RESCALE MEDIA · FOR TEAM REVIEW

Skynet — the unified platform

Status Approved by Mathew (recorded 2026-08-25, ADR-010)

Date 2026-08-18 · addendum 2026-08-25 (Reddit VOC)

Audience Leadership + creative, media-buying, research, dev

No code knowledge needed

---

1 What Skynet is

Skynet is one app designed to run our whole creative operation as a single loop:

> **Evidence in → team decides → scripts → creatives (images + video) → launch with a tracking code → performance back → the app learns → better next batch.**

Today this loop exists, but it is split across four tools and ClickUp, with manual steps between them. Files are downloaded by hand, uploaded by hand, and renamed by hand. The feedback loop — which creative won, and why — is mostly manual and often breaks. Skynet closes the loop in one place.

The speed from insight to execution is the difference between fast- and slow-scaling brands. That was the founding idea of the Skynet program. This app industrializes both sides: the looking (intelligence) and the acting (creative production) — with human judgment kept exactly where the team wants it.

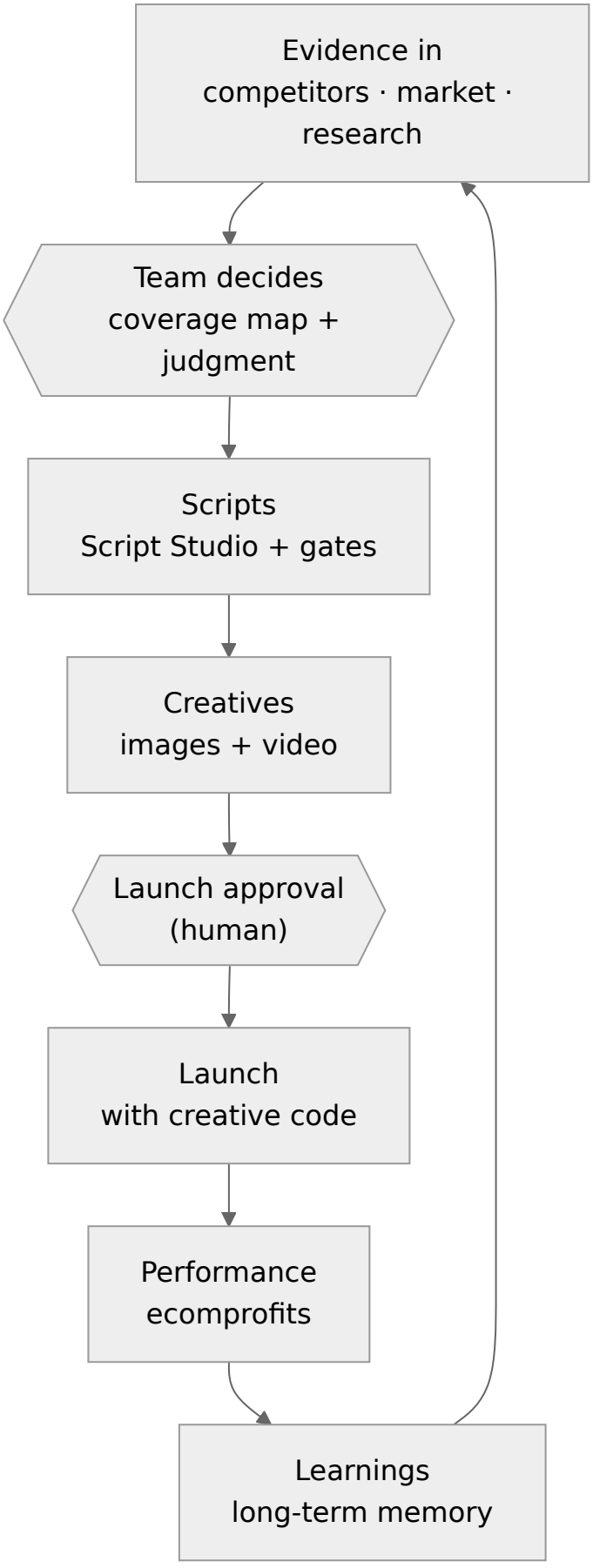

Fig. 1 — The loop. Diamonds are human decision points; everything between them can run automatically.

2 What we already built, and where it goes

Nothing is thrown away. The unified app is assembled from systems that are already live and proven:

TODAY	WHAT IT DOES	IN THE UNIFIED APP
Meta Ads Scraper / Intelligence Engine	Scrapes competitor Meta Ad Libraries at scale, labels every ad (angle, format, awareness, structure), intel dashboards, product research, on-demand deep teardowns — and collects customer voice (Reddit discussions, Trustpilot reviews, Amazon reviews) into one market-intel corpus	The **Intelligence** module. Same engine, shared data spine, one frontend — one login
Static Image Ads	Generates statics (Explore, Expand, Long-Form), video storyboards with AI clips and narration, winner gallery, SuperMemory	The **Production** module and the initial product base
RescaleOS	Agent-driven script engine with human approval gates, skills, scheduled autonomous runs	The **Script Studio + automation runtime**, upgraded from a single-operator system to the whole team — drawing on competitive intel and the live market-intel corpus
rescale-workos	The service behind ClickUp: creative demand, naming, launch workflow, fulfillment	The **Workspace** module. ClickUp's UI can be replaced step by step after pilot approval; until then the current workflow stays in place — the logic is already ours
ecomprofits	All-channel performance data (Meta, Pinterest, Shopify, Klaviyo, Funnelish, Taboola), naming-convention reporting	Stays exactly where it is — the **system of record for performance**. Skynet reads from it and links every creative to its results

