SOULWIRE STUDIO

AI AUTOMATION FOR OPERATORS

The Audit Deck Automation

Discovery calls that write their own follow-up.

A complete build guide for turning a recorded discovery call into a pre-populated audit deck, delivered to a URL you can review on your phone in under two minutes.

You stop writing audit decks from scratch after every call. You start editing ones that already exist.

SECTION I

What You're Building

You get on a discovery call with a prospect. Fireflies records it. The moment the transcript is ready, an automation reads it, pulls out the pain points the prospect actually described, matches them against a library of solutions you've already delivered, calculates the cost of doing nothing, and delivers a mostly-finished audit deck to a URL you can review on your phone in under two minutes.

You edit pricing, the roadmap, and drop in a hero image, then walk into the follow-up call already 80% ready. The point isn't that the deck is finished. The point is you never write an audit deck from scratch again — whether what you sell is an automation, a campaign, a retainer, or a rebuild.

THE FLOW

```
Discovery call runs
   |
Fireflies records → fires webhook when transcript is ready
   |
n8n fetches the full transcript
   |
Claude extracts pain points, objectives, tools, headcount, revenue
   |
Airtable Workflow Library queried   ← verified builds only
   |
Claude matches pain points to library entries
   |
Claude calculates cost of inaction   ← stated numbers only
   |
Payload built → POSTed to Replit → deck goes live
   |
Telegram notifies you: pain points, matches, COI summary
```

> **THE RULE THAT MAKES THIS WORK**
>
> Every Claude step in this build is grounded on purpose. It's told to never invent tools, headcount, revenue, or dollar figures that weren't actually said on the call. A blank slide is correct behavior. A hallucinated one is the thing that gets you called out mid-pitch.

SECTION II

What You Need

01 **n8n**

Orchestrates the whole pipeline: receives the Fireflies webhook, runs the three Claude reasoning steps, queries Airtable, and posts the finished payload to Replit. Self-hosted or cloud both work.

02 **Anthropic (Claude) API key**

Powers three separate reasoning steps: extraction, matching, and cost-of-inaction. One credential, reused across all three nodes. console.anthropic.com.

03 **Airtable**

Your Workflow Library. This is the matching brain, and it's worthless empty. Seed it with builds you've actually shipped before you expect the pipeline to produce anything useful.

04 **Fireflies**

Records the call and fires a webhook the moment the transcript's ready. Needs a paid tier that supports GraphQL and outgoing webhooks.

05 **Telegram bot**

Pings you when the deck's ready to review, with the pain points, matches, and COI summary. Free, via BotFather.

06 **Replit**

Hosts the deck itself: a React app plus an API endpoint your n8n workflow posts to. Needs a paid tier so the app doesn't sleep between calls.

Rough monthly cost floor if you're starting from zero: $70 to $100 a month, mostly Anthropic, Replit, and Fireflies.

SECTION III

Step-by-Step Build

01 Deploy the Replit deck and API

Two apps, one pnpm monorepo: a React + Vite deck served at `/audit-presentation/`, and an Express API served at `/api-server/`. Storage lives in Replit Object Storage under a single key, not a database and not a file on disk, so it survives a cold start without losing state.

REFERENCE: API ENDPOINTS

```
GET  /api-server/api/presentation        no auth, returns current deck JSON
POST /api-server/api/presentation        Bearer token, writes new JSON
PUT  /api-server/api/presentation/editor  no auth, editor-only
DELETE /api-server/api/presentation      Bearer token, resets to sample data
```

Set two env secrets in Replit: `PRESENTATION_API_KEY`, the bearer token n8n will use to post payloads, and `DEFAULT_OBJECT_STORAGE_BUCKET_ID`, which sets itself once you attach Object Storage. Pricing, roadmap colors, and the hero image all get set later in the deck's own editor, not through the automation.

02 Build your Solutions Library

This is the pattern-matching brain, and it's the one part that can't be handed to you pre-built. Whatever your business actually delivers, whether that's n8n builds, ad campaigns, financial plans, or security audits, seed the library with your own real, closed work. Seed it thin or with stuff you can't actually deliver, and the matcher will recommend exactly that.

SCHEMA

Workflow Name, Problem It Solves, Source Type, Source, Source Link, Industry/Niche, Tools Used, Complexity, Deliverable, Tags, Date Added, Review Status, Entry Type. Rename fields to fit what you sell (Tools Used becomes Channels Used for a marketing agency, Frameworks Applied for a consultancy) but keep the shape. The matcher only queries records where Entry Type is Applied Solution and Review Status is Verified, so Base Techniques can sit in the library for scoping reference without competing for matches. Six real entries is enough to go live. Add three more every time you close a deal.

03 Adapt this build for your niche (optional)

Skip this if you're building for AI automation exactly as written. Everything in this build (transcript in, pain points out, matched against real past work, deck out) works for any business that runs a discovery call before a proposal, but the JSON and its three Claude prompts are still written in AI-automation language. Adapt it before you import, not after.

