

Good. That's the only correct answer.

We're not "planning a refactor."

We're executing a **combat retrofit on a live bird in flight**.

Let's turn all the theory into **concrete work Melissa can start on TODAY**.

I'm going to break this into **four parallel tracks** you can literally paste into Linear/Jira/Notion as epics + stories.

---

## EPIC 1 — ARCHANGEL v2 (Recursive Enrichment Engine)

### 1.1 Add Proper Lead State Machine + Attempts Budget

**Goal:** Move from one-shot enrichment to bounded, recursive passes.

**Changes:**

- Extend LeadEvent (or equivalent) with:
  - enrichment\_attempts: int
  - max\_enrichment\_attempts: int (default 3)
  - enrichment\_status: enum
    - UNENRICHED
    - WITH\_DOMAIN\_NO\_EMAIL
    - ENRICHED\_NO\_OUTBOUND
    - OUTBOUND\_SENT
    - ARCHIVED\_UNENRICHABLE
  - Add unenrichable\_reason: str (e.g. "NO\_DOMAIN", "NO\_CONTACT\_INFO", "NO\_OSINT\_PRESENCE")

**Worker behavior:**

- On enrichment run:
  - If enrichment\_attempts  $\geq$  max\_enrichment\_attempts  $\rightarrow$  set ARCHIVED\_UNENRICHABLE + reason, stop.
  - Else:
    - enrichment\_attempts += 1
    - Execute one **pass** of NameStorm  $\rightarrow$  DomainStorm  $\rightarrow$  PhoneStorm  $\rightarrow$  EmailStorm using mission log (below).
      - If success (email found)  $\rightarrow$  ENRICHED\_NO\_OUTBOUND.
      - If partial success (domain only)  $\rightarrow$  WITH\_DOMAIN\_NO\_EMAIL.

- If no progress and attempts exhausted → ARCHIVED\_UNENRICHABLE.
- 

## 1.2 Mission Log (Per-Lead Enrichment History)

**Goal:** Stop re-doing the same dumb shit. Give us and future ML a black box recorder.

**Data model (per lead):**

enrichment\_mission\_log: JSON[] with entries like:

```
{  
  "timestamp": "...",  
  "pass": 2,  
  "phase": "DOMAINSTORM",  
  "action": "duckduckgo_search",  
  "query": "cool running air miami hvac",  
  "result": "no_domain_found",  
  "notes": "top result was directory listing without  
website"  
}
```

**Usage:**

- Each enrichment subroutine (NameStorm/DomainStorm/PhoneStorm/EmailStorm) writes structured entries.
  - Before doing something “expensive” (search, scrape), check log to avoid exact repeats.
- 

## 1.3 NameStorm v2 (Multi-Candidate Names)

**Goal:** We stop pretending there's only one “company name” per signal.

**Implementation:**

- Refactor NameStorm to output:

```
[  
  {"name": "Cool Running Air", "confidence": 0.92, "source":  
  "title_ner"},  
  {"name": "Cool Running HVAC Services", "confidence": 0.78,
```

```
"source": "body_ner"},  
  {"name": "Running Air LLC", "confidence": 0.41, "source":  
"heuristic"}  
]
```

- Store this array on the lead:
    - candidate\_company\_names: JSON[]
  - DomainStorm loops through candidates in **confidence order**, logging which ones have already been tried.
- 

## 1.4 DomainStorm v2 (Layered, Stateful)

**Goal:** Keep trying smarter, not just more.

**Layers (in order):**

1. **Existing domain fields**
  - From signal metadata, structured schema, or lead.source\_url.
2. **Direct article/HTML scrape**
  - Extract outbound links & schema.org Organization.url.
3. **Heuristic domain guessing**
  - Turn Cool Running Air → coolrunningair.com, .net, etc. (only top 2–3 best guesses).
4. **Search engine**
  - DuckDuckGo HTML query: "<company>" <city> hvac etc.
  - Enforce per-lead budget: max 3–5 queries per pass.
5. **Phone-assisted lookup**
  - If PhoneStorm has any number, search that number to find business listing → domain.
6. **Optional external API (future/hybrid)**

