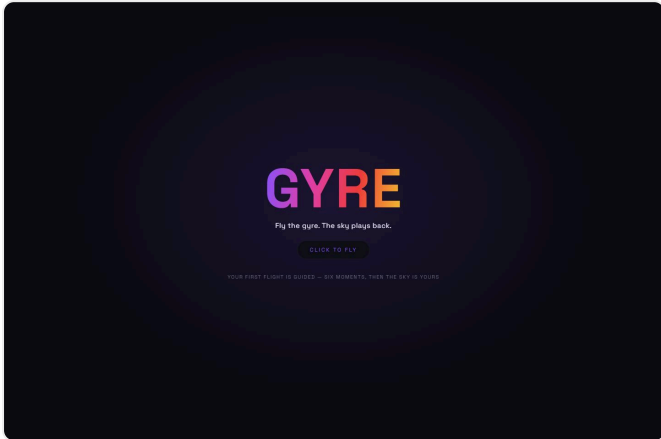
A conceptual sequence, not a claim that every instrument was built in this exact order.

The engines travel, but the receiving tool keeps its own interaction. That distinction—shared law versus a different face—becomes the AV-core contract later in this thread.

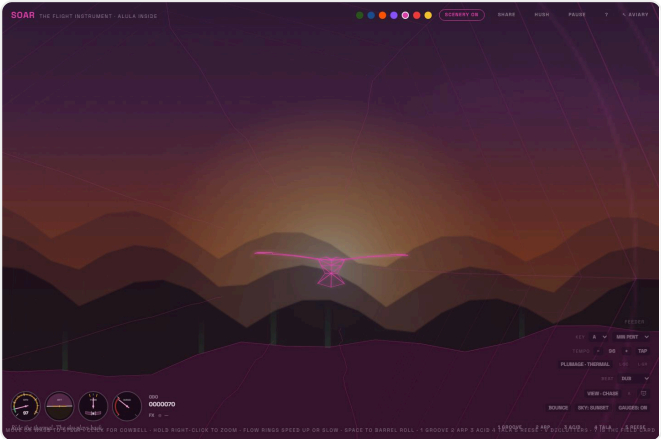
Chirp SUMMARY.md (early retrospective), docs/DESIGN-INDEX.md and docs/PROJECT.md at 9f02c7d. NIGHTJAR remains unlinked, as its project record specifies.

The music became a place—but the place stayed playable

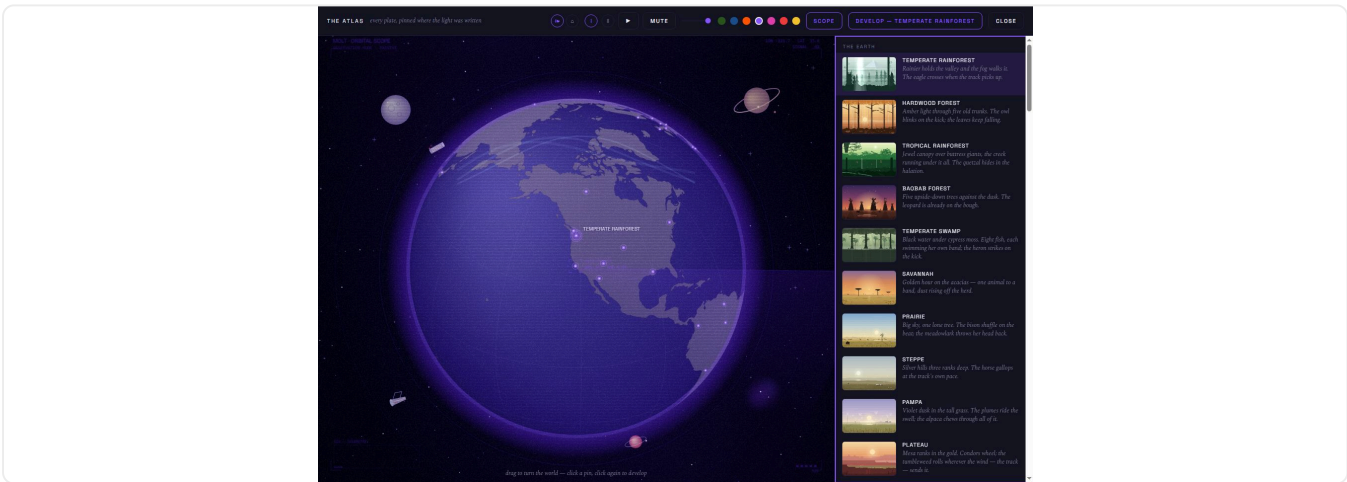
SOAR reassembles four sound engines behind a flight cockpit: altitude selects a scale degree, banking controls the filter and speed changes the note rate. GYRE explores free 3D flight and its handling, layers and onboarding. MOLT began as a photo darkroom and became a world that listens: bands, onsets and a real bar clock drive different features of its scenes.



GYRE · free-flight landing screen. The sky and sound layers share one interaction; no flying input tested here.



SOAR · reusable synthesis engines behind a cockpit, captured at rest. A latched-state bounce is not a recording of the live flight.



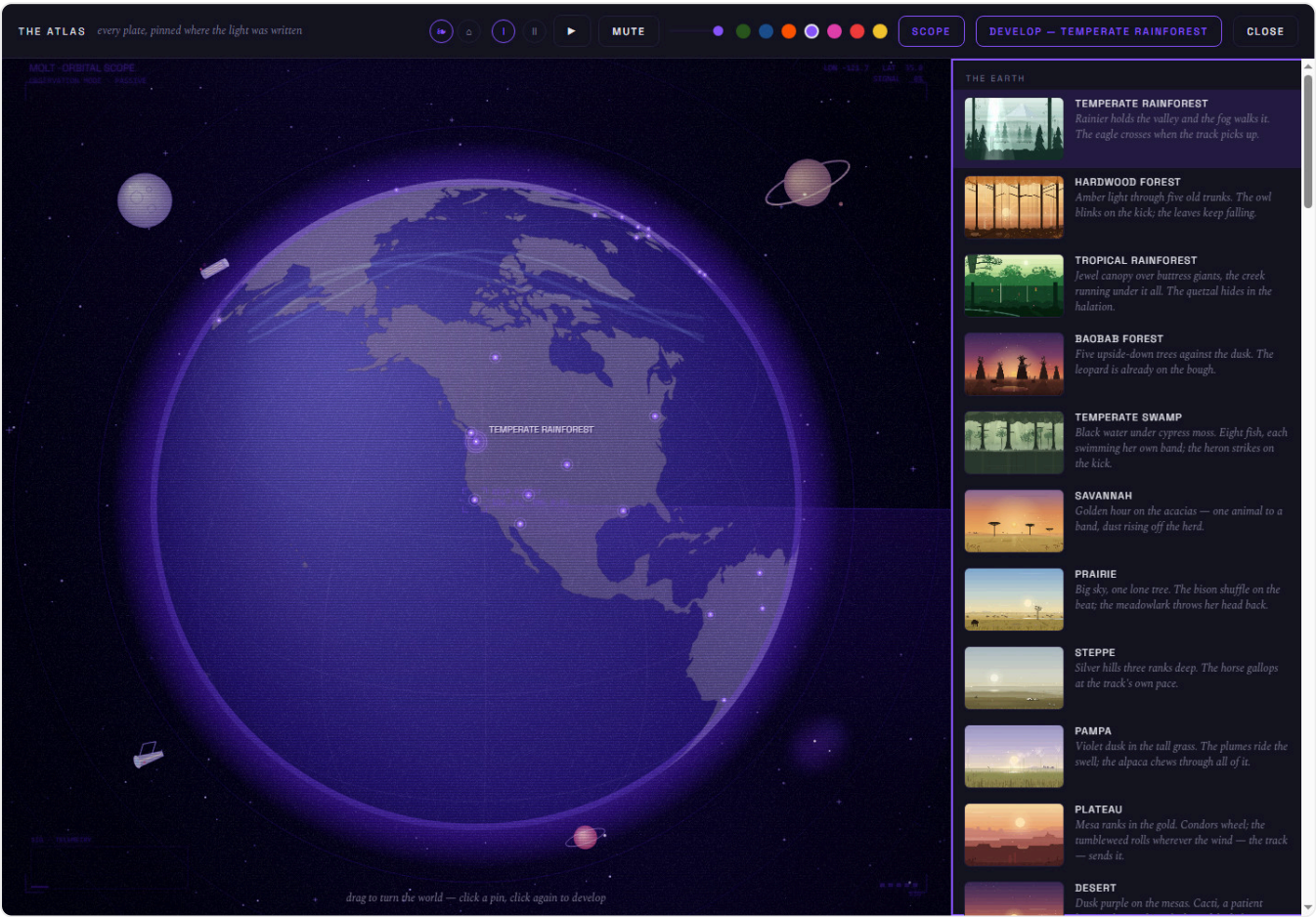
MOLT · a darkroom, atlas and reactive world. The selected place and song are different facts; the screenshot is a landing state, not a playback test.

The useful boundary: flight can be musically forgiving while remaining spatially different in each instrument. A world can react to a track without becoming the track’s geographical origin. The design question determines which behavior should be shared.

Chirp SUMMARY.md, docs/PROJECT.md and MOLT studio/atlas records at 9f02c7d. Public GYRE, SOAR and MOLT screens captured October 13, 2026; no flight, playback or export workflow rerun.

A globe is the entrance, not the whole instrument

MOLT turns a painted place—or an imported image—into a living picture. The Atlas is how I browse that visual world; the Feeder is how I find its music. A plate, a scene and a song are related, but they are not interchangeable.



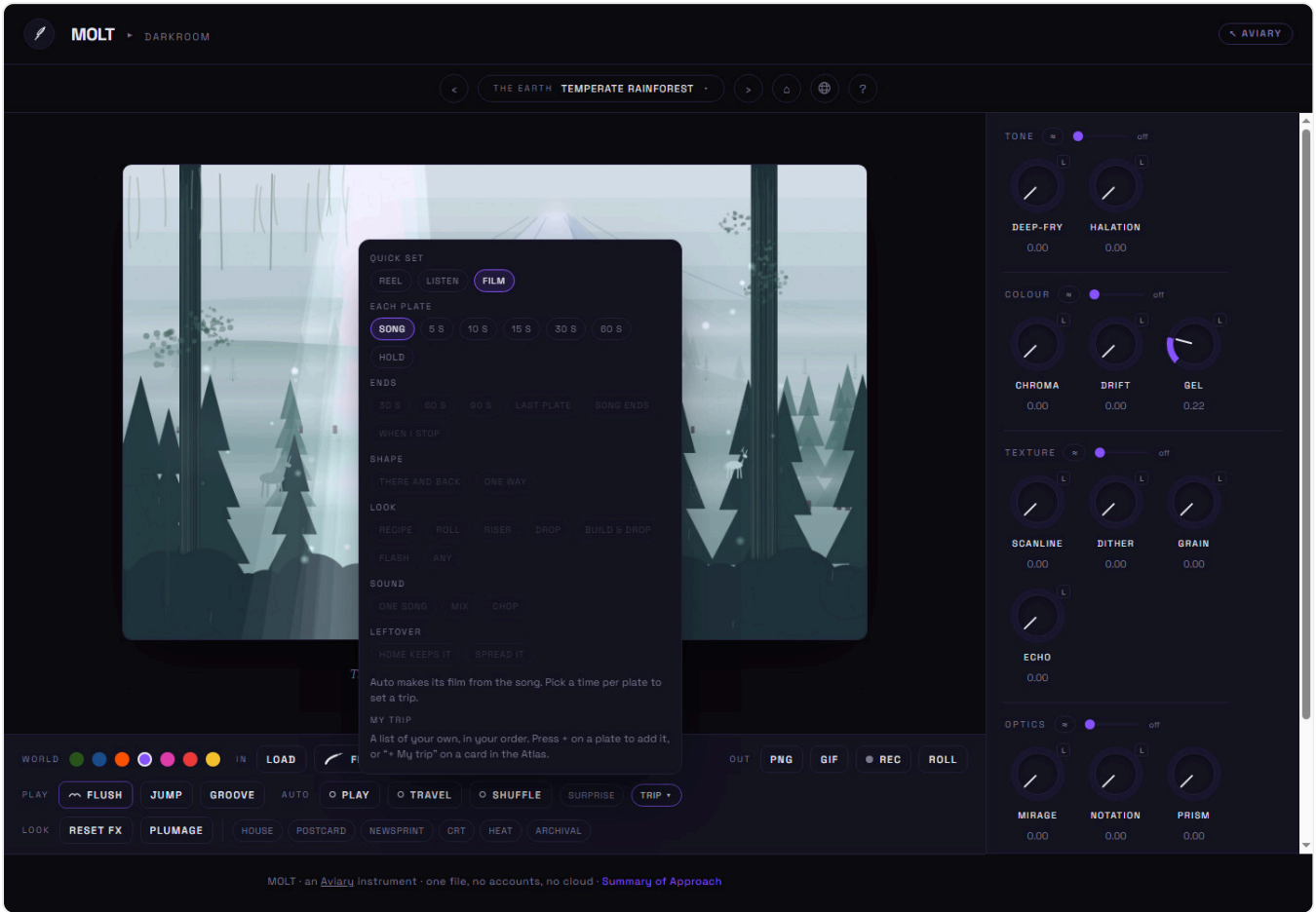
Public MOLT Atlas, captured 13 October 2026. A screen of the world/plate picker, not proof of audio playback or musical origin.

