IThink of the LLM as a **“decision engine”** that, for every email/thread, must answer four big questions:

Every 5 minutes, server hits email, feeds into a database, when a new record is triggered and adds to the database, endpoint connected to this model and returns back to another endpoint setup by Omar

He pings me, I send it back

1. **What *is* this email?** (classifications [task to complete, informational])
2. **What should the user do about it?** (tasks & actions [upcoming meeting potential schedules, submissions, projects, etc.])
3. **How important/urgent is it?** (prioritization)
4. **How should Drakko treat it in the UI?** (filtering + surfacing)

Below is what I’d hand your AI co-founder as the **LLM contract** for v1.

---

1. Core email/thread classifications

These are labels the LLM must output **for each email (or thread)**.

1.1 Actionability

- `is_actionable`: `true | false`
- `actionability_reason`: short natural language reason.
- `required_actions_count`: integer (0, 1, >1)
- `actionability_type` (pick one):
  - `no_action` — pure FYI, notification, or spam.
  - `reply_required`
  - `follow_up_required` (they owe us something later)
  - `decision_required` (approve/choose)
  - `task_required` (do some work, not just reply)
  - `mixed` (multiple of the above in one email)
- `waiting_status`:
  - `waiting_on_me` — they asked *you* to do something.

    - `waiting_on_them` — you already replied and are waiting.
    - `mutual` — both sides have open actions.
    - `none` — no ongoing back-and-forth dependency.

These directly drive:

- `is_actionable` → `email_messages.is_actionable`
- `waiting_on_me/ them` → which list a task goes into (“Today” vs “Waiting on”).

1.2 Intent / action type classification

This should map closely to your `task_action_type_enum`. For each email, the LLM should output:

- `primary_intent`: one of:
    - `reply`
    - `follow_up`
    - `review`
    - `approve`
    - `schedule_meeting`
    - `add_calendar_block`
    - `pay_invoice`
    - `create_document`
    - `update_document`
    - `create_note`
    - `inbox_cleanup`

  - `unsubscribe`
  - `general`
  - `other`
- `secondary_intents`: array of the same enum (there can be multiple actions in one email).

Example:

- "Can you review the doc and then send it to the client by Friday?"
  - `primary_intent = "review"`
  - `secondary_intents = ["reply", "follow_up"]`

These become the `action_type` for the tasks that are generated.

1.3 Content category (semantic category)

A higher-level "what is this email about?" layer. Good examples:

- `category` (one main):
  - `work`
  - `personal`
  - `finance`
  - `relationship`
  - `admin`
  - `knowledge`
  - `health`
  - `misc`

- `topic_tags` (array):
  - Examples:
    - `billing`, `invoice`, `payments`
    - `recruiting`, `candidate`, `interview`
    - `product`, `roadmap`, `bug`, `incident`
    - `legal`, `contract`, `NDA`
    - `marketing`, `newsletter`, `promotion`
    - `support`, `ticket`, `downtime`

These map to `tasks.category` and help later for filtering, analytics, and “smart lists”.

1.4 Priority / urgency

The LLM should extract both explicit and implicit urgency:

- `priority`:
  - `low`
  - `normal`
  - `high`
  - `urgent`
- `urgency` object:
  - `urgency_level`:
    - `none`
    - `time_sensitive`
    - `deadline_today`

        - `deadline_this_week`
        - `overdue`
    - `deadline_text`: the natural language phrase ("by Friday", "within 24 hours")
    - `parsed_deadline`: ISO timestamp if it can be parsed (`2025-12-10T23:59:59Z`)

These map directly to:

- `tasks.priority`
- `tasks.due_at`
- `tasks.scheduled_for` (if it's clearly "today/tomorrow/next week").

1.5 Relationship / counterpart info

Who is on the other side of this email?

- `counterpart`:
    - `name`: `"Jane Doe"`
    - `email`: `"jane@acme.com"`
    - `organization`: `"ACME Corp"`
    - `role_guess`: `"customer" | "vendor" | "manager" | "teammate" | "candidate" | "recruiter" | "friend" | "family" | "unknown"`

This is used to:

- Label "Waiting on ACME · VP Sales"
- Show counterpart context in `WaitingOn` lists
- Potential future: per-contact dashboards.

1.6 Email type / noise classification

To support filtering:

- `email_type`:
  - `human_direct` — someone personally wrote to you.
  - `human_bulk` — mass send but from real person.
  - `newsletter` — typical mailing list content.
  - `marketing` — sales drip campaigns, promos.
  - `product_update` — feature announcements.
  - `transactional` — receipts, invoices, system notices.
  - `notification` — GitHub, Jira, Slack, etc.
  - `spam`
  - `other`
- `is_noise`: `boolean` — "safe to auto-collapse/hide from Today?"
- `noise_reason`: short explanation.

This doesn't necessarily mean "unimportant" (e.g., invoices are transactional but high value), but it helps you hide routine stuff from the "top of Today".

---

2. Task & action extraction requirements

For each email/thread, the LLM needs to output **0–N tasks**.

2.1 Per-task core fields

For each identified task, we want:

- `task_id_hint` (optional but very useful):
  A stable string derived from the thread + content (e.g. `hash(thread_id + sentence)`), to help with idempotency (if we re-process the thread, we update the same task).