**Stateful behavior:**

- DomainStorm consults mission log to skip layers already run for this lead.
  - When a domain is found:
    - Write lead.domain
    - Move lead to WITH\_DOMAIN\_NO\_EMAIL
    - Append {"phase": "DOMAINSTORM", "result": "domain\_found", "domain": "..."} to mission log.
- 

## 1.5 PhoneStorm v2 (Discovery + Pivot)

**Goal:** Phone is not just display data; it's a pivot key.

### Changes:

- PhoneStorm now runs:
    - On any page we scrape.
    - After domain is known, crawl:
      - /
      - /contact
      - /about
      - /locations (if cheap to detect)
  - Validation/classification:
    - Normalize to E.164
    - Tag: MOBILE, LANDLINE, TOLL\_FREE, VOIP, UNKNOWN
  - Store on lead:
    - phones: [{ number, type, source\_url }]
  - In next pass:
    - DomainStorm uses any phones found to do search-by-phone if domain is still unknown or ambiguous.
- 

## 1.6 EmailStorm v1 (Layered Email Discovery)

**Goal:** Move from "maybe info@" to structured, layered resolution.

### Layers:

1. **On-site scrape**
  - Regex find emails on pages crawled.
2. **Generic pattern generation**
  - info@, contact@, sales@, support@ for the domain.
  - SMTP check for deliverability (non-invasive VRFY-style/thin handshake).
3. **Person-based (if we have a name)**
  - Use simple patterns first@, first.last@.
  - Only for high-confidence, named owners from content.
4. **External OSINT query**
  - Optional: Google-style "<company>" "@domain.com" scraping, within budgets.

### Output:

- emails: [{ email, type: "GENERIC"|"PERSONAL"|"SCRAPED", source\_url }]
  - If any deliverable email found → mark lead ENRICHED\_NO\_OUTBOUND.
-

## EPIC 2 — Company Intelligence Layer (The “Organ” We’re Missing)

### 2.1 Company Table (Canonical Entity)

**Goal:** Stop treating every lead as brand new.

#### New table: Company

- id
- name
- normalized\_name
- domain
- hq\_city, hq\_state, hq\_country
- phones[]
- emails[]
- source\_confidence
- first\_seen\_at, last\_seen\_at
- tags (e.g. ["HVAC", "Miami"])

#### Link:

- LeadEvent.company\_id (nullable).
- When enrichment succeeds:
  - Upsert company record.
  - Attach lead to it.
- When new lead appears:
  - First try to **match** existing Company by (normalized name + geo OR domain).

---

### 2.2 Reuse & Cache

#### Behavior:

- If company already exists and has domain/email:
  - Lead enrichment is trivial:
    - Copy domain/contacts → mark ENRICHED\_NO\_OUTBOUND.
    - Log “hit cache” in mission log.
  - This is how we build our own mini-ZoomInfo over time.

---

## EPIC 3 — SignalStorm v1 (New SMB-Heavy Signal Sources)

We don't need the whole planet yet.

We need **high-density, SMB-flavored, OSINT-rich signals**.

### 3.1 Craigslist Connector (MVP)

**Scope:** Start with 1–2 categories in 1–2 geos (e.g. Miami, Broward).

- Use RSS for:
  - services offered
  - jobs
- For each post:
  - Extract:
    - Title (often includes business name)
    - Body
    - Posted email/phone (if exposed)
    - Location
- Convert to Signal with:
  - source: "craigslist"
  - raw\_text
  - location
  - direct\_contacts: emails/phones if present

**Scoring heuristics:**

- If the post **has an email or phone** → score high (e.g. 90+).
  - If the title/body clearly contains a business name (NameStorm confidence high) → high score.
  - These can skip half the enrichment steps because contact is **already in the signal**.
- 

### 3.2 Job Board Connector (MVP)

Pick a single job source to start (e.g. Indeed or a local aggregator if easier).

- Query for certain regions + categories.
- Extract:
  - Company name
  - Location
  - Job title
  - apply\_url (often on the company's own site)
- Feed into SignalNet with:
  - source: "job\_board"
  - company\_name\_candidate
  - apply\_url (which may be a direct site URL = domain).