The three models behind the screen		
Model	Its job	Important distinction
Scene painter	Draw a place and its residents/effects.	106 scene painters in inspected source.
Plate association	Link numbered soundtrack plates to scene homes.	26 plate numbers map to 25 homes; this is not the full scene roster.
Song colony	Offer recordings and searchable music.	240 manifest tracks; a scene location does not establish musical origin.

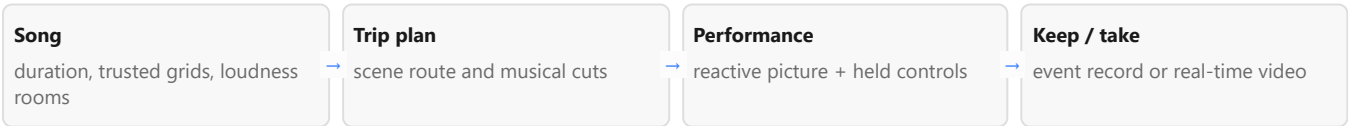
Source: Chirp 9f02c7d, molt.html, atlas-geo.json and the Rookery manifest. Counts describe source data, not a live catalog audit.

Music becomes a director, with room for my hand

Auto DJ is the useful shorthand: I choose a song and a trip mood—Play, Travel or Shuffle—then set pace, shape, sound and look. The director follows the lead song’s playhead and trusted music evidence. I can still hold or move an effect myself.



Public MOLT Trip sheet, captured 13 October before loading a song. Some controls await input; this still does not prove timing, replay fidelity or export compatibility.



PERCH’s song map supplies evidence; it does not guarantee verse/chorus/drop labels.

Mix versus Chop: Mix blends the incoming plate’s song; Chop changes it without that blend. Otherwise one song can carry the trip. A kept trip remembers events, not a self-contained media bundle or identical pixels. Auto needs a song—it cannot read microphone/tab input ahead.

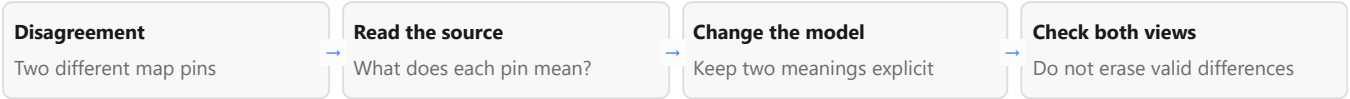
Source: Chirp 9f02c7d, tripPlan, setFx, mixCue and tripReplay in molt.html; songmap.json. MOLT’s director remains an intentional fork, not universal av-core adoption.

Sometimes the bug is a missing idea

Two screens disagree. The tempting AI task is “make them match.” But that can erase something true. Chirp's de-spaghetti notes record MOLT and Rookery disagreeing about map pins: one was showing where a scene was set; the other was showing where the music came from.

Before: one location trying to answer two questions

A scene could be painted in the Sonoran desert while its musical reference belonged elsewhere. Reading the source showed that most of the apparent pin errors were not wrong coordinates. The data model needed **scene location** and **musical origin**, not one winner.



A second anomaly exposed another model: several scenes can share a place. Treating scene and place as interchangeable had hidden entries. The fix was broader than patching the single missing record. This is design thinking in code: the exception teaches you what concept is absent.

Share the law, not every surface

The same records deliberately keep the two globe renderers different. MOLT's painted scene-picker and Rookery's listening scope do different jobs. Shared facts should agree; presentation may intentionally diverge.

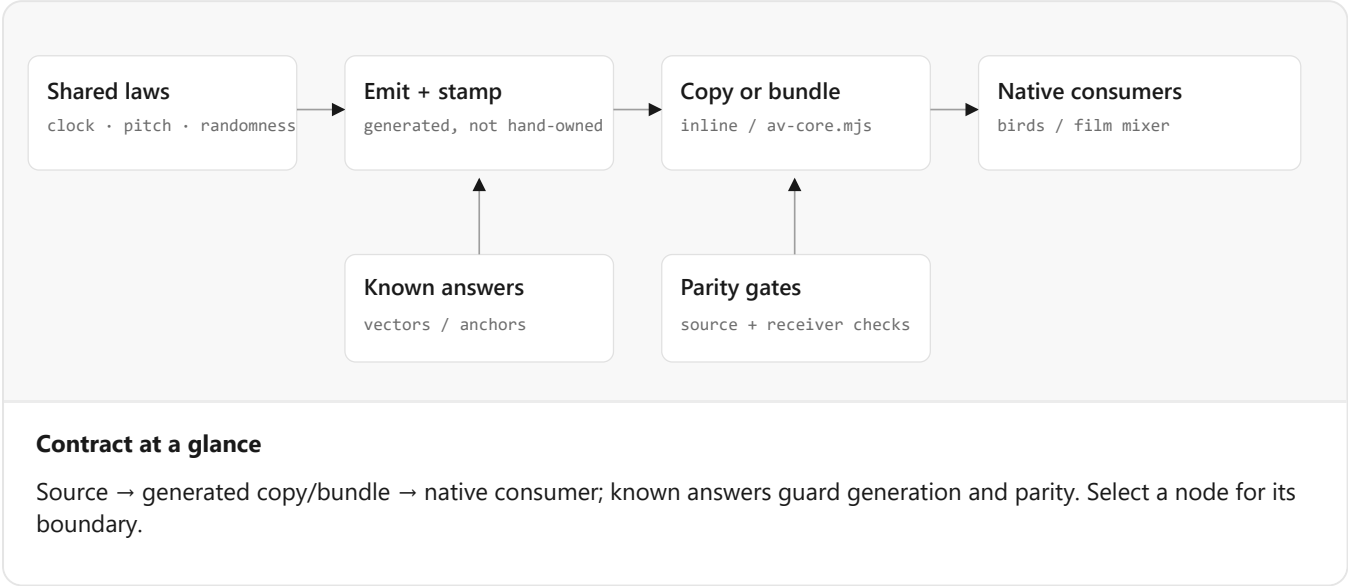
Chirp's **av-core** follows that distinction for sound. Timing, pitch, seeded randomness and musical-theory helpers have one hand-edited source, known-answer vectors and generated inline copies. Pages keep their own menu choices and adapters. Deliberate forks carry reasons instead of being silently overwritten.

The intermediate artifacts are not just a library: they are the vocabulary, the test vectors, the adapter and the generated-copy markers. They make reuse inspectable while preserving the self-contained browser-page format.

Basis: C2, C5. These are specific recorded fixes; they do not imply that all duplication across the fleet is resolved.

A family of instruments led to an explicit contract

After several birds needed the same clock, seeded randomness, pitch conversions and scale rules, the repeated code became a maintenance question. Chirp’s **av-core** keeps one hand-edited source of those laws. Known-answer vectors establish what a correct copy must do. An emitter generates stamped blocks; a checker compares the copies; an individual instrument still chooses its own controls, colours and voice.



A real known-answer example: the pitch vector says `midi: 69, cents: 0, expect: 440`. A copied helper returning 442 Hz fails. This answer also has an external standard (A4 = 440 Hz); many other vectors pin the donor’s present behavior rather than an independent truth.

A contract is more than a shared file		
Piece	Concrete role	What it cannot decide
Source	Core module is edited once, then generated into self-contained pages	Whether the page feels good to play.
Vectors	Known inputs and expected results; the gate can reject a stale copy	Whether two different voices should be merged.
Stamp + markers	Identify what is generated; a page’s adapter lives outside the block	How the host exposes controls.
Intentional fork	Document why a local rule diverges	It does not excuse silent drift.
Dynamics, 2026-10-12	A newer family from the SaffronRider film mix; measured bed → plan	No bird had adopted its block when the source note was written.

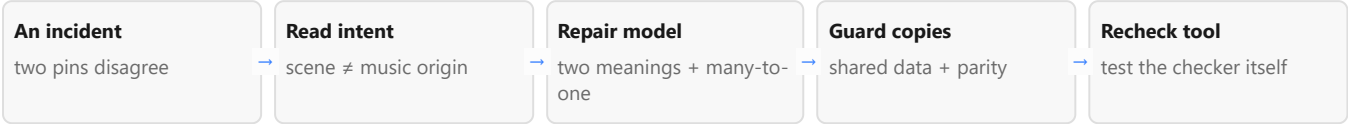
What is not shared: MOLT and ROOKERY both draw a globe, but one is a scene-picker and the other a listening scope. The data should agree; their drawing can differ. Likewise, a page can reuse pitch laws while retaining its own scale menu and presentation. The shared core is a rule about meaning, not a demand for identical interfaces.

MOLT’s current director keeps its own RNG and timing path. A family resemblance does not mean every page consumes every shared module.

Chirp `av-core/README.md` and `tools/vectors/av-core.json` at `9f02c7d`; `docs/learnings/despaghetification.md`. Vectors retain `gesture_synth` provenance. Dynamics arrived after the original guide; consumer adoption is not inferred from its existence.

The Janitor catches promises that stopped being true

When more instruments and documents appeared, the hard part was no longer generating another page. It was knowing which fact belonged where and whether a copy still matched. Chirp’s **garbaggio** (the Janitor) translates actual drift incidents into checks. It runs read-only by default, reports findings and exits nonzero on blockers. Its optional fix mode only addresses a narrow safe subset; ambiguous cases remain human decisions.



The map-pin incident is documented; this is an editorial abstraction of how the repo matured.

Four kinds of rot and the response		
Observed issue	What the first guess missed	What kept it from recurring
MOLT / ROOKERY map pins disagree	Location of the scene is not the origin of its music	Name two coordinates; shared atlas data + LOCKSTEP check.
Changelog claims an archive exists	An entry can name an output that was never saved	ARCHIVE check resolves declared snapshots to files.
A doc calls future behavior “shipped”	Plausible prose is not a product test	Compare current source and status; correct the record.
Copied audio block changes in one bird	Self-contained pages need deliberate duplication	VECTORS and AVCORE checks reject mismatched generated laws.

The Janitor itself needed correction: a version comparison treated v0.20 as smaller than v0.9, and its archive matcher once counted prose describing the rule as a real archive claim. This is the important part of the story: a check is evidence only after it survives a real false-positive test.

Not the same Janitor: Bestiary also has a separate, scheduled fleet-operations janitor for claims and agent PRs. Its scheduled workflow is dry by default until enabled. It is a different program and is not evidence that all fleet repos have adopted Chirp’s checks.

Chirp docs/learnings/despaghetification.md and tools/garbaggio.mjs at 9f02c7d; Bestiary canon/ci/janitor.mjs and .github/workflows/janitor.yml. Source inspection only, not a fleet-wide deployment audit.