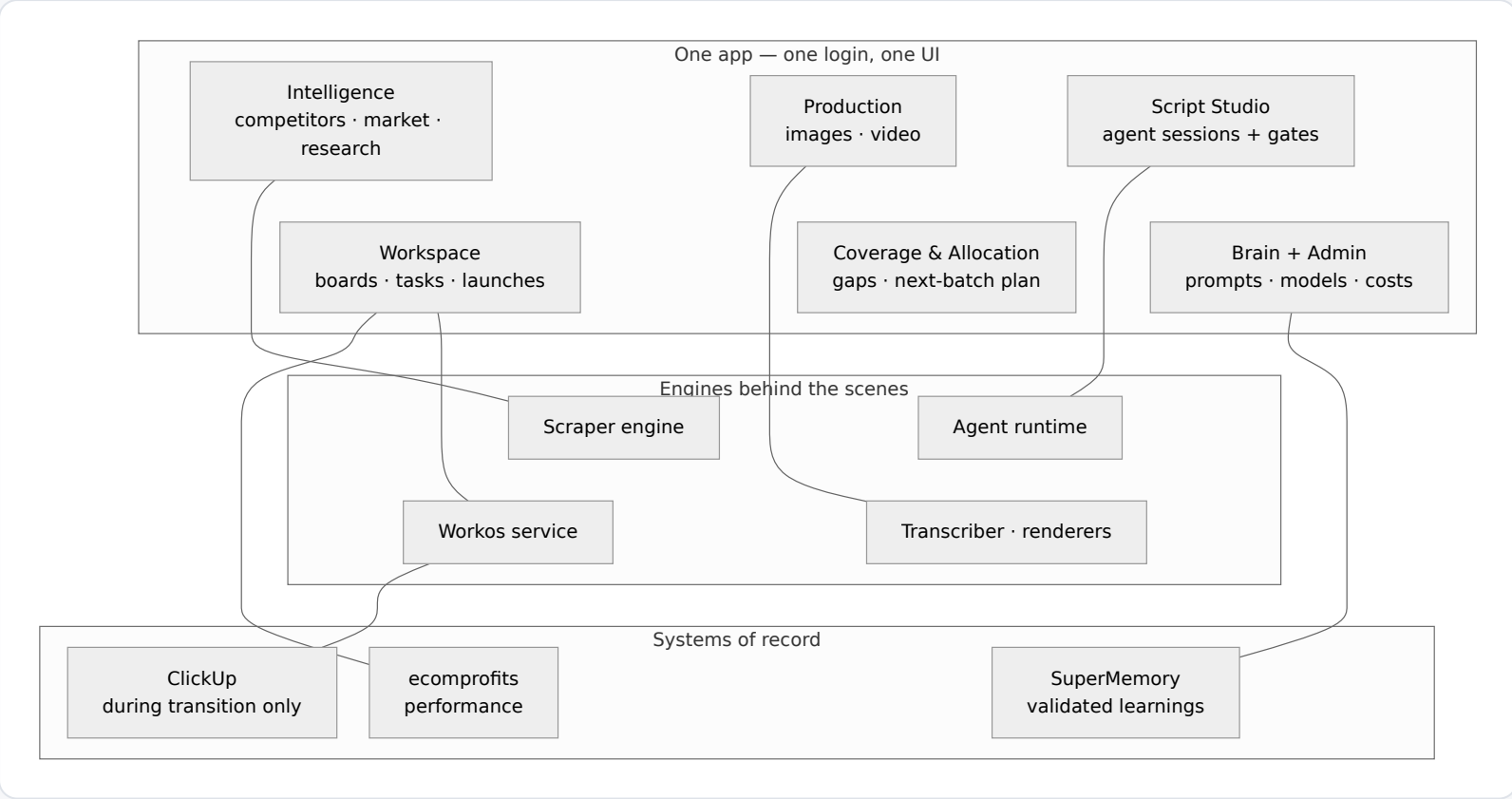

Fig. 2 — Module map. The team sees one app; proven engines keep running behind it; systems of record stay where they are.

3 How it works, step by step

3.0 Foundations (set up once, maintained continuously)

The app has workspaces and roles. A workspace holds:

- **Products** — brand identity, docs, reference looks, offers.
- **Competitors** — tracked once at workspace level (a competitor can matter to several products), linked to products with per-product settings (market, locale, auto-recreate rules). A competitor is a market entity with one or more "lenses": its Meta Ad Library page, its domain, later its Amazon/Trustpilot/Reddit presence.
- **Research keywords** and niches.
- **The taxonomy** — the shared vocabulary for angles, formats, awareness stages, structures. Owned by the team with creative leadership accountable; versioned by the app.
- **Model settings** — which AI model runs which job. Configurable by admins in the app, per job type, without redeploying.
- **Prompts** — layered: company doctrine → workflow prompts → brand/product context → **personal overlays** (each user can tune how the AI works for them without changing the shared base). Editing is guided: built-in authoring assistants help write each layer, and guardrails keep the layers clean — brand facts, visual style, and model rules each live in their own layer — so a non-technical edit is far less likely to break generation.
- **Approval gates** — which steps require a human decision before work continues.
- **Work management** — boards, tasks, and assignments are first-class objects in the app (the ClickUp replacement, phased and gated — see §7).