**Why:** Job posts are **expansion signals** and more structured than news.

---

### 3.3 Reddit Connector (Phase 2)

- Monitor a small set of subreddits:
  - r/smallbusiness
  - city subs: r/miami, r/fortlauderdale etc. (as test)
- Filter for posts mentioning:
  - starting a business
  - "just opened"
  - "recommend an HVAC company in Miami"
- Extract:
  - Business names (if any)
  - Context (need, complaint, recommendation)

Feed these into SignalNet with lower initial volume and stricter scoring until we see yield.

---

## EPIC 4 — Intelligence & UX (Make It Legible To Humans)

### 4.1 Global Metrics & Source Yield

Instrument:

- enriched\_leads / total\_leads (global)
- enriched\_leads / total\_leads per source:
  - news
  - craigslist
  - job\_board
  - reddit
- avg\_attempts\_per\_lead
- Count per unenrichable\_reason

Use this to:

- Kill or downrank low-yield sources.
- Tune scoring thresholds.
- Decide where to invest more engineering time.

---

### 4.2 Portal / Admin Updates

## Admin console:

- Show for each lead:
  - enrichment\_status
  - enrichment\_attempts
  - If ARCHIVED\_UNENRICHABLE: show reason string
  - A condensed view of mission log (last few attempts).

## Customer portal:

- Opportunities shown only when:
  - OUTBOUND\_SENT or
  - ENRICHED\_NO\_OUTBOUND waiting for review.
- For each card:
  - "Why this opportunity?" text
  - Source (e.g. "Found via Craigslist job post")
  - Company name + domain
  - Phone(s) with type badges
  - Email used for outbound

This makes the system feel **decisive**, not random.

---

## How To Move Right Now

If you want a pure **short list of "do this today"** you can DM Melissa, here it is:

1. **Add lead fields:** enrichment\_attempts, max\_enrichment\_attempts, enrichment\_status, unenrichable\_reason, enrichment\_mission\_log.
2. **Implement state machine** around enrichment worker: respect max\_enrichment\_attempts, write mission log, assign ARCHIVED\_UNENRICHABLE with reason.
3. **Refactor NameStorm** to output multiple candidate names + confidences and persist on lead.
4. **Refactor DomainStorm** into layered, stateful logic that checks mission log and writes rich log entries.
5. **Promote PhoneStorm** from "nice to have" to "core": store phones on lead; use them as search pivots.
6. **Implement EmailStorm** as layered email discovery with SMTP validation.
7. **Create Company table** and wire enrichment to upsert + attach leads to companies.
8. **Add simple Craigslist connector** in one market as a new SignalNet input.
9. **Add metrics collector** for enrichment success rate &

per-source yield.

10. Expose basic status & reason fields in admin + portal UI.

That is the **minimal viable structural fix**.

After that's in, SignalStorm + Strategic Intelligence (ForceCast/MacroStorm) can be layered on **without** fighting the same architecture demons.

---

You're right to drop Ron Popeil mode.

The engine in prod is not "immature"; it's **missing major organs**.  
We're grafting them in now.

If you want, next step I can draft:

- DB migration sketch
- Exact enum/state definitions
- Pseudo-code for `enrich_lead(lead_id)` with all the new logic wired in

But conceptually: yes, we move fast and we move **now**.

Yeah, man. You're talking about **the grown-up table now**.

All the **formal paperwork the big dogs are legally forced to generate**:

10-Ks, 10-Qs, 8-Ks, S-1s, earnings call transcripts, bankruptcy filings, new entity registrations, SEC stuff, state corp registries...

That's not "nice to have." That's the **macro weather system the SMBs have to live in**.

Let's bolt that onto what we already specced as a **new intelligence layer, not as a random bolt-on**.

I'll frame it as an addendum EPIC you can just hand to Melissa alongside the enrichment plan.

---

**EPIC 5 — MacroStorm / ForceCast**

(Strategic Intelligence Layer for Public Filings & Big-Co Moves)

## 5.0 What This Layer Is For

This is not about selling ZoomInfo-style contact records.

This is about:

- Watching the *public paperwork* of big players and:
  - Turning it into **structured "pressures" and "shocks"** (Porter / five-forces language),
  - Mapping those pressures to **downstream opportunities for small businesses**.