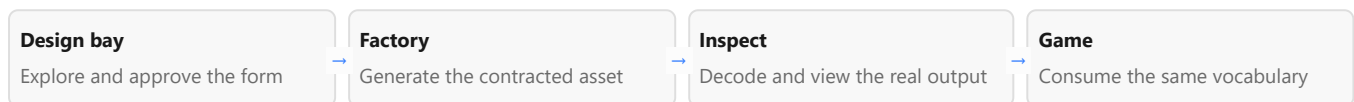
Build the workshop around the game

HALLIATUS holds TALON, the mecha tactics game, and a studio for shaping its parts. The bays are not decorative extras. They create a place to inspect a character, sprite, portrait, icon or cockpit control without needing the whole campaign to be running.

From a design study to a production contract

Hero and villain bays establish silhouettes and visual rules. The Interface Foundry explores mecha panels, vintage meters, cyberdeck readouts, a synth and runway timing. The Icon Foundry is a separate source for glyphs; the shared design system supplies named components and tokens.

Different outputs have different dispositions. A self-contained study can ship as a working lab page. Something embedded in the game must lose external design-runtime dependencies and match the game's own vocabulary. Recording which route an export took prevents “the mock-up worked” from being mistaken for integration.



The sprite grid became a bridge

The recorded mech factory ports bay painters from a 256-unit design grid to a 128-pixel sprite-cell contract. The relationship is explicit, not guessed. Checks cover foot contact, player-core visibility and colour restrictions. The Sprite Lab mirrors real game assets, so inspection does not drift onto a separate mock set.

The portrait factory likewise derives a vector registry and palette rules from the approved lab. That makes the intermediate design source useful at several resolutions without baking labels and colours into every output.

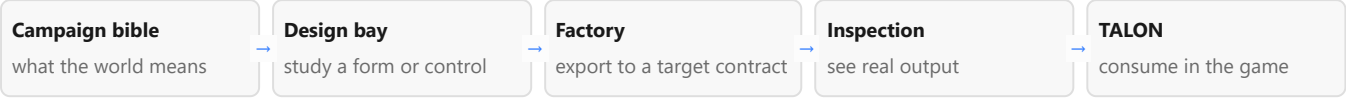
The PMO surface closes the loop

The gated Tool Shed is a project desk: **Status** reads the project record, **Docs** exposes the working knowledge, and **Bays** opens the tools. The design index tells me where a study went. This connects making to managing: not just “what did we generate?” but “where does it belong, and is it used?”

Basis: H1–H2. Lab studies and production features are distinct; factory availability does not approve every generated asset.

The bible gives the bays a world to serve

HALLIATUS is both TALON, a mecha tactics game, and the workshop that shapes it. The campaign bible and project record keep the world and work legible. The Tool Shed organizes **Status**, **Docs** and **Bays**: the project state is read from the project document, not copied into a dashboard. The design index records which study became which production asset.



A navigation map of sources and destinations; an inspected bay is not a shipped campaign feature.

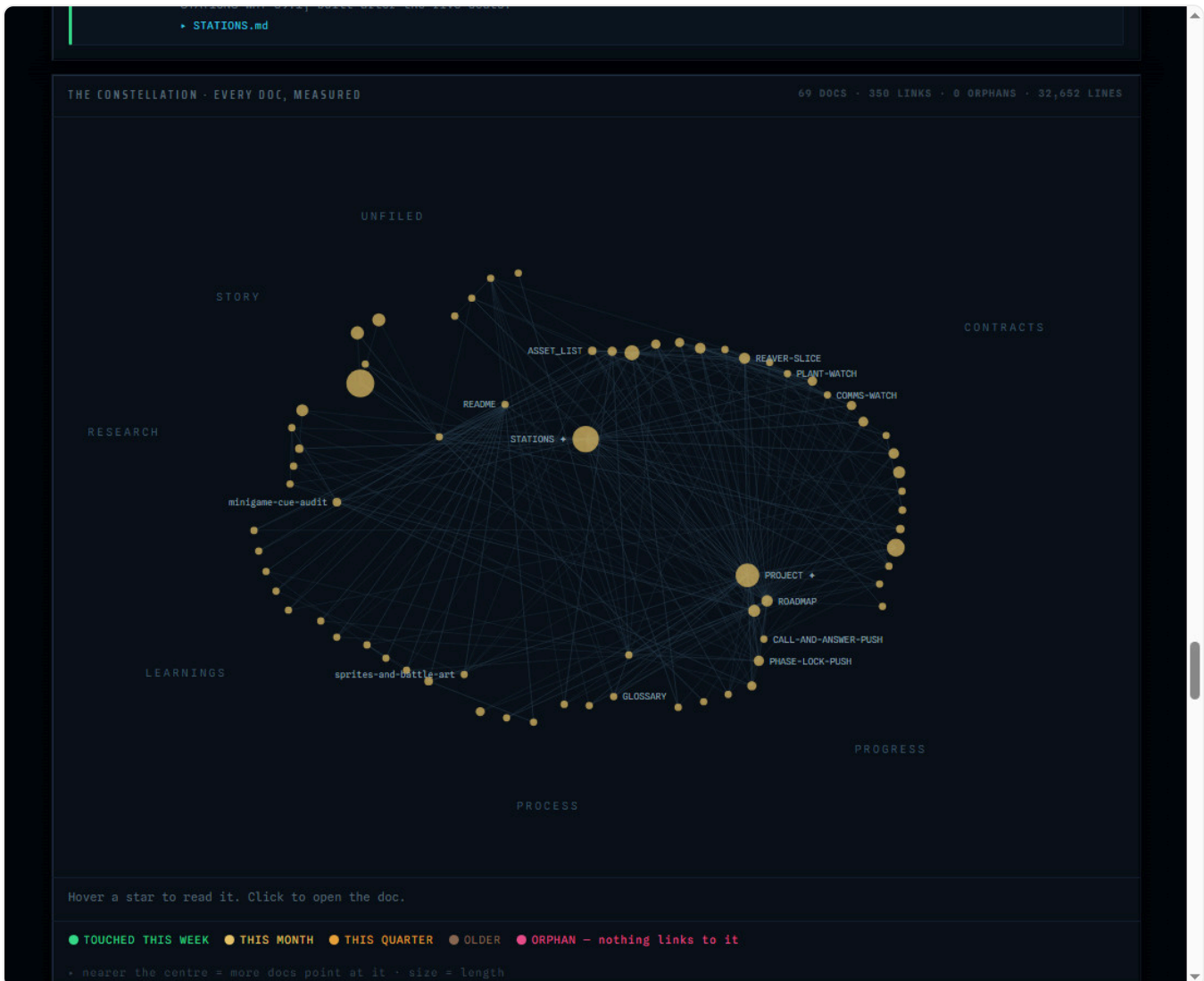
Where a design goes		
Working surface	The output it informs	Boundary
Campaign bible + project record	Story, factions and a current work list	World intent is not a proof of gameplay.
Hero / interface / icon bays	Character studies, readouts, glyph decisions	A lab page can stand alone; a game export needs the game contract.
Mech sprite factory	A 256-unit painter mapped to a 128-pixel sprite cell	Check foot contact, visibility and palette in real game assets.
Sprite Lab + BAY-09	Inspect sprites and compose a stage shot	A single shot does not prove an animated multi-beat scene.

The diagram says why a PMO belongs in a creative process: a project record lets the Status surface show what really exists, Docs explains why, and Bays lets someone inspect or make the next piece. The gated Tool Shed was not screen-captured for this edition; the visible public Scene Factory is shown in the following chapter.

HALLIATUS README, lab/index.html, docs/PROJECT.md, docs/DESIGN-INDEX.md, docs/MURMURATION-campaign-bible.dc.html and docs/learnings/factories-and-pipelines.md at f60371e. Gated Tool Shed not independently retested.

The dot cloud makes the workshop's memory visible

I had contracts, plans, research and story documents—but a folder alone could not show which ones carried the project. The **Document Constellation** is generated from the files and their links, so the map does not become another hand-maintained list.



Local preview of HALLIATUS source f60371e. Its own generator measured 69 docs and 350 links; node positions are navigation cues, not a correctness score. The live lab remains protected.

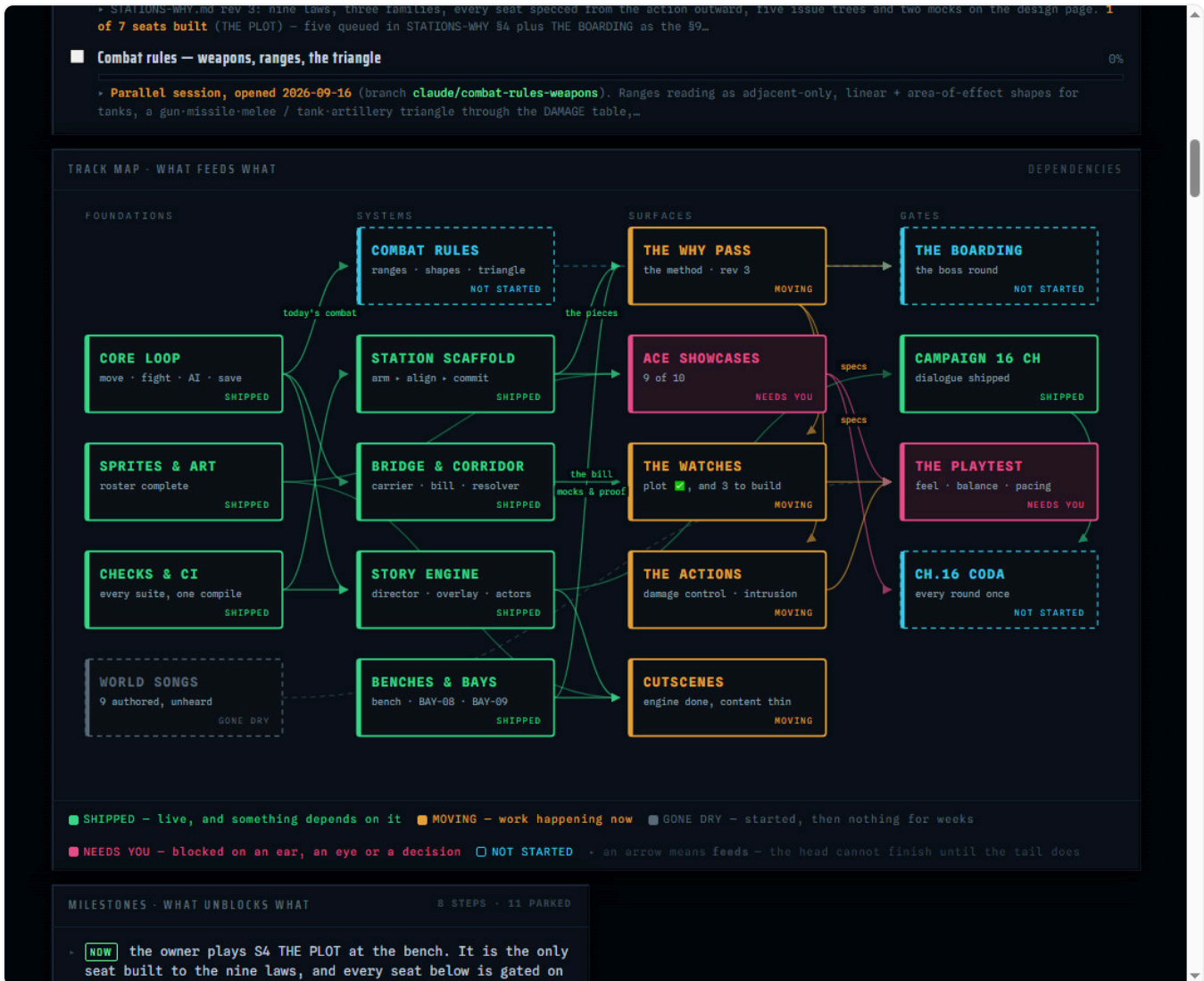
Read the picture: more incoming links pull a document inward; length changes its size; last committed touch changes its colour. A dot can open the source. Orphans would sit on the rim. The filing ledger separately identifies live canon, reference, superseded and historical material.