3.1 Intelligence flows in (continuous, automated)

- **Competitor scraping:** scheduled quick scrapes (ranked top ads) and deep crawls that build a broad, resumable historical catalog for each tracked

competitor lens. New ads are deduplicated into creative families, media is stored, videos are transcribed automatically.

- **Classification:** new creative families are labeled automatically and cheaply with the closed taxonomy. Label quality is measured against a human-labeled gold set. Coverage and confidence are always shown where labels are used.
- **Market intel:** v1 already collects real customer language from Reddit discussions, Trustpilot reviews, and Amazon reviews into one shared corpus. Each source has different access rules, sampling limits, and coverage notes, which are shown in the app. More sources (e.g. YouTube comments) are added source by source. Claims and pains land in the same evidence pool, tagged by source and language. **Reddit voice-of-customer (VOC) pipeline — added 2026-08-25 at Mathew's request after approval, built directly in the unified platform:** continuous Reddit collection for agreed keyword groups (first two: *cravings* and *blood sugar*), accumulating into a master dataset (never one-off), with export for use in briefs and scripts. It reuses the market-intel source adapter and the scoping/keyword model that already exist; it needs no change to the loop or the modules above — it is a configured source plus an export, delivered in the Intelligence module (Phase 1, after the day-14 slice). **Scientific evidence (PubMed) pipeline — added 2026-08-26 at Mathew's request:** the WHY next to the VOC HOW. For each keyword group, six lenses are searched in PubMed — the mechanism, the root cause, the main symptoms, the ingredients, failed solutions, desired outcomes — continuously, with every study classified by evidence strength (meta-analysis / RCT down to preclinical and traditional use). Findings are translated into plain customer language: new mechanism angles, fact-based hooks, authority statements, belief-shifting arguments, symptoms to call out, and reasons other solutions fail — each with its citation and evidence class, so the compliance pass reads strength from the row, never from the prose. It is a second market-intel source adapter plus an MCP the agent runtime and RescaleOS use directly, delivered in the Intelligence module (S2) with the evidence packet attached to briefs next to the VOC packet.
- **Product research:** keyword-seeded crawls cluster ads into products and score them with the research rubric — a continuously updated ranked opportunity list. A promising product is promoted into a tracked product

with competitors and intel pre-linked. Product research is not a separate tool; it is another entrance into the same loop.

- **Traction alerts:** when a competitor's ad family shows momentum (longevity, variant count, impressions where visible), the team is pinged in Discord/Telegram with a link into the app.

3.2 The team looks and decides (human)

Strategists browse competitor libraries with label filters ("what ad types and angles is this competitor leaning on?"), longest-running winners, what's new this week, market claims per niche, and the research ranking. On any ad they can run a **Deep Teardown** — a persuasion-mechanics breakdown (awareness stage, hooks, structure, formula) stored with the ad — or trigger teardowns across a brand's top creative families in one click. Team members tag each other and attach notes: *"competitor of product X — this type is working, let's try a similar script."* Every tag creates an inbox item with the evidence attached. Alerts and quick actions also work from Telegram/Discord, but the app is the canonical place where evidence, notes, and next actions live.

3.3 The coverage map decides what comes next (the strategy layer)

From that evidence, the app and the team decide what the next batch should cover. Creative diversity is managed, not guessed. The app keeps a **coverage map** per product: our messaging mapped over **segments** (who we talk to) × **awareness stages** (what they already know) × **angles** (the argument we make). Our own ads and competitor ads are mapped onto the same grid through their labels and naming data. A **gap finder** ranks the thin spots by importance and reach. From there the team allocates the next batch — with judgment and taste, at a human gate — across three lanes: **net-new** concepts (fill gaps), **variations** (iterate on what works), **scaling** (more of the winners). Brief generation pulls from this queue, so "what should we make next?" has a measured answer instead of a guess. Honesty guard: every cell of the map shows its evidence count and label confidence — the map is never allowed to look smarter than the data under it.

