

Skynet — Team hand-off (what to read, in which order, and what it means for you)

For: Vince, Sead — and Mathew, Bian, Cyrus, Sven for the parts that name them.

1. In one paragraph

Skynet is one app that replaces four tools and the manual steps between them: the static-ads tool, the Meta ads scraper, RescaleOS and the ClickUp back-end (workos), with ecomprofits staying the performance system of record. It is built in five slices. The first slice puts the unified app live on `server.ecomprofits.io` with Production native and the creative-code loop proven on one real ad; the later slices merge Intelligence, Studio and Workspace natively and end with SaaS hardening. The project is delivered only at 100 %. There are no calendar dates — only a sequence with gates. Best implementation, no quick fixes.

2. The documents — read in this order

All files are in `development-processes/skynet/final/`. The main documents are available as markdown, HTML and PDF; `decisions/ADR-010-functional-design-approval.md` and `skynet-techdesk/appendix-b-import.json` are source-format only.

#	DOCUMENT	READ	WHY YOU READ IT
1	Functional design (skynet-unified-app-functional-design)	§1–§5 fully (~30 min)	What the app is and does; the 27 user stories; approved by Mathew. Everything else derives from it.
2	Workflow source map (skynet-workflow-source-map)	fully (~45 min)	How each of the 16 workflows runs today, how it runs in Skynet, and where every step comes from: PORT (rebuilt natively from the static-ads tool), PORT (phased) (scraper, RescaleOS — merged later; bridge track B is the only allowed standalone exception while S1 runs), REUSE (workos on Railway, via API), READ (ecomprofits, never written), NEW , REPLACES MANUAL . The legend in §2 is the rule you build by.
3	Architecture and deployment design (skynet-architecture-and-deployment-design)	§1–§9 and §12 (~25 min); §10–§11 as reference	Where things run and the boundaries between them; the monorepo layout; the stack; the public URLs and host paths; how a push to main reaches production; the resource limits on the shared box; the S1 exit gate.
4	Delivery plan (skynet-delivery-plan)	§1–§6 fully (~20 min); §7 when you need the later-slice sequencing	The rules, what slice and wave mean, the slice gates, the S1 waves — who does what and in what order, and what must be true before the next wave.
5	Task specs (skynet-task-specs)	your lane's table, then every card of yours in the current wave (reference thereafter)	Each card: context, files to read first, deliverables by name, acceptance command, out of scope, what it waits on and what it unblocks. Every card is written to be started from its text alone — if you cannot, that is a defect in the card, not in you: say so on the task.
6	Preparation pack (skynet-preparation-pack)	§2 and §4 (~10 min)	The working agreements, the Definition of Done, and the access checklist to clear before work starts.
7	ADR-010 (decisions/ADR-010-functional-design-approval)	fully (~2 min)	The approval record for the functional design and the Reddit VOC addendum.

#	DOCUMENT	READ	WHY YOU READ IT
8	TechDesk import (skynet-techdesk/appendix-b-import.json)	only if you load or audit the task set	The import payload for the same card set as the task specs; use it to verify keys, titles and owners if TechDesk needs a reload.

The folder `archive/` holds the earlier drafts. They are history; where they differ from the final set above, the final set wins.

2a. Day-one orientation

- **Repo.** The codebase is the monorepo `Rescale-Media/skynet`. Production lives at `/opt/skynet` on `server.ecomprofits.io`; staging at `/opt/skynet-staging` on `server.rescale.media`.
- **URLs.** Production `app.rescale.media`; staging `staging.app.rescale.media`. The scraper and RescaleOS stay on their current hosts and are reached by links only.
- **Terms.** A `slice` is a sequential delivery block (S1–S5). A `wave` is the gated batch inside a slice; the next wave starts only when the current gate is true.
- **TechDesk** is the working task board: stand-ups, blockers, acceptance artefacts and decision comments live on the task. The JSON above is the import payload for the same cards.
- **If a card cannot be started from its text**, comment that on the task immediately and treat it as a card defect.

3. The decisions you are building on (do not re-open)

1. **Host.** The unified app runs on `server.ecomprofits.io` (production) with hard resource limits; staging on `server.rescale.media`.
2. **Standalone apps stay standalone.** The scraper and RescaleOS keep running on `server.rescale.media` until their functionality is merged natively (S2, S3). The shell links to them; nothing is mounted or proxied. Bridge track B is the only allowed standalone change set before those merge slices.
3. **workos stays on Railway** and is reused over an API; **ecomprofits is read-only**.
4. **Only static-ads is ported in S1.** It is the product base for the unified app.
5. **One creative-code generator, one counter** (workos grammar, `rescale_service.next_naming_seq`). Never a second one.
6. **CI/CD and alerting follow the ecomprofits pattern** (GHCR → SSH → compose → Telegram), with digest pinning and a staging smoke gate before prod.