The model can show connectivity and declared status. It cannot tell me whether a document is correct—or whether a feature in it actually works.

Source: HALLIATUS f60371e, scripts/gen-doc-graph.mjs and lab/index.html. Local preview reuses pinned git history; no protected production access or campaign save was changed.

A useful “decision tree” shows what must unblock what

The actual surface is called the **Track map**. It is not a branching game choice: it connects foundations, systems, surfaces and gates. The arrows mean “feeds”; colour and text say whether a track is shipped, moving, gone dry, waiting on me or not started.



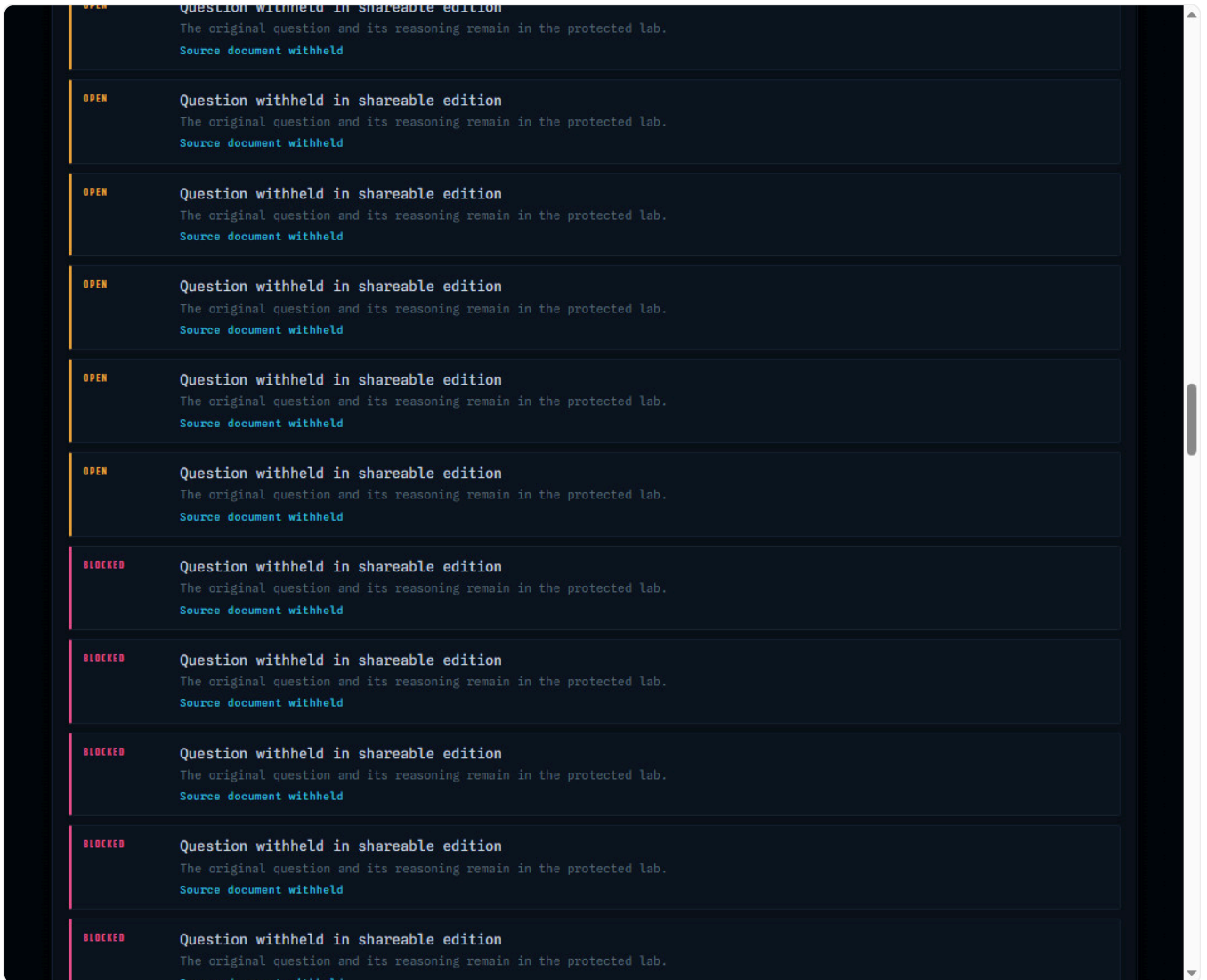
Local source preview, f60371e: Foundations → Systems → Surfaces → Gates. Shipped/moving/needs-you labels come from authored project docs, not a fresh test audit.

A blocked result can become a smaller question: do we need a rule, a visual cue, an ear pass or an owner choice? The map helps route the question to the right layer rather than adding another general to-do list.

Source: HALLIATUS f60371e, docs/PROJECT.md and lab/index.html. This is the Tool Shed's PMO-like surface, not a separate PMO application.

The quest list keeps unfinished decisions in the documents

Questions stay next to the reasoning that produced them. A visible Markdown marker gives each one a state and a track; the Quest log reads those markers into one filterable surface. There is no second database to reconcile.



Redacted local source preview, f60371e. The Quest log derives questions from Markdown markers; question text is withheld because the lab is protected.

A real marker shape, with editorial example text

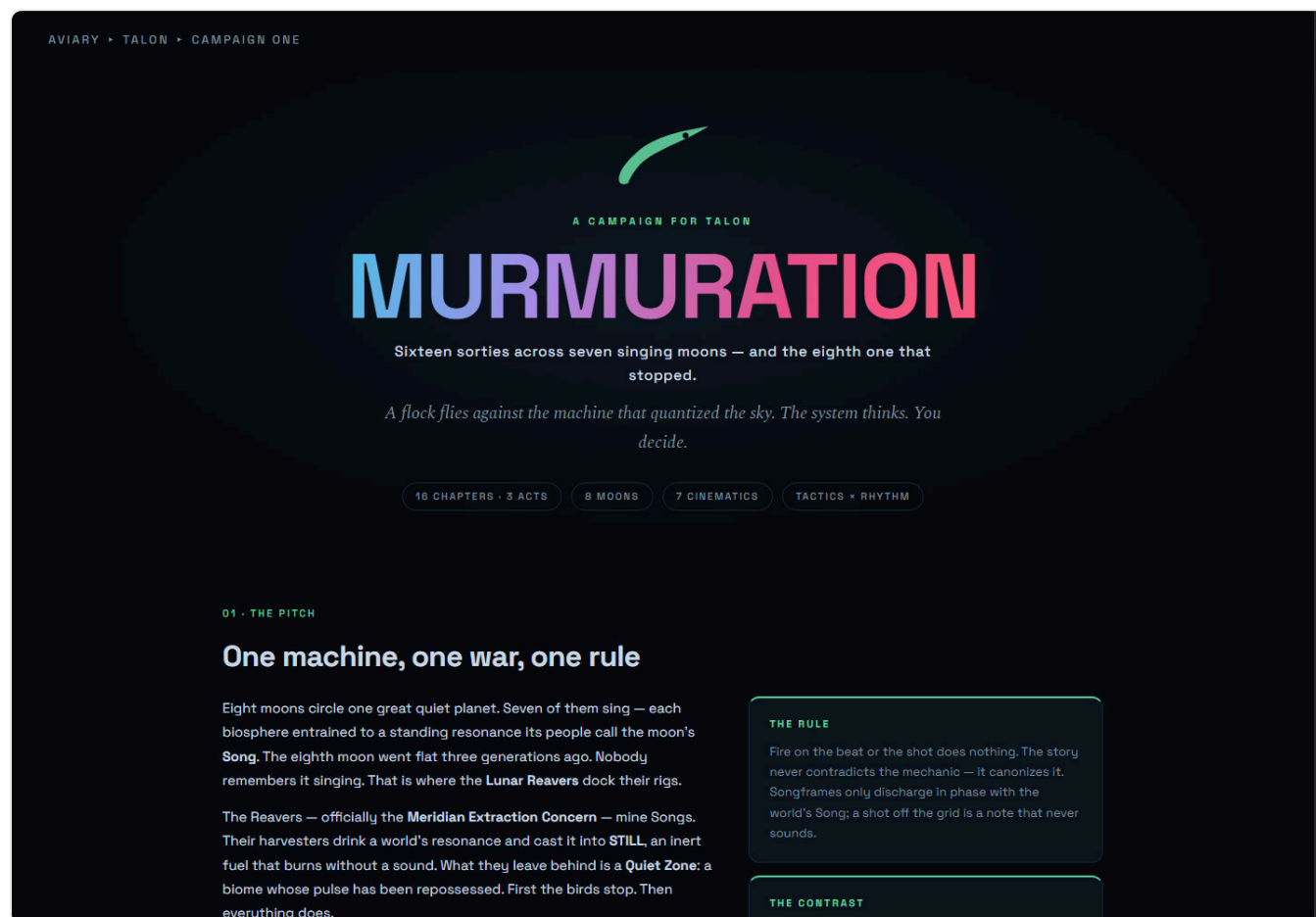
```
**QUEST · OPEN · cutscenes** – Which shot needs the next readability pass?
Keep the evidence and decision in the source document; the log collects it.
```

At the inspected revision, the six configured source documents contain **43 markers: 22 open, 7 blocked, 12 decided and 2 ideas**. Those numbers describe authored questions—not a completion percentage.

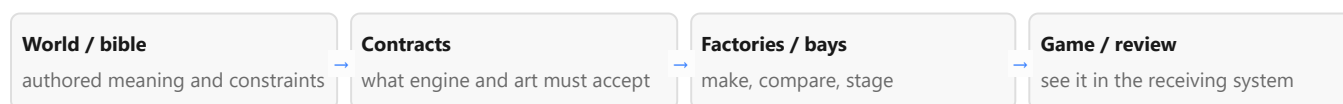
Marker syntax and parser: HALLIATUS f60371e, lab/index.html and docs/PROJECT.md. Example question above is editorial, not a copied owner decision.

The campaign bible is a readable model of the game

A campaign needs more than combat code. Its bible lets me inspect world, characters, arcs and operations as a reading surface; factories and bays then help produce the assets and experiences that express it.



Public Campaign Bible opening, captured 13 October. This is a story/operations reading surface; its older mechanics prose has a documented canon discrepancy.



Conceptual production flow, not a claim that every canon disagreement is resolved.

The bible, sprite factories and Scene Factory do different jobs. A sprite sheet provides poses/frames. A stage beat places them in a shot. The current Scene Factory exports one composition; a multi-beat editor remains parked.

Source: HALLIATUS f60371e, MURMURATION-campaign-bible, sceneBay.ts and SCENE-FACTORY-SLICE.md. Public opening only; no restricted story content reproduced.

Rehearse the shot with the pixels that will ship

BAY-09, the Scene Factory, is a cutscene composition bench. I can choose a backdrop, place battle sprites, props and symbols, drag and resize them, change an actor's animation and order the cast. It uses the game's stage painter and its 768×432 frame—not a convincing but unrelated mock-up.

The composition is the intermediate model

The export is a **StageBeat** snippet: a plate stack and actor entries in the game's own scene language. That lets a visual choice cross into production without being redescribed as “roughly here, about this big.” An author still integrates it; it is not a rendered movie or automatically published scene.