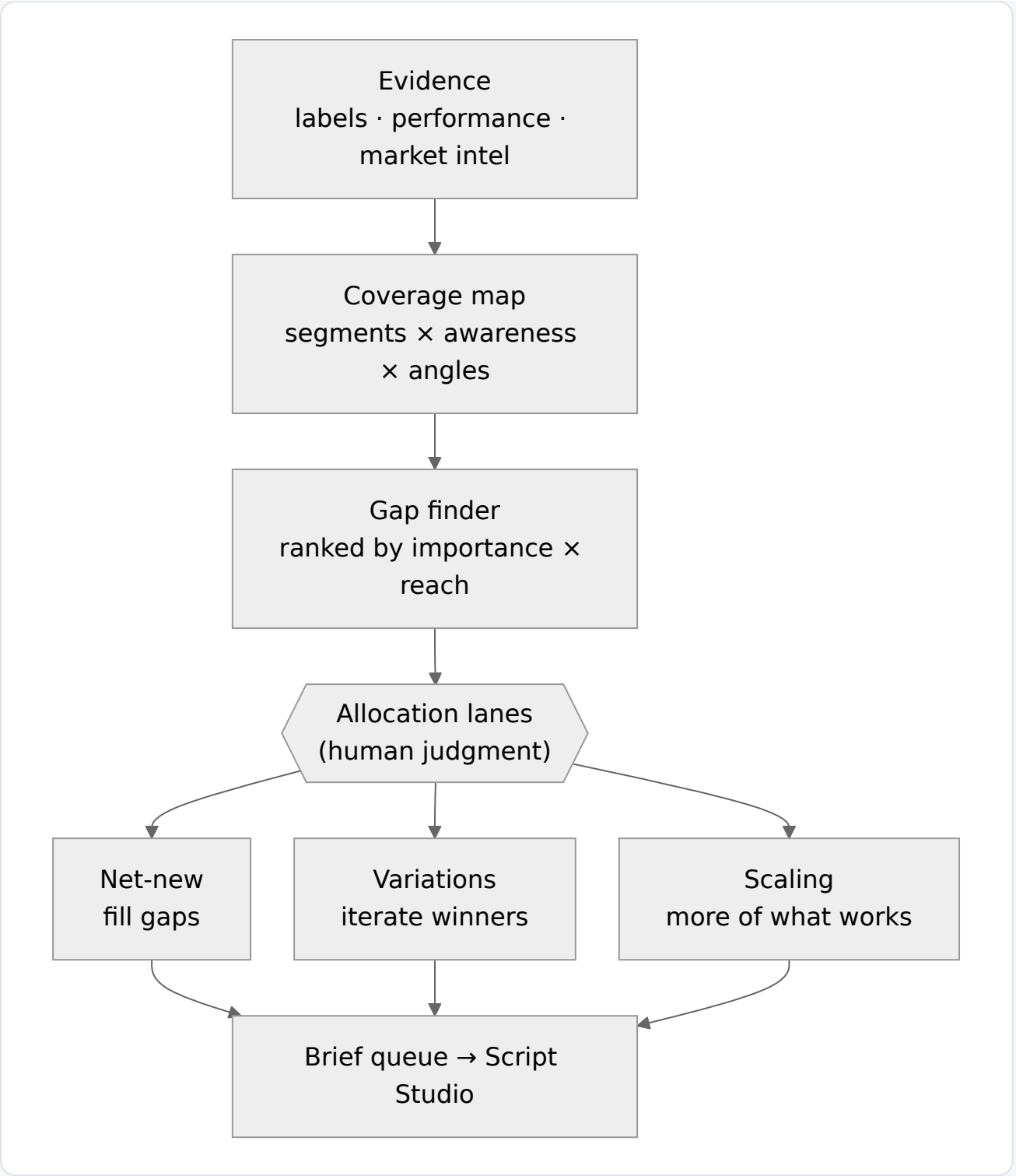

Fig. 3 — The strategy layer: what to make next is a measured decision. Every cell of the map carries evidence counts and confidence.

3.4 From insight to script (Script Studio)

A script starts from evidence: a scraped ad (+ its teardown), a set of market claims, one of our own winners, or a brief pulled from the coverage queue and synthesized from patterns and hypotheses. The **brain** assembles context automatically: product docs and reference assets, brand identity, our past

winners and their labels (from ecomprofits performance), long-term learnings (SuperMemory), and the selected evidence. Script generation runs as an agent session with the team's skills; drafts pass **approval gates** — a human approves or edits at checkpoints. Every run records exactly which evidence, prompts, overlays, and models produced it, so the team can always see what made what.

Three disciplines inside the studio:

- **The hook gate comes first.** Hooks are generated and approved before bodies and headlines, because the hook decides whether anything else gets seen. The app runs a hook session by overproducing, culling against the brand's rules and one job question — "would this target stop scrolling?" — and showing only the survivors; strategists pick keepers and kill the rest with reasons. Ideas are judged plain before polished wording, using live competitor evidence, a calibrated reference set, and the market's own language from the market-intel corpus as inputs. These references guide generation; they do not override claim validation or product truth. Approved hooks still pass brand-tone, claim, and truth checks.
- **A hypothesis ledger.** The team's bets ("we think X will work because Y") get a real home. Some briefs deliberately test one, and the result is recorded against it — hypotheses stop disappearing into chats.
- **The studio learns from editing — the instinct loop.** It does not learn a person's taste wholesale; it learns the checklist they use to approve or reject work, and runs it on every draft before review. For each strategist and deliverable type it keeps approved exemplars, two-tier rules (hard rules that auto-reject; tendencies saved with their context), before/after edit pairs, and a judge that scores drafts before review. Edits, picks, and kills with reasons are the signals — but the app only proposes captures, and humans ratify them. A one-off edit becomes a tendency; a hard rule requires recurrence or an explicit decision. Rules tagged "me" travel across deliverables; rules about a format stay with that format. A rule applies to that user's drafts after their review and to the whole team only after approval, so one person's taste never silently rewrites the system. Periodic rule audits keep the checklist honest. The loop starts with hooks and scripts, and only shipped or human-rated work enters the exemplar pool.

3.5 From script to creatives (Production)

An approved script (or a winning image, or a competitor ad) feeds the production **recipes**:

- **Images:** Explore (12 concepts → rendered statics → AI review), Expand (clone + localize a competitor ad with our brand), Long-Form story ads. Canva polish round-trip included.
- **Video:** Storyboard (script → scenes → frames → AI-rendered clips with narration → assembled cut), animated packs, and future video types.

New ad types plug into the same recipe system instead of becoming separate tools. **Statics are the testing ground:** hooks, angles, and concepts are probed as cheap statics first; winners graduate to video recipes, and the strategies behind winning statics become reusable input for briefs and video. Product accuracy is protected by a **locked product spec**: the app derives the product's exact look from its reference photos once, and that spec rides along in every render and edit. Approved creatives land in the winners gallery with full lineage: which evidence → which script → which session.