So:

- Bigco files a 10-K saying they're rolling out 45 new distribution centers → Local SMBs in **construction, logistics, staffing, HVAC, security, catering, uniforms, etc.** have opportunity.
- Bigco files Chapter 11 → Vendors, landlords, competitors, replacement vendors all have *pain* and *opportunity*.
- A PE fund files a Form D and buys up a string of clinics → Local independents just got a new, well-funded competitor, and the supply chain is about to change.

MacroStorm = “**Here's who's moving tectonic plates.**”

HossAgent = “**Here's how those tremors become leads for you.**”

---

### 5.1 Data Classes We Care About

We don't need *all* filings. We need the ones that map to real-world moves.

#### A. SEC & Public Company Filings (US to start)

- **10-K / 10-Q** – strategic direction, segment growth, capex plans, store rollout counts, “risks & competition.”
- **8-K** – material events:
  - acquisitions, restructurings, layoffs, divestitures, new projects, major contracts.
- **S-1 / F-1** – IPO filings: who's going public, what their model is, how they plan to grow.
- **Earnings call transcripts** – this is where the CEO says the quiet part out loud:
  - “We're pivoting into X,”
  - “We'll be exiting Y geography,”
  - “We're consolidating suppliers in Z category.”

## B. Corporate Vital Signs

- **Bankruptcy filings (Chapter 11, 7):**
  - Which chains / big customers are dying,
  - Who stands to lose: landlords, suppliers, franchisees.
- **M&A announcements / antitrust filings:**
  - Two majors merging → vendor consolidation → smaller competitors suddenly look more attractive to buyers & suppliers.
- **State business registries & new entity filings:**
  - New regional subsidiaries, logistics entities, REIT shells, etc., that reveal where large players are planting flags.

## C. Local & Sector Filing Equivalents

- For banks/insurers/utilities:
  - Regulator reports, call reports, rate filings.
- For healthcare:
  - Certificate-of-need filings, hospital / clinic expansion announcements, etc.

We do **not** need to boil the ocean on day one. This is a **source category** we add carefully:

- Start: SEC + one bankruptcy feed + 1–2 state corp registries.

---

## 5.2 How MacroStorm Plugs Into HossAgent

Architecturally, it's another SignalNet producer, but its outputs are different:

- News/Reddit/Craigslist signals → "company X is doing Y" (micro)
- MacroStorm filings → "sector S in region R is undergoing Z force" (macro)

We want both, feeding into the same downstream system.

New object: MacroEvent

Example structure:

{

```
"macro_event_id": "macro-2025-SEC-MCD-10K-ops-expansion",
"source_type": "SEC_10K",
"source_ref": "CIK-0000063908-2025-10K",
"company_name": "McDonald's Corporation",
"ticker": "MCD",
"headline": "Plans 500 new units, 120 in Florida over 3
years",
```

```

"geographies": ["Florida", "Miami-Dade", "Broward"],
"segments_affected": ["QSR", "logistics", "construction"],
"force_type": "EXPANSION",
"time_horizon": "1-3_years",
"risk_impact": {
    "local_competitors": "INCREASED_COMPETITION",
    "suppliers": "INCREASED_DEMAND",
    "labor_market": "TIGHTENING"
},
"raw_snippet": "...from MD&A / earnings call...",
"confidence": 0.86
}

```

Then we define mapping logic:

"Given macro event E and a user in sector S / region R, what **SMB-level opportunities** exist?"

That mapping is where HossAgent turns into a **tiny McKinsey that actually does something.**

---

### 5.3 How MacroStorm Generates Signals For SMBs

We add a Macro → Micro Edge in the pipeline:

1. **Ingest Filing / Transcript**
  - Use a scheduled job to:
  - Fetch new SEC filings (10-K, 10-Q, 8-K) for a watchlist of tickers / sectors.
  - Fetch new bankruptcy filings for known chains / categories.
  - (Later) fetch call transcripts from a provider or scraped source.
2. **Extract Strategic Moves**
  - For each document:
  - Run LLM/NLP classification to tag:
    - Expansion vs. contraction vs. restructuring.
    - New markets/geos.
    - Capital projects (data centers, warehouses, plants, clinics, stores).
    - Supplier strategy (consolidating vendors, increasing diversity, new requirements).
  - Extract:

- Named geographies ("South Florida", "Miami", "Broward").
  - Named partner types (contractors, 3PLs, clinics, etc.).
  - Timelines.
3. **Generate MacroEvents**
    - For each distinct move, create a MacroEvent (like above).
    - Persist them in a macro\_events table.
  4. **Macro→Micro Rules Engine**

We build a simple rule engine (phase 1 = if/else; phase 2 = more learned heuristics):

For a MacroEvent like:

Big REIT expanding 12 medical office buildings in Miami over 2 years.

We generate implied **SMB opportunity templates**:

- "Local HVAC & mechanical contractors in Miami"
- "Electrical contractors in Miami"
- "Cleaning & facilities vendors in Miami"
- "Furniture suppliers who serve clinics in Miami"

So MacroStorm outputs **target profiles**, not individuals:

```
{
  "macro_event_id": "...",
  "smb_opportunity_segments": [
    {
      "segment": "HVAC_CONTRACTORS",
      "geo": ["Miami-Dade", "Broward"],
      "urgency": "HIGH",
      "window": "0-24_months"
    },
    {
      "segment": "COMMERCIAL_CLEANING",
      "geo": ["Miami-Dade"],
      "urgency": "MEDIUM"
    }
  ]
}
```

**Then HossAgent uses existing machinery:**

- Look up local businesses matching that segment + geo from:
  - Google Maps / business registries / existing company cache.

- Generate **LeadEvents** for those SMBs:
  - "Target: local HVAC business"
  - Context: "MacroShock: Bigco building X in your backyard."
  - Enrich via ARCHANGEL (names, domains, contacts).

So MacroStorm → "**here's who to go after**"  
ARCHANGEL → "**here's how to reach them.**"

---

## 5.4 Types of Macro Plays We Support First

To keep this sane, we pick a few "plays" we know how to script.

### Play 1 — Big-Co Expansion In Your Backyard

Trigger sources:

- 10-K / 10-Q / 8-K, earnings call:
  - "We plan to open N new locations / facilities in region X."
- Public REIT filings about new builds.

SMB opportunity mapping:

- Local trades: HVAC, electrical, construction subs.
- Local staffing agencies.
- Local security firms.
- Local marketing/PR agencies.

Outbound flavor:

"I'm tracking expansion patterns in [your region]. [BigCo] just announced [X]. That's going to spike demand for [type of work you do] within [Y months]. Here's what I'm seeing and how you could position."

### Play 2 — Big-Co Retrenchment & Closures

Trigger sources:

- 8-K / press release:
  - "Closing stores / facilities in region X."
- Bankruptcy filings for big chains.

SMB opportunity mapping:

- Competitor SMBs → can capture displaced demand.
- Landlords / brokers → need replacement tenants.
- Liquidation & auction services.
- Specialty service providers who can "backfill" (e.g., local clinics when a

hospital exits a market).

Outbound flavor:

"A major player in your space is exiting [area]. That leaves a gap in [service]. Here's what's likely to happen and how you could take advantage of the opening."

## Play 3 — Supply Chain Consolidation

Trigger sources:

- 10-K MD&A, risk section:
  - "We are consolidating suppliers / renegotiating terms / centralizing procurement."
- 8-K:
  - Major sourcing / vendor changes.

SMB mapping:

- Existing suppliers who might get squeezed and need new customers.
- New entrants who can position as "nimble / local / diversified" suppliers.

Outbound flavor:

"Your upstream partner is consolidating suppliers. That kind of move usually means [pressure A, pressure B] for companies like yours.

Based on [filing snippet], here's what I expect and how you could hedge."

## Play 4 — Regulatory / Rate / Policy Shocks

Trigger sources:

- Regulator filings (for utilities, telecom, insurance).
- Public rate case filings; regulatory approvals of mergers.

SMB mapping:

- Service providers downstream (agents, brokers, local contractors, etc.) who will see margin compression or opportunity created by policy changes.