A still shot cannot prove pacing

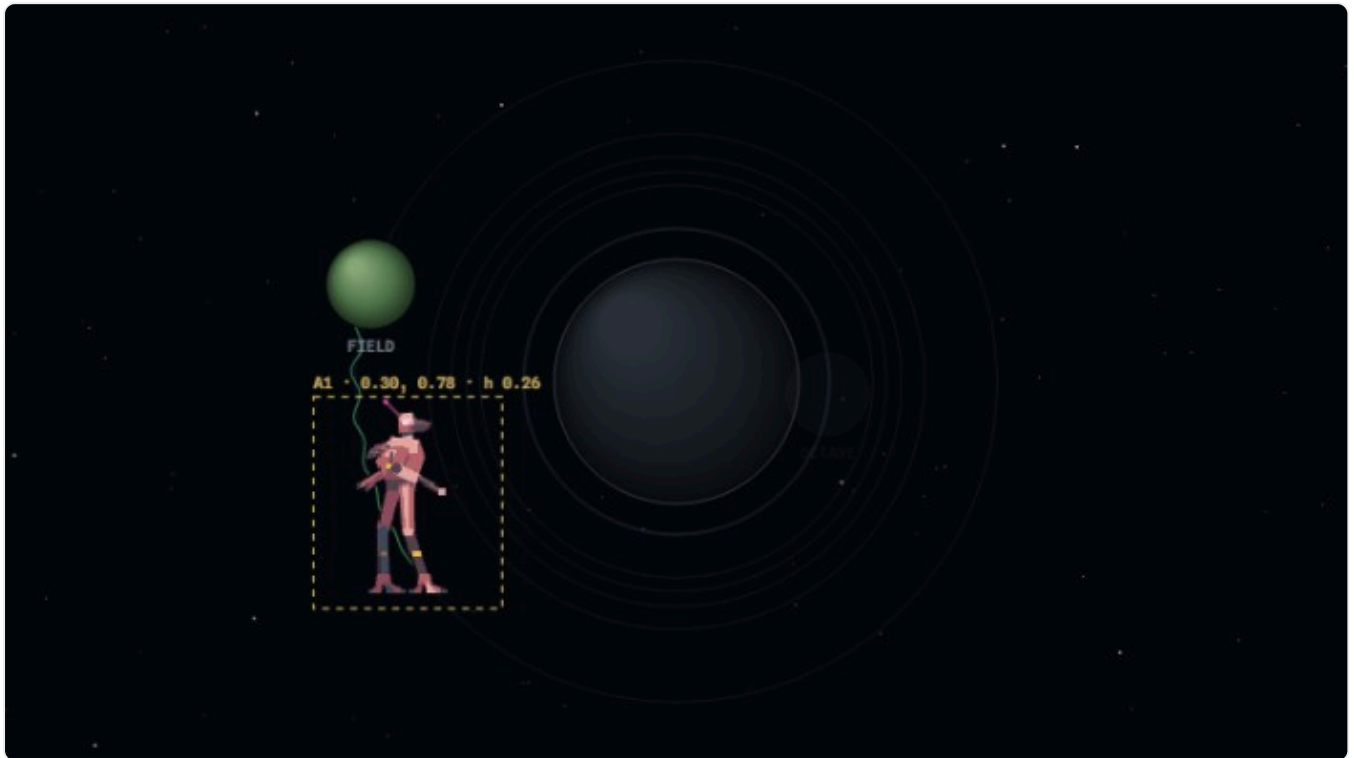
The recorded next-slice plan is explicit about the limit. The current bay proves composition, not a durable multi-beat draft, transition timing, audio/text rhythm or whole-scene playback. Its proposed progression is shot compositor → multi-beat editor → preview through the production runner.

That expansion is **parked, not completed**. The architectural lesson is still useful: one painter, one vocabulary, one runner. A future editor should author data for the existing player, not grow a second cutscene engine that behaves differently.

Basis: H3 and the sceneBay implementation. The screenshot proves the route is reachable, not a full interactive validation of staging/export.

Compose with the pixels that will ship

The public Scene Factory, shown at larger scale: an Orrery backdrop and one scout placed for this illustration. This is a temporary example from the bench, not an authored campaign scene or a production approval.

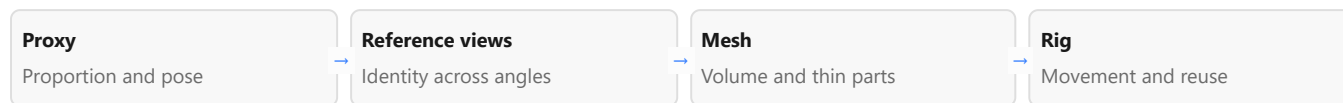


Stage detail from BAY-09 · the same painter and 768 × 432 frame used by the game. The bench's export panel carries the selected plate and actor coordinates into a StageBeat snippet.

Three things to notice: the asset is inspected in the destination frame; its placement is data rather than a prose guess; no campaign save is mutated by the bench.

An image became a sequence of better questions

The rider needed to remain recognisable across views, gestures and media. The answer was not one perfect generation. It was a chain of representations, each reducing a different uncertainty.

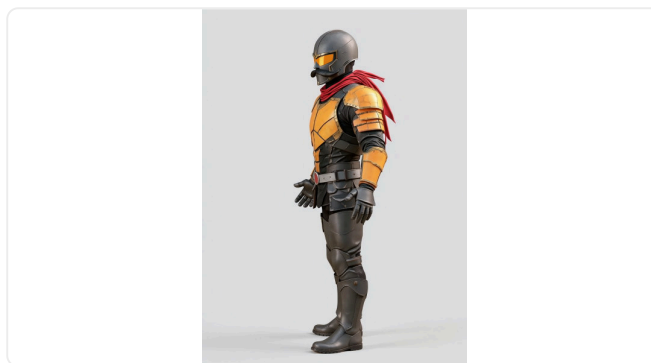


The failure changed the input, not just the wording

The October 4 reference-sheet notes record Firefly reading “mustache + armour” as a bare-faced medieval samurai. More wording did not protect the masked identity. A suited-hero frame anchored the front; that chosen front then guided the side and back.



The early detailed front reference carries identity and surface detail. It is not the final palette or proof of rig quality.



A real profile makes the multi-view input more useful than a flattering three-quarter image.

A strict angle instruction helped create a true profile. Using the proxy-side image as the sole reference fixed the view but copied its blocky style. **Different references constrain different things:** geometry, identity and finish are not interchangeable jobs.

Inspect before promoting

The recorded front/side/back route produced a recognisable Tripo mesh. The next questions were different: did the mustache tips survive, was the scarf consistent, and could the mesh be rigged? The run card's rule was mesh first, rig after. Rigging a flawed mesh only makes the flaw more expensive.

Basis: S1–S3. Original early reference artifacts, resized. Vendor settings in the historical notes are not current instructions or pricing claims.

A generated model is material, not a finished asset

A mesh describes a shape. A rig supplies movement. A production asset also needs scale, conventions, tests and a receiving tool. Treating all of that as one “AI model” hides where a failure actually lives.

Test the simplest useful movement first

The recorded move order was namaste-to-point, idle, then the airborne kick. The first gesture exercised the fingers without root travel. The kick came last because hips, rigid plates and motion through space added more unknowns. A held kick pose could expose weight problems before a whole clip was rendered.

The fingers worked in the auto-rig test; the scarf needed dedicated bone chains. An early GLB re-export was faulty, while the FBX in the original bundle was clean. The moves script used that source. “It exported” was not enough—the receiving tool had to reveal whether the skeleton and orientation survived.

Reusable archetypes emerged from actual characters

The later rig process distinguishes humanoids, a beast with jaw and tail, an armoured arthropod with claws, an odd two-faced humanoid, vehicles and props. They share some posing machinery, but not the same repair recipe.

THEL's claw hinges needed measured axes and both halves moving; a one-sided motion looked like a lever. MAYASURA's unusual limb orientation could be intentional, so it had to be tested before being “fixed.” The bike did not need the humanoid skeleton an auto-rigger put inside it.

A print asks different questions from an animation

The early print mesh had open edges, and its mustache became too thin at miniature scale. The recorded repair thickened surfaces and remeshed the output into a millimetre-aware 3MF. That solved physical preparation questions but softened edges. A print derivative can need different compromises from the screen master.

The reusable result is therefore a **master plus recipes**, not one universal file. Keep fixes in scripts, measure scale against the character rules, and derive the output for the destination. A watertight report still does not prove a physical print succeeded.

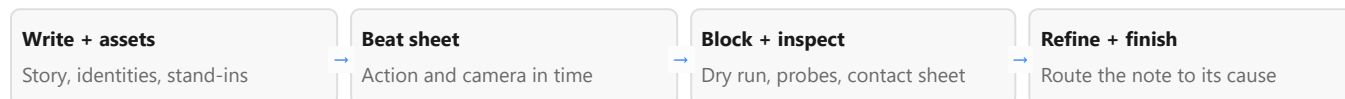
Basis: S3–S4. Reported mesh/rig/print-prep results are historical evidence, not a fresh export or printer test.

The beat sheet makes the scene discussable

SaffronRider's film process grew from actual scene work: write what happens, prepare assets, specify poses/camera/shading, block the scene, take notes, refine, then finish. A note can send the work back to any earlier step—not just trigger another full render.

The intermediate models are a plan, a clock and a blocking pass

A scene plan identifies shots and lengths. A beat sheet says which camera sees which action, with the rider's moves and the opponent's moves named on the same grid. At the recorded 148 BPM and 24 fps, a beat is about 9.73 frames; the audio grid is the timing reference, not a convenient whole-frame guess.



Notes loop upstream. Existing poses are used or adapted before new work is built.

Proofs caught things the script could not explain away

The process notes record early hits landing metres short. Anchor measurements exposed the gap. A camera-visibility check reduced how often another body covered the rider's head. Probe frames and enlarged contact sheets checked whether the shot actually read, not just whether it rendered without errors.

Later polish records make the same distinction: measured planted-foot corrections and sample-identical audio refactoring are mechanical evidence. A short rendered kick proof still needed visual revision because the first trail looked like a laser, not flame. Neither kind of proof replaces the other.

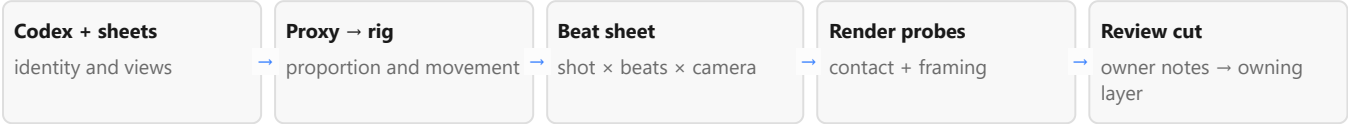
Fix the layer that owns the problem

A sound or callout note can go to the mix or 2D pass; a set note can reuse character layers; a pose or camera note returns to the scene; a mouth or claw problem goes back to the rig. New version names and frame caches preserve comparison. The intermediate layers make this possible without regenerating the whole idea.

Basis: S5, S7. The latest inspected polish record still leaves full-speed watch/listening approval and final-master work open; this guide does not call the film finished.

A character becomes a production by passing through tools

The Saffron Rider workshop spans a public Studio, a model bench, a print bench, a game, a 2D puppet, an animatic and a film pipeline. These are different inspection points, not proof that every output is finished. The reference sheet keeps the covered face, visor, moustache, scarf and armour coherent; a proxy tests proportion and camera; a rig tests reach; a beat sheet tests timing.



The film process routes a note to the first layer that can really fix it.

Which tool answers which question?		
Tool or artifact	What it makes visible	Still needs
Model / print benches	Geometry, pose, material or printable clearances	Final rig/physical inspection for the destination.
Beat sheet and animatic	Action, camera and music over the 148 BPM grid	Moving production frames and a human watch-through.
Probe frames and contact sheet	Whether a hit lands and the hero is visible	Full-rate timing, sound, finish and approval.
Versioned review cut	What changed after each round of notes	A rendered master and explicit owner sign-off.