3.6 Launch with a creative code (the bridge)

When a creative is exported for launch, the app **registers a creative code** and stamps it into the ad-naming convention. A launch task is created in the app's workspace with the assets and the code attached — during the transition, mirrored into ClickUp where needed. The media buyer launches with that name; later, direct API publishing. From that moment the ad is trackable end to end.

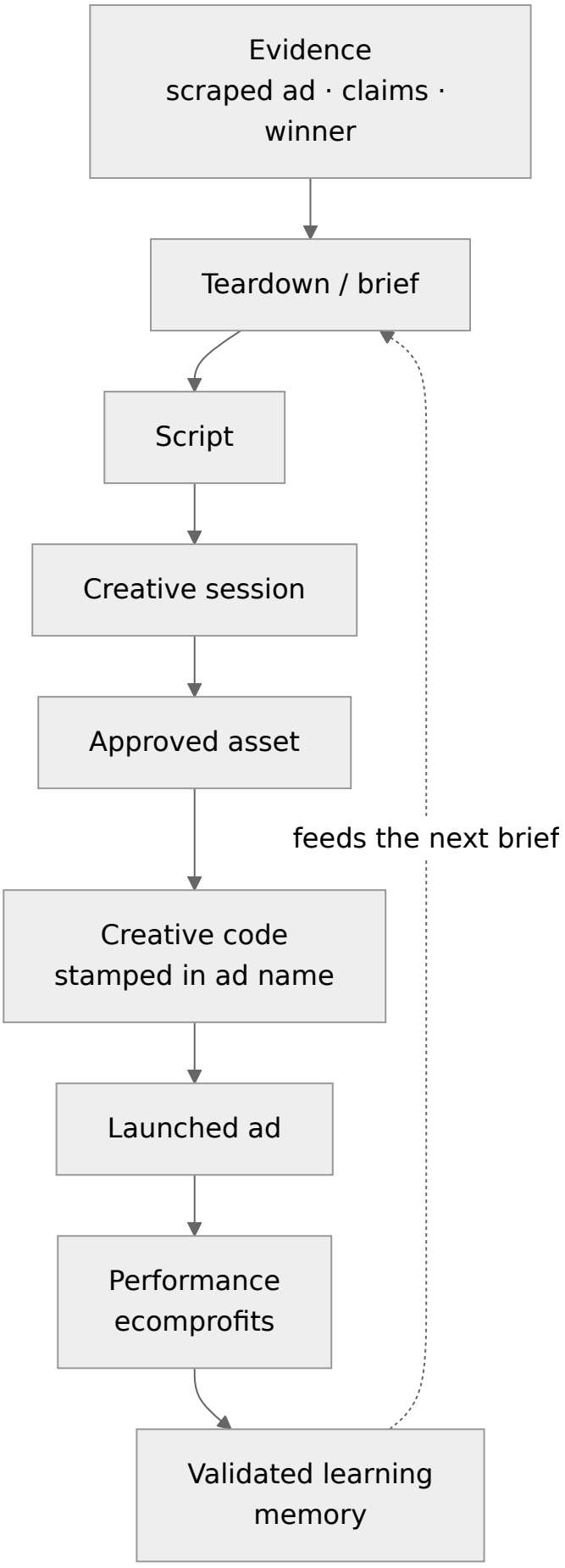
Evidence
scraped ad · claims · winner
Teardown / brief
Script
Creative session
Approved asset
feeds the next brief
Creative code
stamped in ad name
Launched ad
Performance
ecomprofits
Validated learning
memory

Fig. 4 — Lineage. Every object knows what produced it; the creative code is the link that closes the loop.

3.7 The loop closes (automatic)

ecomprofits ingests performance from all channels and parses the naming convention. The app reads results back per creative code. Winner/loser status is derived by the team's rules, with human confirmation where required. Hook rate is read separately (did people stop scrolling?), so hooks are judged on their own job — and the hook gate learns from early stop-scroll signals where available, not only from final cost per result. Winning patterns per product and funnel stage are aggregated; **validated** learnings are written to long-term memory ("statement-hook UGC statics win for DE pelvic-floor audience"); the next script or brief can use them automatically. Personal taste artifacts stay separate from this shared memory — what a strategist likes never mixes with what the market proved — and only shipped or human-rated work ever becomes an exemplar. Underperforming angles are flagged, every result updates the coverage map, and the app proposes the next batch from what the data says.

3.8 Autonomy, gated

Recurring jobs run without anyone asking: scrapes, classification, research runs, performance sync, pattern refresh, weekly proposal batches ("10 recommended scripts/creatives from this week's intel + performance"). Autonomous work follows gate policy: internal analysis runs on its own; public output, launch decisions, and anything high-risk stops for human approval. AI work is cost-metered behind a **spend guard**: a rolling spending ceiling with a warning level, an automatic pause level, and a manual kill switch — broken down per provider and per job. Cost per creative is visible.