COPY THIS: THE ADAPTATION PROMPT (FOR CLAUDE CODE)

```
I'm adapting the Audit Deck Automation build for a different
niche. My business: [YOUR NICHE]. What I deliver to clients:
[YOUR DELIVERABLE].

Here is the n8n workflow JSON and the build guide markdown.
Go through both and:
1. Rewrite the three Claude system prompts (extraction, matching,
   COI) to match my niche's language. Keep every grounding rule
   intact word-for-word: never invent tools, headcount, revenue,
   or dollar figures not stated on the call.
2. Rename the Airtable field references the nodes depend on
   (Tools Used, Entry Type, etc.) and update every node that
   references those fields.
3. Rename the Replit deck's slide labels to match what I sell.
4. Leave the pipeline mechanics untouched.

Show me a diff before applying anything.
```

WHY THIS STAYS MANUAL

Auto-renaming fields and prompts without a human reading them risks breaking a node reference or softening a grounding rule you didn't mean to touch. Review the diff the same way you'd review pricing or a hero image: judgment doesn't run unsupervised.

04 Import the n8n workflow

In n8n: Workflows → Add workflow → Import from File, and select the workflow JSON (adapted or original). Every credential is stripped on export by design; you'll attach your own in the next step.

REFERENCE: WHAT EACH NODE DOES

```
Fireflies Transcript Ready       webhook, fires when transcript's ready
Normalize Webhook Payload        handles field-name drift across payloads
Fetch Full Transcript            GraphQL call for sentences + attendees
Extract Pain Points              Claude step 1: industry, pain, tools, revenue
Search Workflow Library          queries verified Applied Solutions only
Match Against Workflow Library Claude step 2: fewer matches over weak ones
Calculate COI Estimate           Claude step 3: stated numbers only
Build Presentation Payload       assembles the final JSON
POST to Replit Presentation APIsends it live
Notify Telegram for Review       pings pain points, matches, COI
```

05 Wire up credentials

Airtable nodes need a personal access token with `data.records:read` scope on your Workflow Library base. Anthropic nodes need one API key, created once and reused across all three Claude nodes. Fireflies needs a bearer token, Telegram needs a bot token from BotFather.

> **THREE PLACEHOLDERS STILL NEED MANUAL SWAPS**
>
> The POST node has your Replit URL and API key hardcoded as placeholders. The Telegram node has your chat ID and Replit URL the same way. Find-and-replace on `REPLACE_WITH_` after import surfaces every spot that still needs your real values.

06 Wire up Fireflies

Activate the workflow in n8n and grab the production webhook URL from the Fireflies Transcript Ready node. In Fireflies: Integrations → Webhooks → add webhook, paste the URL, set the trigger to "Transcription completed." Test with a 30-second recording and confirm n8n shows an execution.

07 Test the full flow

Two ways to do this. Run an actual test call, with someone playing a prospect describing two or three pain points that map to your library. Or skip waiting on Fireflies entirely: inject a saved transcript ID straight into n8n's Execute Workflow. When it works, Telegram fires 60 to 90 seconds after the transcript completes, and the deck shows real pain points, real matches, and a COI number that's either a genuine estimate or blank. Never invented.

WHAT STAYS MANUAL, AND WHY

Pricing is yours to set per client based on scope and posture. Roadmap sequencing needs your judgment about what to promote from "opportunity" to "core." The hero image is where a real photo of the prospect's operation beats any placeholder. Automate these three and the deck looks automated. Leave them manual and it looks considered.

WHAT'LL BREAK, AND HOW TO READ IT

No library matches means your library isn't seeded deep enough yet, not that the pipeline's broken. A blank COI means the prospect didn't state numbers on the call, which is correct behavior, not a bug. If you soften a grounding prompt to "always fill every field" to make either of these go away, you'll get a deck that invents facts instead. Don't loosen them.

SECTION IV

The Upgrade Path

Run this through a handful of real calls before you touch anything. Those first few decks show you where extraction gets sloppy or a match feels forced, and that's what tells you whether the problem is your prompts or your library. Tighten from evidence, not from guessing.

From there the same skeleton scales sideways. Point the matcher at proposal generation once the deck's proven itself. Feed the library into a second pipeline that turns verified work into a public case study page. Fork the deck template per vertical if you sell into more than one industry, since the slide order stays the same and only the framing changes.

YOUR NICHE	WHAT FILLS THE LIBRARY	WHAT THE DECK SELLS
Marketing agency	Past campaigns, tagged by channel and result	Channel mix and projected lift
MSP / IT provider	Security frameworks, past remediation projects	Risk score and remediation roadmap
Financial advisor	Planning packages, past client outcomes	Gap analysis and projection
Management consultant	Frameworks and closed engagements	Diagnosis and 90-day plan
Recruiter / staffing firm	Past placements by role and industry	Pipeline plan and time-to-fill estimate