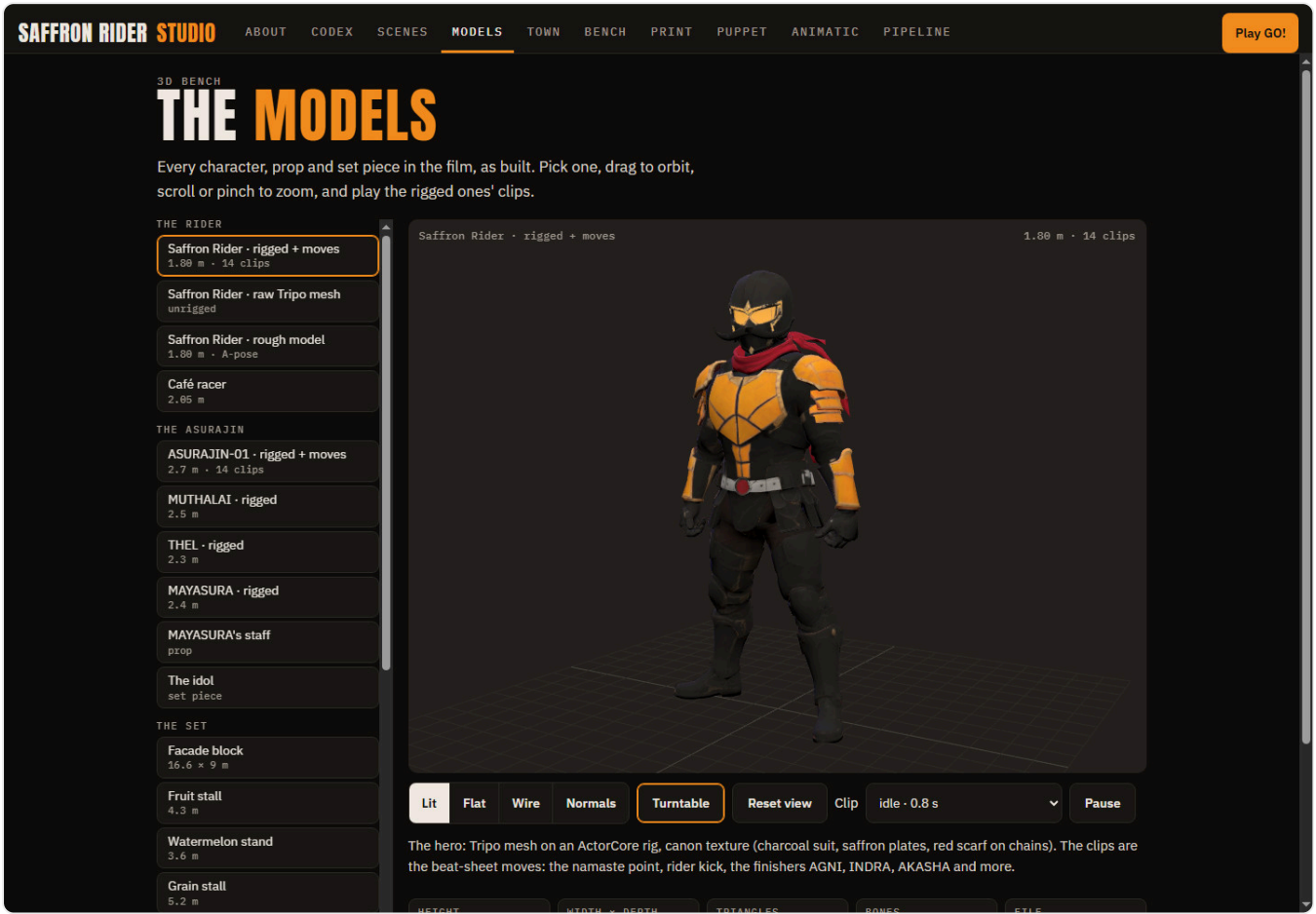
Public Saffron Rider Studio landing screen. The tools are destinations for inspection, not production approval.

For example, if a punch misses, the fix is in animation blocking or measured anchors—not a prettier UI. If the silhouette fails, go back to the design sheet or model. A technical pass should not be retold as a finished-film approval.

SaffronRider docs/PIPELINE.md, docs/film/FILM-PROCESS.md, docs/henshin/TRIPO-REFSHEET-LEARNINGS.md and docs/film/POLISH-REVIEW.md (read from source; the studio screenshot is a visible landing page captured October 13, 2026).

The Studio is the front door to a production workshop

The tabs separate different questions: does a model look right, can a rig move, can a pose read, does a beat fit, and does a scene work? A useful tool does not collapse them into one “approved” badge.



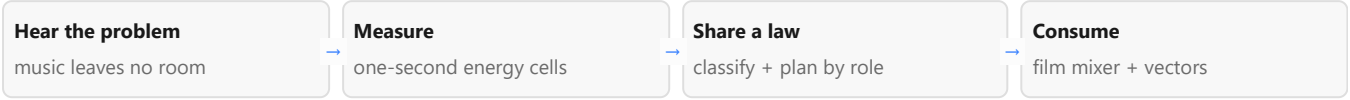
Public SaffronRider Studio Models view, captured 13 October. Published gallery content can lag private production source; inspection is not film completion or rig approval.

A bench is a model of one production problem		
Tool	What it helps inspect	What it does not prove
Model Bench	Silhouette, skeleton, clips and beat-grid timing.	A browser rig editor or final animation approval.
Puppet / animatic	Poses, motion tests, layered SVG and beat-sheet previews.	That the Blender shot or final film works.
Print Bench	Dimensions, open edges, overhangs and stability estimates.	A physical print guarantee.

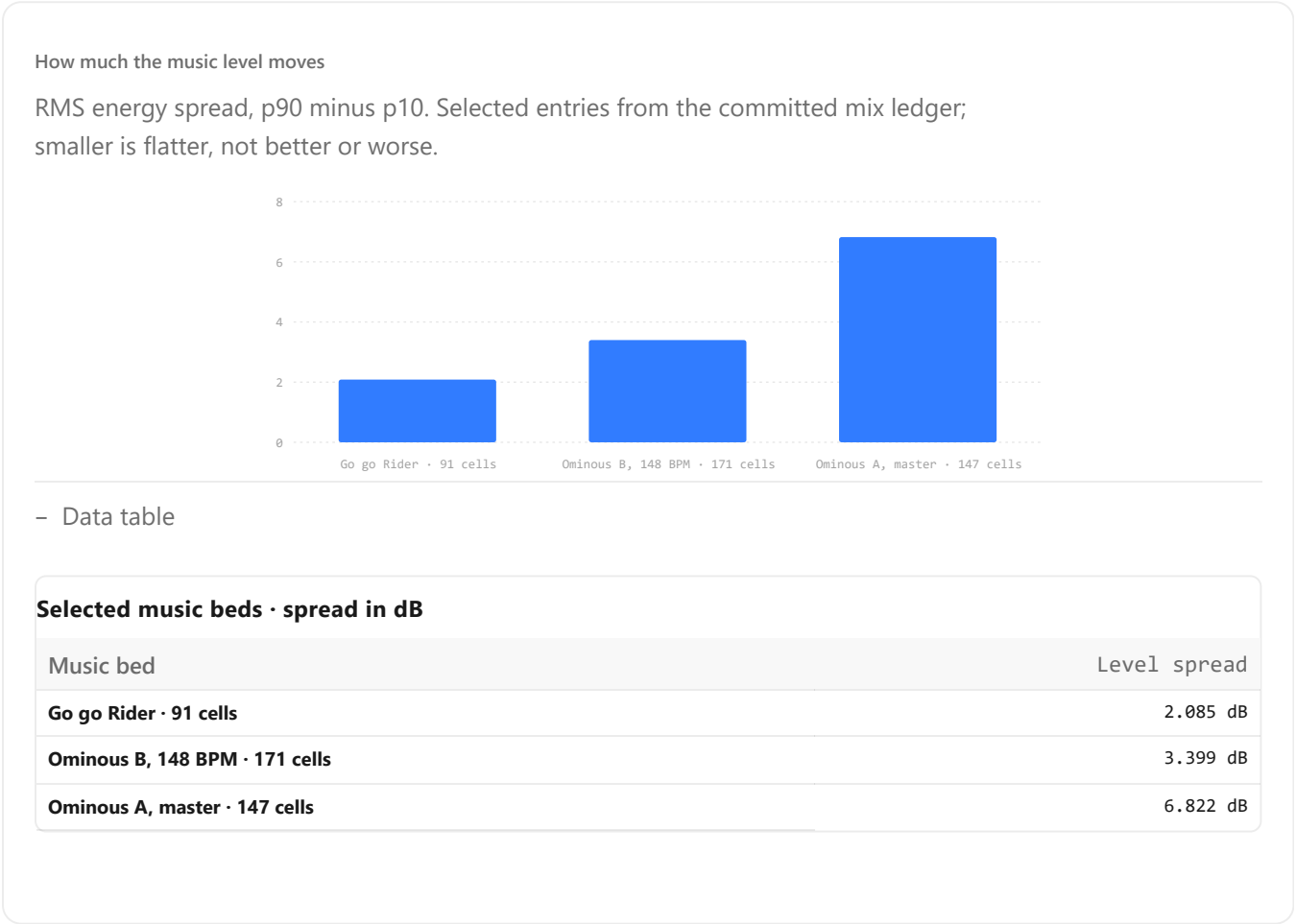
Source: SaffronRider 0fe400d, Studio and tools. Episode 1 remains in blocking; selected published cuts may lag production source.

The film mix sent a new question back to Chirp

A film scene needs a voice to read over music. A mastered song may stay nearly the same level from verse to chorus, leaving little space for dialogue. SaffronRider measured that behavior, then Chirp’s new **dynamics** family turned it into a repeatable mix plan. The film mixer now consumes a pinned generated bundle and the plan’s numbers—not a second hand-written version of the law.



This is a documented feedback loop from a real production need, not speculative generalization.



Each cell is one second of mono audio at 48 kHz; silence below –45 dB is omitted. The windows and durations differ, so this is a diagnostic comparison, not a quality ranking. The ledger stores the source hash, statistics, classification and plans for two roles: music under dialogue versus intro/outro theme.

The contract: the same measured bed can get a different plan depending on its role. A flat score bed may need trim, a dialogue-frequency hole and deeper ducking. The source law and anchored vectors make those choices inspectable; the final scene still needs a human listen.

Chirp av-core/README.md at 9f02c7d (dynamics, October 12, 2026); SaffronRider assets/music/dynamics.json at fb00516. Selected paths: assets/music/go-go-rider.mp3, assets/music/ominous-battle-b-148.mp3 and assets/masters/music/ominous-battle-a.mp3. Ledger values, not new measurements.

The useful thing may be underneath the product

Reuse became more concrete when there was a second real customer for a component—not just a hope that a generic framework might be useful someday. Two other projects in Bestiary's knowledge collection show this from different sides.

Pixelscapes: a scene as a recipe, not a saved picture

The recorded landscape engine describes a scene through an archetype, variant and seed, painted through a limited palette. Hand-built landscapes and data-driven recipes share a renderer. The intermediate model is the recipe: enough instructions to reconstruct the scene, not an opaque generated image.

A frozen prototype supplies a reference for comparison. The records distinguish a scene's identity from its presentation, so adding a new palette need not silently change an old shared scene. Changes that tests cannot represent, such as intentional re-tints, go into a decision record.

Heron later consumed a thin landscape core rather than the whole donor app. Provenance, parity checks and a repeatable update path made the transfer inspectable. The engine was reusable; the calendar and the landscape explorer remained different products.

Phrasebloom: simulation as a model of play

The music-match puzzle uses simulations to explore board rules and player behaviour. Skilled, casual and idle bots are different test models, not a substitute for all players. An idle bot is especially useful: it asks whether the system rewards doing nothing.

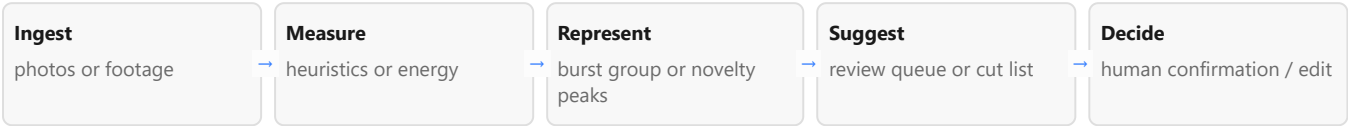
The recorded work committed decision rules before examining the tables. A harder board floor failed a casual-player condition, so scope changed instead of quietly adding help to make the number green. Simulation offered a reason to choose, not proof that humans would enjoy the result.