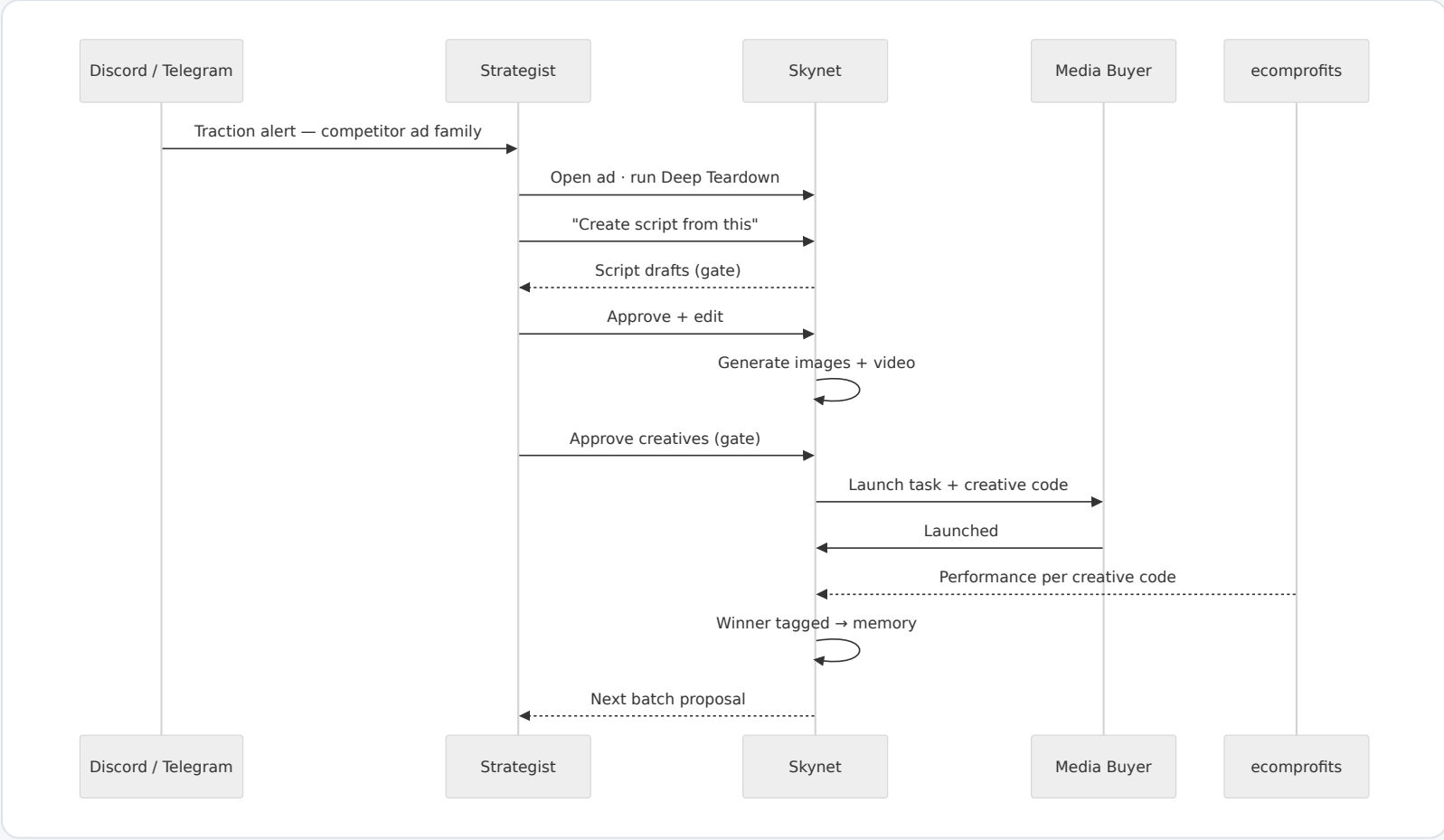

Fig. 5 — A competitor ad becomes a launched, tracked ad — with two human gates and no manual file-shuffling.

4 User stories

CREATIVE STRATEGIST

1. I get a Discord / Telegram alert that a competitor's new ad family is gaining traction. I open it in the app, see its labels and how long each variant has run, and run a Deep Teardown.
2. From the teardown I click **"Create script from this"**. The app drafts scripts using our product docs, our past winners for this audience, and the teardown. I approve at the gate, edit where I want.
3. I run a hook session: the app generates a large batch, culls it against my rules and the stop-scroll test, and shows me the survivors; I pick keepers and kill the rest with reasons — the next batch is sharper.
4. I browse a competitor's full library filtered by angle and format and see what they lean on — and what they quietly stopped running.
5. I search real customer language (Amazon, Reddit, Trustpilot) in the market-intel corpus for a niche and pull exact phrases into a script brief; I search the evidence corpus (PubMed) for the same niche and pull the plain-language findings and citations that explain why the problem happens.
6. I open the coverage map for a product and see where our messaging is thin — by segment, awareness stage, and angle. The gap queue tells me what to make next, and why it matters.
7. I review the weekly proposal batch and approve, edit, or reject each item; my personal prompt overlay tunes how drafts are written for me.
8. My edits, picks, and kills at the gates teach the system: the app proposes rules from those signals, applies them to my drafts after my review, and shares them team-wide only after approval.
9. I tag a teammate on any ad or claim with a note; it lands in their inbox with the evidence attached.

COMPETITIVE-INTEL / RESEARCH OPERATOR

10. I add a new competitor once, link it to several products, and the app reuses the same tracked evidence everywhere it is relevant.
11. I review low-confidence labels and gold-set checks so the taxonomy stays honest — and I can see exactly how accurate the labels are.

MEDIA BUYER

12. I pick up a launch task that already carries the assets and the correct ad name (with the creative code). I launch and mark it done — no manual renaming, no wrong dropdowns.
13. I see performance per creative appear automatically, joined to what produced it; winners and losers are flagged by our rules.
14. When something wins, I can see exactly which script and evidence produced it — so I know what to ask for more of.

