AI Emailing Structure

Dual-engine outbound & inbound AI automation infrastructure — a zero-middleware execution stack that runs on raw Google infrastructure, direct REST API calls, and LLM compute.

1 · Core Philosophy

We eliminate third-party SaaS middleware (Zapier, Make, Instantly, Lemlist) and run the entire pipeline on native infrastructure and LLM compute. The result: complete data control, zero monthly middleware bloat, and near-zero marginal cost — roughly **$0.20 per 1,000 custom messages generated**.

[Outbound Pipeline] → [Google Sheet Database] → [Apps Script Engine] → [Outbound Send] ↓ [Inbound AI Responder] ← [LLM Core / JSON Engine] ← [Listener Engine] ← ← ← ← ←

2 · Outbound Engine — Cold Outreach

Method A · Native Google Workspace

- **Database:** Google Sheets (SpreadsheetApp)
- **Engine:** Apps Script bound to the campaign sheet
- **API:** `UrlFetchApp` to the LLM endpoint, keys stored in ScriptProperties
- **Delivery:** GmailApp.sendEmail() / createDraft()

A **generateAndSendOutreach()** loop reads rows where **Status = Pending**, builds the JSON payload from the system prompt + lead variables, parses the model response into the Subject/Body columns, dispatches, then timestamps the row as Drafted or Sent.

Method B · Any-Email Routing

- For external domains and non-Google infrastructure, delivery decouples from GmailApp.
- **Engine:** Sheets + Apps Script (unified visibility retained)
- **Delivery:** UrlFetchApp → external SMTP / mail APIs (SendGrid, Mailgun, custom endpoints)
- Credentials held in PropertiesService; a **sendViaExternalSMTP()** helper carries From / Reply-To headers.

Limits: standard Workspace accounts allow ~1,500–2,000 dispatches/day via Apps Script, processed in batches on time-driven triggers.

3 · Inbound Engine — AI Auto-Responder

The responder listens for inbound replies, passes message content to the LLM, evaluates it against business rules, and drafts or dispatches a reply with full thread context.

Method A · Gmail Native Stack

- Gmail → GCP Pub/Sub → webhook → fetch email → AI → send via Gmail API
- users.watch() triggers push notifications (renewed daily; the watch expires every 7 days)
- Pub/Sub delivers emailAddress + historyId; the webhook pulls the exact message via users.history.list()
- Loop prevention ignores our own address and auto-reply headers; the last 2–3 thread messages are assembled for context
- Replies sent with users.messages.send() carrying threadId + In-Reply-To / References for clean threading

Method B · Universal (Any Provider)

- Yahoo, Outlook, Zoho, cPanel, Namecheap and standard domain mail servers
- **Option 1 · Direct IMAP/SMTP:** a lightweight worker using IMAP IDLE for instant, poll-free arrival alerts; replies via SMTP (587/465)
- **Option 2 · Unified API layer** `preferred` — normalize every inbox into standard webhooks (Nylas, Unipile, Postmark/SendGrid inbound) forwarded to our listener

4 · LLM Core — Decision Prompt

The inbound listener classifies every message and returns strict JSON, so routing logic stays deterministic:

```
You are an automated executive email assistant.
Parse incoming emails and draft or send responses from the provided context.

RULES
1. If the email is spam, marketing, or an automated alert → {"action":"IGNORE"}
2. Otherwise, analyze the context and thread history
3. Keep tone concise, professional, and clear
4. Output JSON strictly:
   { "action":"REPLY"|"IGNORE"|"ESCALATE",
```

```
    "reasoning":"brief explanation",
    "reply_body":"clean text of response" }
```

5 · Safeguards

Draft-mode first. Every new campaign defaults to writing drafts (or a review column) before any direct-send function is enabled. No third-party middleware sits in the routing path, and launching a new campaign requires only duplicating the master sheet, updating the system prompt, and pasting the new lead list.