The harvested record still leaves human playtests and blind listening open. That boundary is the learning: **an experimental model can disprove a mechanism without proving a market, an experience or a finished product.**

Basis: B2–B4, harvested October 3–4, 2026. These examples use dated Bestiary evidence, not a fresh full audit of those products.

Photo and video have different models—but the decision stays visible

Photo Sharingan is the Photo Observatory's active home: it prepares Lightroom/folder images for culling, groups bursts and scores sharpness/exposure/similarity, offers optional AI suggestions while keeping human verdicts separate, then asks the photographer to confirm. **Video Sharingan** grew from drone cut-lists into motion-based shot analysis and reel/timeline outputs. Its video representation is an energy envelope and novelty curve adapted from Chirp's PERCH music detector.



Comparable design pattern, not identical software. Video can also render a reel; a person still finishes it.

One method, two different contracts		
Decision	Photo Sharingan	Video Sharingan
What is grouped?	Frames in a burst, including bracket/pan protection	Time segments from novelty and duration rules.
What must not happen?	Heuristics never commit a reject by themselves	Do not label duration splits as detected events.
Intermediate output	Three queues, scores, human verdict and export audit	Cutlist, Premiere XML, markers or a rendered review reel.
Tooling guard	Read-only until explicit export; no secret in browser	FFmpeg doctor checks required filters before analysis.
Final decision	Photographer confirms pick / reject / unflag	Editor accepts the selects and the final rhythm.

Reuse is a translation: a music detector’s envelope → novelty → boundary idea transfers to footage, but audio sections are not video shots. The selected video profile and review expectations remain native to the receiving tool. Photo Sharingan is not the old “Photo Observatory” placeholder repo; Warble and ROOKERY remain in Chirp.

Current boundary: uniform footage can still be split by duration. A novelty peak is detected change evidence, not the only possible edit boundary. Photo’s persistent score cache is not decision-session restoration.

Photo Sharingan README and docs/HOW-IT-WORKS.md at d976d17; Video Sharingan README and docs/editing-theory at 4a44611; Chirp PERCH. Outcomes are source-described; neither photo cull nor video edit was rerun for this document.

The model starts with an authored world, not an unlimited prompt

Vantara: After the Sunfall is a browser RPG prototype set in a ruined magitech world. Four hero roles take cases from a city hub. The design promise is consistency: AI may help interpret or invent, but should not quietly rewrite the world or its rules.



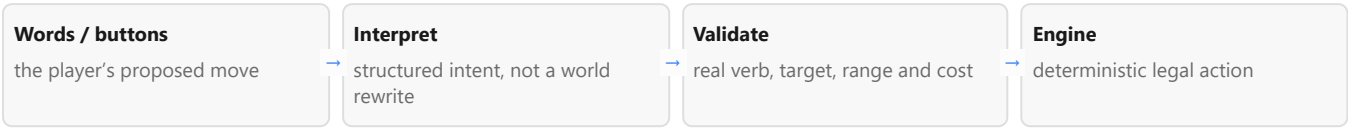
Local Vantara design-kit prototype at 2c0c894. These are cosmetic hero cards, not proof of the AI quest service or runtime game mechanics.

What the prototype contains	
Present in runtime source	Not yet a connected product loop
Hero selection, hub, shop and retirement archive	Generated quests with repair and post-case summarization.
Two authored demo cases; grid and dialogue engines	Consequential personal lore feeding the next case.
Seeded maps, legal actions and intent validation	Eight world-bible outlines are not eight playable quests.

Source: Vantara 2c0c894, docs/PRD.md, src/quests/demoQuests.ts and appStore.ts. Figure is the separate design kit. No public app URL or model service was verified.

Words can propose an action; they cannot change the rules

A natural-language box feels open-ended. The engine is deliberately not. Typed intent is interpreted, then checked against actions legal *now*. A false target, out-of-range move or impossible verb ends at the gate; low-confidence interpretation needs confirmation.



Implemented intent-gate structure; model availability and a successful service call were not tested here.

A tiny excerpt from an actual authored demo case

```
questId: 'demo_ghat_job',
archetype: 'spatial',
title: "The Fence's Ledger",
objective: { type: 'retrieve', target: 'relic_crate',
  successCriteria: { carriedToExit: true } }
```

The designed generation airlock	
Checks already have modules	Connections still planned
Schema and referenced IDs	Generator / repair / authored fallback selection.
Hard canon, personal canon and knowledge	Post-case summarizer and meaningful lore updates.
Playability constraints	Generation → play → memory → next quest.

The reusable idea is the boundary: a creative generator can propose within a contract. A validator module alone does not establish a working end-to-end AI game.

Source: Vantara 2c0c894, src/intent/validate.ts, src/generation/validator/index.ts and docs/ARCHITECTURE.md. Generation airlock is a design, not a shipped loop.

Vibe coding needs a memory and a boundary

Chirp's approach calls this AI-assisted "vibe coding" held to a steward's discipline. The useful part of the vibe is fast exploration: describing a feeling, trying a control, generating a first draft and seeing what happens. The discipline is what lets that speed accumulate into something coherent.

Give AI the right working context

Start with the project's current purpose, locked design rules, the relevant files, the exact host or engine, and the one question this slice should answer. Read the existing work before making more. A whole knowledge library is usually less useful than the few authoritative records that govern today's decision.

Then separate exploration from execution. First ask for alternatives, assumptions and the cheapest tests. Choose the direction yourself. Only then ask for implementation inside a named scope. AI should not turn every attractive side idea into another system.

A starter prompt you can adapt

Goal: [one person, moment and desired result]. Read first: [current project record and relevant design/contract docs]. Preserve: [identity, behaviour, format and production path]. This slice: [one observable outcome]. Not this slice: [explicit exclusions]. Before editing: show two plausible approaches, risks and your recommendation. After I choose: make the smallest useful change. Use existing components. Prove it in [real engine/tool] with [pass condition]. Report what you checked, what you did not, and the next uncertainty. Do not publish or change access settings.

Close the session, not just the chat

Review the actual diff or asset, run the relevant checks, open the result where it will be used, and make the decision visible. Update the project record and write why the choice was made. A chat answer saying "done" is not a durable handoff.

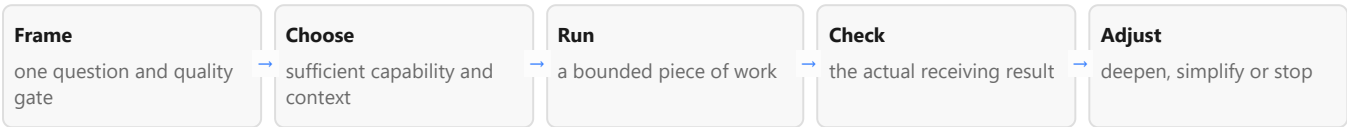
Chirp's "The system thinks, you decide" captures the division: AI can propose, generate, compare and explain; I own direction and approval. I also need to resolve rights, licensing and publication boundaries. A generated asset does not automatically settle any of those questions.

Basis: C1–C2, H4, B1. The prompt is an editorial template, not a transcript or a promise that every repo enforces this workflow.

Spend reasoning where it changes the result

With Devin, I am now learning to adjust model usage as the task changes. Capability, context and compute budget become part of designing the work, not just a bill I notice afterwards.

By **tokenomics**, I mean the practical economics of tokens and agent effort, not crypto tokens. This is judgment-led adjustment, not a claim that I have built an automatic model router.



Keep the acceptance standard fixed while changing the reasoning budget.

Match the effort to the uncertainty	
Work	A cost-aware move
Routine, well-bounded edit	Use a suitable lighter setting and focused context; deepen only if ambiguity or failures justify it.
Design synthesis or a difficult diagnosis	Spend more reasoning on the consequential uncertainty; do not replace judgment with repeated cheap guesses.
Independent cross-repo questions	Use focused parallel lanes when they earn their added cost; parallelism saves elapsed time, not automatically money.
Mechanical checks and repeated steps	Prefer scripts, fixtures and precise handoffs to asking a model to rediscover the same facts.

Token price and the bill are different layers

An API can price uncached input, cached input and output differently, with extra cache-write, tool or media charges. A short prompt can still trigger substantial work. Devin usage depends on the product and plan: Desktop quotas, enterprise ACUs and cloud-agent resources are not one universal per-prompt tariff. Use current rates and actual usage, not an old model-price screenshot.

The useful measure is cost per accepted, verified result. Include retries and child work; record human review time alongside the bill. A lower-priced attempt can cost more overall if it repeatedly fails. Less context can also remove a crucial constraint, so saving tokens must not weaken the quality gate.

The learning is real; the savings are not yet measured. I have not supplied a per-model cost ledger or a like-for-like comparison. The internal edition proposes one small record to learn from the next task.

Owner-reported learning, 13 October 2026. Billing boundaries checked against [Devin Desktop usage](#), [Devin billing](#) and provider API pricing. No personal plan, rates or savings inferred.

Ask for evidence at the right level

“The tests pass” can mean the code parses, one state changed, a canvas painted, a file exported or a real user completed a task. Those are not equivalent claims. The workshop’s strongest checks make the boundary visible.

Four different kinds of evidence

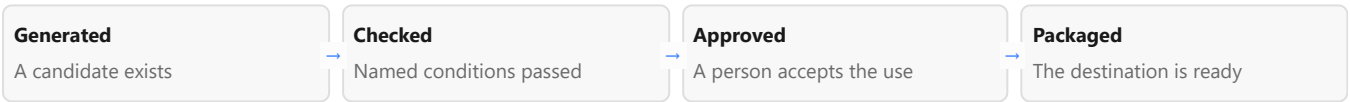
- 1. **Mechanical:** known inputs produce expected outputs. Test the actual module or renderer, not a copy of its logic in the test.
- 2. **Integrated:** the host consumes the artifact correctly. Open the exported rig, use the game painter, or check the deployed route.
- 3. **Perceptual:** a person can read, hear and use the result. Look at the screen, listen on the intended device, watch the shot at speed.
- 4. **Decision:** the owner explicitly accepts the result for its purpose. Technical evidence supports that choice; it does not make it automatically.

Prove that the checker can say no

Bestiary calls this earning a check by mutation: deliberately introduce the failure the check claims to detect and confirm that it fails. A test that only looks for a word can pass commented-out code. A test for “file exists” can accept a silent stem or a broken export.

Chirp’s de-spaghetti work found another version of the problem: an inventory tool reported duplicated capability because it was counting generated library code as hand-owned duplication. The checking tool itself needed correction. Evidence should be checked against reality, not believed because it has a reassuring label.

Keep the promotion path honest



Stages can overlap, but none should be inferred merely from the previous label. A destination check can still send the work back.

Basis: B1, C2–C6, H2, S7. Technical checks and human creative approval remain deliberately separate.

Leave the next cut ready for someone who was not in the room

A cold handoff lets a new person or AI session continue without the previous conversation. It carries the image of the whole, the reason for the last cut and the next useful slice. "Done" is not enough; a pile of transcripts is not a substitute for current state.

The small packet a fresh session actually needs	
Carry forward	Make explicit
The intended whole	Purpose, receiving user, identity constraints and the roadmap; distinguish chosen direction from parked possibilities.
The actual position	Repo, branch/commit, unmerged work and local-only files/assets. Do not assume main or another machine has them.
The last slice	Exact artifact/version and paths; generated, checked, approved and untested are different states.
The proof and limits	Setup and reproducible checks; known failures, permissions and what this agent cannot reach. Never copy credentials.
The decisions	Why this cut was chosen, what must stay true and which older instructions are superseded.
The next cut	One bounded outcome, explicit non-goals and its acceptance check; a paste-ready pickup, not a fresh mega plan.

Test the pickup, not the length of the handoff

Give a fresh session the canonical status and linked artifacts, without the old chat. Can it locate the right state, repeat a relevant check and restate the next slice before editing? Missing context is a handoff gap to repair, not an invitation to guess. This is a proposed check, not a drill completed across the fleet.