DESIGNER / VIDEO EDITOR

15. I receive storyboard deliverables (scenes, clips, narration, manifest with exact cut lengths) in my inbox and assemble the final cut — or accept the app's auto-assembled cut.
16. I open a generated static in Canva, polish it, and return it — the polished version stays linked to the original.

PRODUCT RESEARCHER

17. I manage keyword seeds and run research crawls; I get a ranked product list with the reasons for each score stated (and gaps shown honestly).
18. I promote a winning opportunity into a tracked product — competitors and intel come pre-linked, ready for the script → creative → launch loop.

OPERATIONS / WORKSPACE MANAGER

19. I manage launch queues, fulfillment state, and the board-by-board ClickUp transition without breaking the team's current process.

TEAM LEAD

20. I see one pipeline: what is in research, scripting, production, review, and launch — and where things wait on a human.
21. I set the split for the next batch — net-new concepts vs variations vs scaling — and the app fills each lane from the coverage map and the winners.
22. I see cost per creative and per AI job, with caps I control.

23. I set the approval-gate policy: what runs autonomously and what stops for sign-off.

LEADERSHIP

24. I see one dashboard: where work is blocked, which products are moving, which creatives are winning, and where cost is going.

ADMIN

25. I change which AI model runs which job from a settings page — no redeploy. I manage keys, budgets, users, and roles in the app.
26. I audit any output: which evidence, prompt versions, personal overlays, and model produced it.

ANY TEAM MEMBER · TELEGRAM / DISCORD

27. I get alerts and can respond with quick actions ("assign to X", "start a script"); the full work always lives in the app.

---

5 What changes for the team — and what does not

CHANGES (THE TARGET STATE)

- One login, one app, one inbox — instead of four tools plus ClickUp.
- The download → Drive → ClickUp → rename → upload chain disappears; the creative code makes every ad trackable automatically.
- The script workflow with approval gates becomes available to the whole team, not one operator.
- ClickUp's UI is replaced board by board — only after leadership

DOES NOT CHANGE

- ecomprofits stays the system of record for performance — dashboards and reports keep working untouched.
- ClickUp keeps working during the transition; nothing is switched off until its replacement is accepted.
- Existing tools keep running while their modules move into the unified app; data is migrated, not recreated.

approves, and only when each board's replacement is proven. The ClickUp bill drops at cutover completion, not day one.

- No team switches tools overnight; each operational surface moves only after it is proven and signed off.

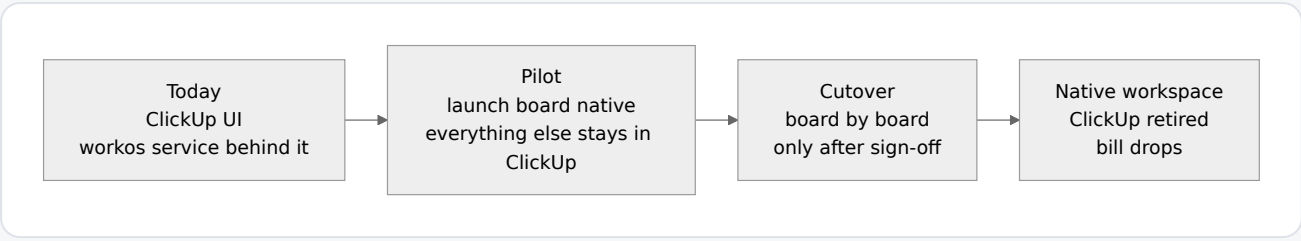

Fig. 6 — The ClickUp transition. Gradual, gated, reversible at every step.

---

6 Rollout phases

Phase 0 — **Foundation.**
One product shell, one login, one UI, shared product contracts. The existing tools' features appear in one place, unchanged. Team impact: log in once, find everything.

Phase 1 — **The closed loop (first slice).**
Competitor ad → teardown → script → images/video → export with creative code → launch task → performance readback → winner tag → lineage + initial memory capture. The launch board is the first native workspace pilot, if leadership approves that pilot. The coverage store starts filling from day one — every label and creative code lands on the map. This proves the whole idea end to end on one product. Also in Phase 1, right after the first slice: the Reddit VOC pipeline (keyword groups, continuous collection, master dataset, export) in the Intelligence module, and the PubMed evidence pipeline (six lenses, classified findings, MCP, export) after it in the same module.

Phase 2 — **Script Studio + the strategy layer.**
The full agent runtime for the team: skills, gates, scheduled runs, per-user overlays. The strategy layer goes live: the coverage map

with gap-driven proposal batches, the instinct loop (hooks and scripts first), and the hypothesis ledger. The live market-intel corpus (Reddit / Trustpilot / Amazon) and the evidence corpus (PubMed) are wired into briefs, script generation, and review; later sources such as YouTube are added source by source.

Phase 3

Workspace cutover + proactive layer.

Board-by-board ClickUp replacement; traction alerts; the variation engine (defined iteration knobs on winners: concept, angle, style, hook, edit, audience); deeper video intelligence.

Ongoing

Growth.

New image/video recipe types; multi-brand / external hardening later if we ever choose to productize it.