Outbound flavor:

"Regulators just approved [policy change] affecting [sector] in [region].

That impacts [pricing / demand] for businesses like yours.

Here's a quick breakdown and why now might be the moment to [move X]."

## For You (Operator / Product)

- New Admin views:
  - MacroEvents list:
  - Source (10-K, 8-K, bankruptcy)
  - Force type (EXPANSION / CONTRACTION / MERGER / SUPPLY\_CHAIN)
  - Geos, segments, time horizon
  - For each MacroEvent:
  - The **derived SMB target list** (which segments & geos we're generating leads for).
  - Counts:
    - Leads created from this macro event.
    - Enriched / contacted / replied.

## For End Users

### In their Portal:

- Some opportunities are "**Micro**":
  - "We saw an article about YOUR company. Here's outreach to a potential customer/partner."
- Some are "**Macro-aware**":
  - "We're seeing [BigCo] expand/contract/merge in your backyard. Here's a list of local businesses or potential clients where this creates an angle."

Card might show:

- **Macro header:**

"Macro Signal: [Bigco] is doing [X] in [Region]."

- **Opportunity card:**

- Target SMB: "Cool Running Air, Miami"
- Why now: "They will be directly affected by [MacroEvent]."
- Email / phone and suggested outreach written in that "unsolicited McKinsey" voice.

---

## 5.6 Constraints & Guardrails

### We should be explicit about what we're not doing:

- Not building a full SEC terminal; we're mining filings to generate **structured MacroEvents**.
- Not selling proprietary investor data; we're using **public filings and transcripts**.
- Not trying to map every filing; we pick **high-yield patterns** (expansion,

closure, restructuring, capex).

- Not letting MacroStorm run wild; we:
    - Start with a small **ticker/sector watchlist**.
    - Start with a **few plays** that map clearly to SMB opportunities.
    - Watch enrichment + outbound performance.
    - Expand incrementally.
- 

## 5.7 Concrete Work Items Melissa Can Actually Build

You can hand her this part almost verbatim:

1. **Add MacroEvent model & tables**
  - macro\_events
  - Fields like: source\_type, source\_ref, company\_name, tickers, geographies, force\_type, segments\_affected, time\_horizon, raw\_snippet, confidence.
2. **Build SEC Ingestion Job**
  - Poll EDGAR for new 10-K, 10-Q, 8-K for a short watchlist (e.g., top 50 tickers in sectors relevant to your users).
  - Store raw filings in S3 / DB for LLM processing.
3. **Add LLM/NLP extraction pipeline**
  - Given a filing:
  - Extract mention of expansion / closure / restructuring.
  - Extract numbers (e.g., new units, capex, geos).
  - Generate 1-N MacroEvents with structured fields.
4. **Implement Macro→Micro rules engine (v1)**
  - Hard-code a few mapping rules:
  - "If MacroEvent.force\_type=EXPANSION and segment includes 'RETAIL' in [Miami/Broward] → target: local construction, HVAC, cleaning, marketing."
  - "If MacroEvent.force\_type=CONTRACTION/BANKRUPTCY → target: competitors, landlords, auctioneers, etc."
  - Turn these into LeadEvents pointing at SMB segments + geos.
5. **Hook into existing enrichment**
  - For each Macro-generated LeadEvent:
  - Use same ARCHANGEL v2 pipeline (NameStorm/DomainStorm/PhoneStorm/EmailStorm).
6. **Display Macro context in portal**
  - When a lead originated from a MacroEvent, show:
  - Macro title ("BigCo expansion in South Florida").
  - One-sentence "So what for you."
7. **Track metrics**
  - Leads generated per MacroEvent.

- Enrichment success per MacroEvent type.
  - (Eventually) reply / conversion per MacroEvent type.
- 

## TL;DR in Hoss Language

- ARCHANGEL v2 = **the bounty hunter**.
- SignalStorm = **the scouts in the brush**.
- MacroStorm / ForceCast = **the satellite picture and intel brief from division HQ**.

You're not just telling SMBs "**we found an email.**"

You're telling them:

"Here's what's about to happen around you,  
here's who's going to feel it,  
and here's the move you should make —  
and, by the way, we already drafted the first phone call / email for you."