Turn the last cut into a reusable lesson

"The mesh broke" is an anecdote. The useful learning names the failed export, the clean source, the receiving-tool check and the boundary of the claim. Janitor incidents become cause/fix/prevention records. Keep one owner status surface; archive older decisions with dated forward pointers instead of making several competing "latest" files.

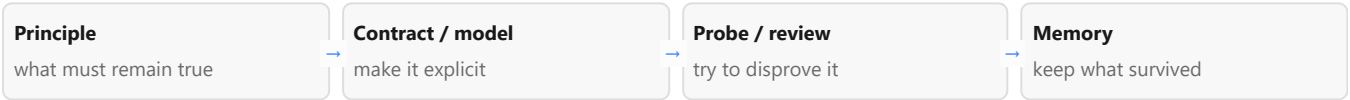
Reuse still needs translation: retain provenance and test the adapter in its receiving tool. An old branch is a proposal against current work, not an automatic restoration. Bestiary holds reusable guidance, Flyway records intentions/handoffs, and Watchtower presents selected truth; their scaffold is not proof of full-fleet adoption.

Basis: inherited C1-C2, H1-H4, S4-S5, B1; Bestiary ways-of-working P41/P43/P44/P48/P49/P54; owner sculptor framing, 13 October 2026. Editorial practice and proposed pickup check, not a new enforced policy.

A principle becomes useful when it changes the next decision

These are working principles distilled from the projects—not new canon adopted across the fleet. Each one should point to a model, a boundary or a receiving test. Otherwise it is just a slogan.

Principle → practical move → concrete example		
Principle	How I use it	Example
Purpose before surface	Picture the intended experience; sketch an adjustable roadmap.	A Chirp instrument starts with its sound path, not a generic dashboard.
Separate fact from inference	Give measurements, guesses and suggestions different jobs.	A Photo Sharingan score is not the photographer's verdict.
Rules before improvisation	Make the stable contract small and inspectable.	Vantara's words pass a legal-action gate before the engine acts.
Chisel one useful slice	Close a small end-to-end result that answers the next doubt.	A proxy, rig/pose and shot each answer a different question.
Use the real receiving surface	Test where the result must actually work.	Scene Factory uses the game's stage painter; an XML file still needs an editor import.
Make the handoff cold-ready	Leave state, decisions, evidence and one next slice for a fresh session.	The repo carries the context; the previous chat is not required.



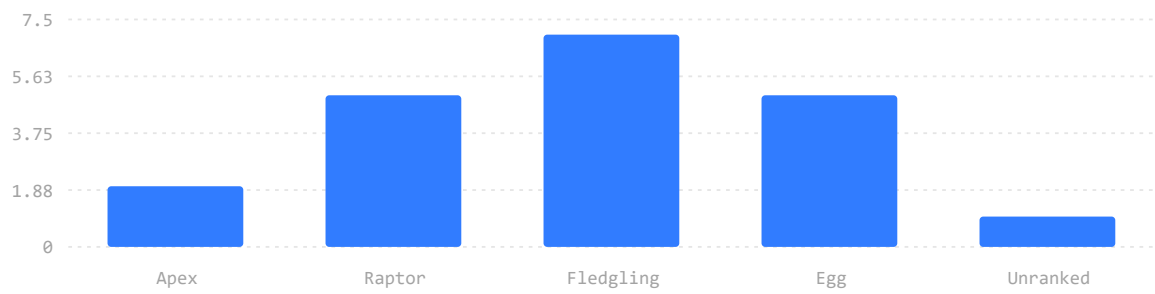
The structure is reusable; the project's sound, story and identity remain its own.

AI helps make alternatives and artifacts faster. Design judgment decides which question they answer. The Janitor, learnings logs and source-pinned handoffs help that judgment survive the next session.

Editorial synthesis from Bestiary's standards and documented Chirp/HALLIATUS/SaffronRider/Vantara/Sharingan patterns. Not a new fleet policy.

Apex and emerging describe different amounts of carried context

The rank ladder is about what the next agent inherits: a mission, a gate, learnings, pickup context and reusable exports. It is not a score for the art, a market valuation or the owner’s creative priority.



– Data table

Recorded labels, not a new rank audit

Rank	Repos
Apex	2
Raptor	5
Fledgling	7
Egg	5
Unranked	1

Registry snapshot, 13 October 2026: 20 repos after the owner’s Apollo/school-project exclusions. Bestiary is the hub, not one of these registry entries.

Read the ladder without over-reading it

Recorded label	Plain meaning
Egg	The next agent still needs basic mission/status context.
Fledgling	The mission and current state can travel to the next session.
Raptor	The documented ladder adds gates, learnings and deploy evidence.
Apex	The ladder adds pickup continuity and reusable exports.

A useful counterexample: SaffronRider is an Apex *production focus* in the owner’s language, while its registry label remains Fledgling. Video Sharingan’s Egg label does not erase its working tooling. The opportunity analysis belongs in the internal edition, not inside this engineering chart.

Recorded flyway/fleet.yaml at Bestiary 8736085 and gold-standard.md. Historical audit labels; criteria were not re-probed for this guide. 2 Apex, 17 emerging (Raptor/Fledgling/Egg), 1 unranked.

What each major repo is for—and what it contains

A quick map for someone entering the workshop. Feature presence is source evidence, not a fresh acceptance test. The chapters explain the models and the limits behind these summaries.

Making and playing		
Repo / surface	Product description	Main contents / boundary
Chirp / AVIARY	Browser instruments and reactive worlds. Gyre, Soar, MOLT, bird synths and listening surfaces explore different ways to play.	Audio graphs, plates/trips, PERCH analysis and a shared av-core. Not every page uses the same core.
HALLIATUS / TALON	Music-judged mech tactics plus cockpit stations, a campaign and a creative workshop.	Songs/stems, campaign bible, sprite/icon/interface factories, Tool Shed and single-shot Scene Factory.
SaffronRider	An authored Tamil-flavoured tokusatsu world, film drafts and a game prototype.	Studio, character/asset intake, model/print/pose benches, beat sheets, Blender shots, post and assembly.
Vantara Chronicles	A browser RPG prototype with an authored magitech world and four hero roles.	Two demo cases, grid/dialogue engines, legal intent gate and world schemas. Generated quest/memory loop is planned.

Reuse map: Chirp contributes selected audio laws and music-analysis ideas; HALLIATUS contributes production-contract/factory patterns; SaffronRider carries authored creative identity. Vantara demonstrates a constrained-intent architecture, not a ready-made AI quest backend.

Current source pins in the edition’s source notes. Component reuse and design-pattern reuse are intentionally distinguished.

The other half of the workshop selects, checks and connects

Selection and connective tissue		
Repo / surface	Product description	Main contents / boundary
Photo Sharingan	A local-first cockpit for reviewing a photo shoot.	Folder/catalog/album intake, worker scoring, bursts, optional AI help, human verdicts and confirmed exports. No decision-resume UI established.
Video Sharingan	Local video analysis and editable cut plans, not a browser NLE.	Motion/novelty models, shot selection and sequencing, optional music alignment, XML/render plans and directed-reel scripts.
Bestiary	The reusable memory and working-system hub.	Canon/standards, source harvests/cards/themes, fleet registry, Flyway scaffolding, doctor and these guides. Recording a method is not fleet adoption.
Watchtower	The owner’s monitoring and reading site.	Fleet status, Bestiary-backed Focus/field-guide views and a protected workshop reader/PDF route. This pack does not switch that route.

Other creatures, without another roadmap

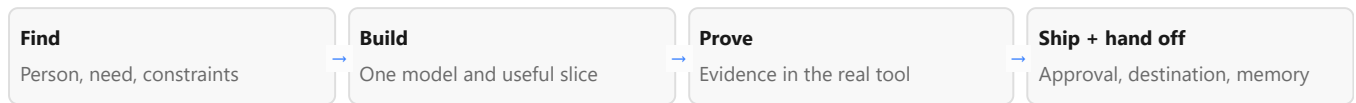
Heron is an agenda-first calendar PWA using a seeded Pixelscapes scene; **Pixelscapes** is the donor visual engine. **Phrasebloom** explores a playable-album/game premise. **Lodestar** is a personal operating system with one CSV source and agent handoffs—not a public entertainment product. These descriptions preserve the October 3 harvest, not fresh feature audits.

The full registered fleet remains in Bestiary. Apollo and the school analytics repo are excluded from this edition’s scope; Warble and Rookery are Chirp components, and Photo Observatory is now covered by Photo Sharingan.

Primary eight repo specs above use the research pins. Wider descriptions: Bestiary knowledge/repos summaries dated 3 October. No personal Lodestar records opened or reproduced.

Make one useful thing, and leave it understandable

Use this as a worksheet, not as seventeen new obligations. A small sound tool, a single scene, a character pose or one screen is enough. The point is to move one idea through a complete learning loop.



If the evidence changes the problem, return to Find. This is a loop, not a waterfall.

1 · Before you ask AI to build

- **Moment:** who is doing what, and what is difficult?
- **Known / assumed / unknown:** one line each.
- **Must stay true:** identity, behaviour, rights and format.
- **NOT-list:** what this slice will deliberately not do.

2 · Choose the cheapest honest model

- **Options:** two or three genuinely different routes.
- **First question:** the uncertainty that blocks progress.
- **Model:** graph, clickable study, proxy, reference view, beat sheet or fixture—and what it cannot prove.
- **Pass line:** a visible, audible or measurable condition chosen before the trial.

3 · Build, inspect and decide

- **AI boundary:** relevant source, approved direction, allowed change and excluded scope.
- **Reality check:** actual engine, receiving tool or device, plus one bad/null case.
- **Human judgment:** what you saw, heard or tried, and whether it is good enough for this purpose.
- **If it fails:** change the owning layer; do not regenerate the whole project by default.

4 · Close the loop

- **Deliver:** a result someone can open or use, checked at its destination.
- **Record:** one decision, one learning and the next uncertainty.
- **Keep:** the source/master, reproduction path and useful evidence.
- **Reuse later:** extract the proven mechanism when a second actual need appears.

The measure of progress: not how much AI produced, but how much uncertainty you removed—and whether the next person can continue without guessing.

Start with a tool, then look at how it was made

The reader links below lead to live Cloudflare Pages surfaces, not private repositories. Some are working prototypes or development tools. Original access checks dated October 12, 2026; selected pictured routes rechecked October 13; a reachable page is not a complete product test.

[Chirp / AVIARY](#)

Public · start with a browser instrument.

[Chirp · Summary of Approach](#)

Public · the early build story and design philosophy.

[HALLIATUS / TALON](#)

Public · the game and its interactive world.

[HALLIATUS · Tool Shed and design labs](#)

Password-protected · Status, Docs, Bays and foundries.

[HALLIATUS · BAY-09 Scene Factory](#)

Public route · stage a shot using the game painter.

[SaffronRider](#)

Public · the world and playable rhythm-fighting prototype.

[SaffronRider · Studio](#)

Public · production and inspection tools.

[Watchtower](#)

Password-protected · a reading surface for the fleet.

Where this guide's examples come from

C1–C6 · Chirp: Approach, design principles, de-spaghetti, audio/visualizer and single-file learnings, av-core and verification notes. **H1–H4 · HALLIATUS:** overview/PMO, factories, Scene Factory plan and implementation, decision-page learnings. **S1–S7 · SaffronRider:** pipeline, proxy, first reference-sheet run, rigs, film process, 2D trial and polish review. **B1–B4 · Bestiary:** ways of working; dated Pixelscapes, Phrasebloom and Heron harvests.

Design-thinking framing also draws on the [Design Council Double Diamond](#) and [Stanford d.school Bootleg](#). These clarify the method; they are not claimed as the origin of my projects.

Prepared October 12, 2026. Revision-pinned source paths, caveats and regeneration instructions are maintained beside this guide in Bestiary. Read without a network; outbound links need access. No protected credentials are included.

Edition 13 October 2026. Source evidence is dated; screenshots illustrate surfaces, not comprehensive product tests. See the edition pack's source notes.