---

7 Decisions we need (the approval box)

FOR LEADERSHIP REVIEW

1. **Approve this functional design** — the loop, the modules, the phases. **Approved by Mathew** ("good to go"). Mathew's later Reddit VOC and PubMed evidence requests are folded in above (§3.1, Phase 1) and are built in the unified platform, as agreed with him.
2. **RescaleOS migration** — its owner's sign-off and a coexistence plan while his live loops keep running.
3. **ClickUp replacement** — confirm the existing Native Workspace decision points D1–D4 (funding, ownership model, pilot sponsor, store hosting).
4. **Workspace transition ownership** — who owns the rollout, support, and board-by-board sign-off during the ClickUp transition.
5. **Pilot scope** — confirm the first pilot product(s) and the first native workflow (we propose the creative-demand/launch board). **Approved in principle with this design**: the first native workflow is the launch-board pilot; the exact pilot product is chosen in discovery (gate G2).
6. **Taxonomy ownership** — who approves vocabulary changes (angles, formats, awareness), and at what cadence.

7. **Naming convention** — approve adding the creative code to the ad-naming convention (small, backwards-compatible change). **Approved with this design.**
8. **Winner/loser rules** — the thresholds the app should use, and where human confirmation is required.
9. **Model policy** — confirm the hybrid approach: standard API routing by default, with the subscription-based agent runner allowed for internal workloads as a cost lever. **Approved with this design.**
10. **Pre-pilot discovery** — document how each brand launches statics today (request source, approval chain, asset packaging, naming, account destination, who publishes, where it breaks) before the launch-board pilot. Required, not optional.

---

8 Open questions for the team

- Which video ad types should the first video recipes target (beyond storyboard + animated packs)?
- Alert thresholds: what counts as "competitor traction worth pinging the team"?
- Which boards after the launch board should cut over from ClickUp first?
- Which products should the Phase 1 pilot run on?
- The coverage map's segment axis: confirm the segment/persona vocabulary per product (the awareness and angle axes come from the existing taxonomy).

---

9 The team's system map — what it changed here

A full system map of the creative machine was shared during review. We compared it against this plan node by node and arrow by arrow. The two models agree on the core loop — the map validated this plan's shape. Each was stronger in places, and this plan now takes the best of both.

STRONGER IN THE SYSTEM MAP — NOW ADOPTED

- **The coverage map + gap finder + allocation** (net-new / variations / scaling) — the map's central idea, and a stronger planning model for this system than pattern-mining alone. Added as the strategy layer (§3.3, Fig. 3), with the coverage store filling from Phase 1 and the full layer landing in Phase 2.
- **Learning from human gate decisions** — edits, picks, and kills with reasons become proposed checklist rules that improve future drafts (§3.4), with review before anything applies to the whole team.
- **The hook gate as the most important gate** — hooks are generated and approved as their own unit, and hook rate is measured back separately (§3.4, §3.7).
- **A hypothesis ledger** — the team's bets get a real home, and results are recorded against them (§3.4).
- **For later phases:** the variation engine with defined iteration knobs on winners (Phase 3); seed-bank discipline for saved ideas; balancing gap-led and idea-led ideation against an inventory target.

STRONGER IN THIS PLAN — KEPT UNCHANGED

- **The creative code** closing the loop from launch to performance. The map ships to Meta and reads account data, but leaves the link between them open — a wire this plan already handles explicitly.
- **Label honesty** — classification (what an ad is) and scoring (how it performed) are separate instruments; label accuracy is measured against a human-labeled gold set; coverage and sampling bias are always shown. The coverage map inherits this: every cell carries evidence counts and confidence.
- **Multi-user operations** — gates with owners, one inbox, roles and permissions, and a full audit trail of what produced every output.
- **Cost control** — AI work metered, capped where configured, cost per creative visible.
- **The wider platform** — work management with the gradual ClickUp transition, the product-research lane, and the market-intel and evidence corpora with per-source access and sampling caveats.

Terminology bridge (for discussions — same things, two vocabularies): *primers* = the context packets the app assembles for each generation; *seed*

bank = saved ideas and swipes (collections); *gambits* = hypothesis-ledger bets; *the cube* = the coverage map.

Further team inputs folded in during review: the instinct-transfer protocol (how the studio learns, §3.4), the hook-engine method (the hook gate, §3.4), and statics-first testing (§3.5).

---

10 A market check — what a competitor teardown showed

In August 2026 we studied a commercial AI ad-production platform in depth. Two conclusions matter for this plan.

It validated this plan's direction. The competitor has excellent generation craft — and nothing after "generate". No performance readback, no attribution from launched ads back to what produced them, no competitor or market intelligence, no strategy layer, no learning loop. Those are exactly the parts this plan is built around: the closed loop is the differentiator, not a nice-to-have. Their weak spots also confirmed our engineering choices — durable server-side jobs (theirs orphan when a browser tab closes), full lineage records (theirs keep a whole project in one blob), and private storage (theirs leaks prompt fragments publicly).

We adopted their best craft into this plan. Where they are genuinely strong is generation operations, and this plan now carries those ideas: the guided prompt authoring with layer guardrails (§3.0), the spend guard (§3.8), and the locked product spec (§3.5). More of their mechanics enter the technical design list: providers pluggable as configuration, context blocks with explicit authority rules, prompt-safety checks that catch phrasing image models render wrong, and safe propagation of prompt improvements to users who customized them.

The bar it sets is user experience. Their production flow feels like one polished machine — stage by stage, skippable at every step, quality checks built in. The unified app's production flows are designed to meet that usability bar, while adding the intelligence, attribution, and learning layers they lack.

Prepared by Agon · Feedback: reply in the team channel or tag Agon.