

Prospect Dashboard — Scoping Doc

From: Loom Date: 2026-05-19 Status: Pre-build — owner decisions needed before kickoff

TL;DR Recommendation

Build it inside Atlas. Atlas owns the build, Riv supports the inbound capture seam (Gmail/LinkedIn → Atlas).

- A B2B bookkeeping practice with ~40 clients does **not** have CRM-shaped volume. You're likely managing 5–25 active prospects at a time, not 500.
- A simple Atlas table + kanban view = one source of truth, lives next to the rest of PKA (Echo can drop emails in, Vox can flag inbound LinkedIn leads, no SaaS subscription, no babysit tax).
- A real CRM (HubSpot/Pipedrive/Copper) gets you running faster but adds **another tool to log into**, another data silo from Echo/Atlas, and a monthly bill for features you won't use.
- The deciding factor: Echo and Vox already handle the upstream channels where leads arrive. Keeping the pipeline inside PKA means inbound capture is plumbing, not integration.

I am explicitly **not** building yet. I need the audit answers below before Atlas writes a schema.

What I Don't Know Yet (audit questions for Jimmie)

I can't produce a real SIPOC without these. Quick answers — bullets are fine.

#	Question	Why I'm asking
1	Where do prospects come in today? LinkedIn DMs (Vox), Gmail inquiries (Echo), referrals, web form, phone, networking events? Estimate the % split.	Determines the capture seam Riv needs to build
2	Where are they tracked right now? A Sheet? A Trello board? In your head? Nowhere?	Migration scope. Also tells me how much pipeline data exists today vs. starts-from-zero.
3	Roughly how many active prospects at any given time? 5? 20? 50?+	Sizing the build. ≤ 25 = lightweight table is fine. 50+ starts to want CRM features.
4	What does your current sales process actually look like? First reply → discovery call → proposal → signed? Or different stages?	I need to know the stage taxonomy before designing the dashboard.
5	Who else touches prospects? Just you, or do the remote bookkeepers see them too?	Determines read/write access in Atlas

6	What does the dashboard need to DO when you open it? Show counts and stale ones (passive)? Or prompt next action (active)?	Passive = simple table. Active = needs a "next action" engine. ADHD-friendly = the second.
7	Any prospects you'd consider "the one that got away" because you forgot to follow up?	If yes → stale-prospect alerting is a Day-1 feature, not a Phase-2.

Provisional Scope (subject to audit answers)

Pipeline stages — strawman to react to

New Lead → Qualified → Discovery Call Scheduled → Proposal Sent → Signed
 ↘ Lost / Nurtured

Reject, redline, or replace. This is just a starting point — every firm's stages differ.

Dashboard surfaces (strawman)

1. **Counts by stage** — how many prospects sitting at each stage right now
2. **Stale list** — anything untouched > N days (you set N per stage; "Proposal Sent untouched 7 days" matters more than "New Lead untouched 14 days")
3. **Next action queue** — top 5 prospects needing your attention today, sorted by stage urgency
4. **Source attribution** — where leads came from (LinkedIn / referral / web / etc.) so Vox can see what's working
5. **Won/lost tally** — close rate by stage, by month

Data model (strawman)

Field	Notes
name	Person + company
source	LinkedIn, referral, web form, etc.
stage	From the taxonomy above
owner	Default: Jimmie. Future: remote bookkeepers if relevant.
last_touched	Drives the stale alerting
next_action	Free text — "send proposal", "follow up on discovery"
next_action_due	Date
notes	Append-only log
outcome	Won / Lost / Nurture (only set at end)

Build Path Tradeoff — full picture

Dimension	Atlas build (recommended)	Existing CRM via Riv
Time to MVP	~1 week (Atlas adds a table + kanban view)	~1–2 days (sign up, import, configure)
Cost	\$0 ongoing	\$15–\$50/mo per seat, escalates with features
Integration with PKA	Native — Echo / Vox / Atlas all share one DB	Another silo; needs syncs
Data ownership	Fully yours, local-first	Vendor-hosted
Babysit burden	Internal — Atlas already in active dev	Another SaaS dashboard to remember to open
Risk	Atlas already has pending work (GDrive watcher idle) — this adds to that queue	Vendor lock-in if you outgrow the free tier
Best fit if...	You want one PKA-wide system of record	You expect to scale past 100+ active prospects and want CRM features (sequences, email templates inside the CRM, mobile app)

Why Atlas wins for J2 specifically:

1. **Volume doesn't justify CRM features.** You're not running a 200-rep SDR org. You need a list with stages and dates.
2. **Echo already lives in your email.** If a CRM sits between you and Gmail, Echo's value drops. If the pipeline is in Atlas, Echo can mark emails as "prospect activity" and Atlas captures it natively.
3. **You've already invested in Atlas.** Adding a prospects table is incremental, not a new platform.
4. **You said you prefer shipping faster with more supervision burden** (per Larry's notes on you). Atlas-with-Riv-on-capture matches that pattern: one week to ship, you supervise the build, then it's yours.

Where I'd flip to CRM: if audit answer #3 is "50+ active prospects" or #6 is "I need sequences, drip campaigns, and email tracking baked in." Then we're not building a dashboard, we're buying a sales tool.

Owner Decisions Before Kickoff

These are the calls only you can make. I will not start the brief to Atlas until I have these.

1. **Confirm the build path** — Atlas (recommended) or CRM. I need a yes on Atlas before

assigning.

2. **Stage taxonomy** — accept the strawman above, edit it, or send me your version.
 3. **"Stale" thresholds** — how many days at each stage before it's flagged stale? (Default proposal: New=14d, Qualified=10d, Discovery=7d, Proposal=5d.)
 4. **Access** — just you, or do remote bookkeepers get read or write?
 5. **Capture mode** — manual entry only (you add prospects when they come in) OR semi-auto (Echo/Vox flag inbound, you confirm)? Semi-auto is more setup but matches the ADHD-friendly "externalized memory" pattern.
 6. **Mobile?** — do you need to update this on your phone, or is desktop fine?
-

Recurring vs. One-Time Work

Preempting the automation probe:

- **One-time build:** schema, dashboard views, stale-alert logic. Atlas + Riv ship this once.
 - **Recurring work that stays on your plate:** advancing prospects through stages, writing the next action, taking the discovery calls. This is sales judgment — not automatable, and shouldn't be.
 - **Recurring work the system handles automatically:** stale-flagging, next-action sort, source attribution rollup. Set-and-forget once shipped.
-

What Happens After You Answer

1. I write the delegation brief to **Atlas** (schema, views, stale-alert logic) and **Riv** (capture seam from Gmail/LinkedIn).
 2. Atlas builds the table + kanban + dashboard. Riv builds the inbound capture.
 3. You get a working MVP in roughly a week. We add nothing else until you've used it for two weeks and told me what's missing.
-
-

Next Action

Jimmie — answer the 7 audit questions and confirm the 6 owner decisions above. Reply in any format. Once I have them, briefs go to Atlas and Riv same day.