7. **No dates. Delivered = 100 %. No quick fixes.** A card that would need a shortcut is split or re-planned, never closed short.
8. **S2–S5 are decided:** real port of the scraper data (no PostgREST shim), engine on the production box as a stack service, native Intelligence UI; Studio fully on the stack with in-app skills/prompts; only the 13 creative-workflow ClickUp lists are replaced; SaaS = multi-tenant invite-only, manual invoicing.

4. How we work (the short version — full text in the preparation pack §2)

- **One card, one owner, one branch.** Branch per card, ≤ 2 days; PR title starts with the card key.
- **Contracts first.** Every cross-lane seam is a schema + fixtures in `packages/contracts` by the contract wave. Build against the fixture, not against someone's branch.
- **Every card goes through draft → hard review → fix → re-verify before a human looks.** Human cross-lane review only for contracts, migrations, auth, compose/Caddy, workflows.
- **Done = merged + acceptance artefact posted on the TechDesk task + docs named in the spec exist.** No artefact, no done.
- **Blocked = a comment naming the blocker + the unblock condition.** The upstream owner ships a stub or fixture the same day.
- **Discovered work → a new card first.** No silent TODOs, no "while I'm here" in a sidecar.
- **Daily written stand-up** in TechDesk (one line per card: yesterday / today / blocked). Gate reviews at the end of each wave. No standing meetings.

5. What each person does first

Sead (Lane C — data spine and bridge). Read the map's W10 and W12 first — the creative code and the readback are your seam and the loop depends on them. Your S1 cards: S0a (workos scrub, with Sven), S1a/S1b (core auth), S3 (creative-code service), S4/S4b (lineage, bindings contract), S5a (ecomprofits read layer), S8a (the narrow workos bridge), S7 (readback), S10 (the loop with one real ad), S12, S13, M5.

Vince (Lane B — production and shell). Read the map's W7–W9 first — that is your tool, ported. Your S1 cards: V1 (SSO in Production), V2a (shell), V3 (Production on unified Postgres), V4 (bindings + spend guard), V6 (shell with links), M1, V7 (creative code on export + lineage), V8, V10, V5, V13, X2, M2.

Agon (Lane A — platform). A0 box prep on both boxes, A1a/A1b (repo + CI), A2/A2c/A3 (compose, limits, Caddy), A3b (deploy pipeline), A6 (staging), X5/X7 (drills, prod stack), X3/X3b/X4/X4b, X1, M3, M4, M6.

Before the build starts, all three: tick the access checklist (preparation pack §4); create your TechDesk account; read your lane in the task specs and flag any card you cannot start from its text.

6. What we ask of the others

- **Mathew** — approved the functional design; the Reddit VOC ask is in (S2, A18). His this-week asks run as **bridge track B** next to S1: first the PubMed `/evidence-layer` skill in RescaleOS (six lenses, no scraper change needed), then `/lfs` + the Primal Queen continuous job (ranked long-copy ads with Grok breakdowns, filed into his folders). The bridge is temporary and is retired when the native flows land. Later: the hook/strategy interview (D32) and the VOC seed groups (D23/D23b).
- **Bian** — bridge track B may add temporary RescaleOS features before S3 (`/evidence-layer` , `/lfs` , related docs/rules), but the console remains the live standalone until the Studio merge and cutover; before S3 we need his explicit coexistence agreement (D31).
- **Cyrus** — nothing changes for creative and media buying in S1 except that exported creatives carry the creative code and the package appears on the Launch card. Later: the launch-process interview (D41).
- **Sven** — the S0a credential scrub with Sead, and a short confirmation of the S1 bridge scope (two endpoints in the old workos repo, deployed on Railway).
- **Leadership** — D1–D4 gate the Workspace cutover (S4) only.

7. What the first slice proves — and what it does not

Proves (the S1 exit gate): `app.rescale.media` serves the unified app from `server.ecomprofits.io` ; one login for Skynet itself (the scraper and Studio keep their own logins until their merge slices); Production fully native at parity; the loop from export to performance readback on a real ad; model bindings and spend guard; the export package on the existing ClickUp Launch card; staging, CI/CD, backups, rollback, monitoring.

Does not include: the Intelligence and Studio merges, the strategy layer, the launch-board pilot, ClickUp replacement, or SaaS features — those are S2–S5, already scoped to card level in the same documents.

8. Where to ask

Questions about a card → comment on the TechDesk task; the owner keeps the decision trail there. Questions about a decision → Agon; decisions live in `decisions/` in this pack now and `docs/decisions/` in the repo, plus the delivery plan, never only in chat.