- `action_type`: one of the `task_action_type_enum`.
- `title`: short, actionable, self-contained:
  - `"Reply with pricing options for ACME"`
  - `"Approve Q1 headcount plan"`
  - `"Schedule onboarding call with Orbit Labs"`
- `instruction`: 1–3 sentences describing exactly what to do:
  - “Review the attached Q1 headcount plan and reply to People Ops with either an approval or requested changes.”
- `priority`: (`low | normal | high | urgent`)
- `category`: (`work | personal | finance | ...`)
- `owner`: for now always `"user"` (Drakko’s user), but later you might support assigning internally.
- `suggested_due_at`: ISO timestamp or `null`
- `suggested_scheduled_for`: date string (`YYYY-MM-DD`) if appropriate.
- `is_today_focus_candidate`: boolean — should this be pre-flagged as a Today focus candidate?

These map directly to `tasks`:

- `action_type` → `tasks.action_type`
- `title` → `tasks.title`
- `instruction` → `tasks.description`
- `priority` → `tasks.priority`
- `category` → `tasks.category`
- `suggested_due_at` → `tasks.due_at`
- `suggested_scheduled_for` → `tasks.scheduled_for`

- `is_today_focus_candidate` → `tasks.is_today_focus` (or used in heuristic).

2.2 Additional metadata per action type

Some tasks need extra structured info in `tasks.metadata`.

a) Reply / follow_up

- `reply_required`: boolean
- `thread_reference`:
  - `provider_thread_id`
  - `provider_message_id`
- `to`: array of email addresses
- `cc`: array
- `subject`: suggested reply subject (if needed)
- `reply_style_hint`: `"short_ack" | "detailed" | "apology" | "proposal" | "question"`
- `suggested_reply_draft`: (optional, can be produced by a *second* LLM call)

Mapped to `metadata`:

```
{
  "email_message_id": "<uuid in email_messages>",
  "provider_thread_id": "xyz",
  "provider_message_id": "abc",
  "to": ["jane@acme.com"],
  "cc": [],
  "subject": "Re: Q1 headcount plan",
  "reply_style_hint": "detailed"
}
```

b) Schedule meeting

- `participants`: list of `{ name, email }`
- `duration_minutes`: 15 / 30 / 45 / 60
- `time_constraints`:
  - `preferred_days`: e.g., `["Mon", "Tue"]`
  - `preferred_time_ranges`: e.g., `["09:00-11:00", "14:00-17:00"]`
  - `deadline_text` / `parsed_deadline` if meeting must happen before a certain date.
- `location_hint`: `"Zoom" | "Google Meet" | "In person"` (and maybe address)

c) Review / approve

- `artifact_type`: `"document" | "spreadsheet" | "slide_deck" | "PRD" | "contract"`
- `artifact_link`: URL if present
- `review_depth`: `"skim" | "detailed_review" | "sign_off"`
- `decision_expected`: `"approve" | "give_feedback" | "choose_option"`

d) Pay invoice / finance actions

- `invoice_id` (if present)
- `amount`
- `currency`
- `vendor_name`
- `due_date_text` + `parsed_due_date`

e) Create document / note

- `document_title_suggestion`

- `document_type`: `"PRD" | "meeting_notes" | "spec" | "summary"`
- `outline` or key bullet points

f) Inbox cleanup / unsubscribe

- `cleanup_type`:
  - `"unsubscribe_sender"`
  - `"archive_old_thread"`
  - `"mute_notifications"`
- `sender_email`
- `list_unsubscribe_link` (if extractable)
- `suggested_rule`:
  e.g., “Archive future emails with subject containing ‘Daily digest’.”

—

All of this is **optional and sparse**; the LLM only fills what’s relevant. Everything goes into `tasks.metadata`.

---

3. Filtering & surfacing decisions

Beyond tasks, the LLM should tell Drakko *how to treat this email* in the product.

3.1 Filtering

- `is_noise`: boolean
- `noise_type`: `"newsletter" | "promotion" | "notification" | "spam" | "low_priority_update"`
- `should_suppress_from_today`: boolean — “don’t show this in top Today views.”
- `suppress_reason`: short reason.

These feed:

- `email_messages.is_actionable`
- `email_messages.processing_status = "ignored"` (for low-value stuff)
- UI logic to decide what appears in Today vs a "low priority" bucket.

3.2 Surfacing / focus hints

- `surface_level`:
  - `top_focus` — user should see this at the top of Today.
  - `secondary` — important but not top-tier.
  - `background` — keep for context, not attention-grabbing.
- `pin_candidate`: boolean — should this thread/task be pinned?
- `group_with_thread`: boolean — whether this email should be grouped with previous emails into a single combined task.

These can influence:

- `tasks.is_pinned`
- Ordering in Today / Insights.

---

4. Thread-level understanding

Processing the entire **thread**, not just a single message, the LLM should also provide:

- `thread_summary`: 2–4 sentence summary.
- `unread_summary`: "Since you last read, they asked for X and confirmed Y."
- `open_questions`: array of questions the other side is waiting on.
- `commitments_i_made`: promises **you** made in the thread.

- `things_others_owe_me`: what you're waiting for from them (drives "Waiting on").

Example structure:

```
{
  "thread_summary": "ACME is negotiating a renewal for Q1. They asked
for updated pricing and a revised contract.",
  "open_questions": [
    "Can you confirm the final discount percentage?",
    "Will the new pricing apply retroactively to Q4?"
  ],
  "commitments_i_made": [
    "You promised to send a revised pricing sheet by Wednesday."
  ],
  "things_others_owe_me": [
    "ACME needs to send their final approval after reviewing the
contract."
  ]
}
```

From this, Drakko can derive one or more follow-up tasks automatically.

---

5. Putting it together: an example LLM output object

Per *email* + *thread*, I'd ask the LLM to return something like:

```
{
  "email_id": "provider-msg-id-123",
  "thread_id": "provider-thread-id-999",

  "actionability": {
    "is_actionable": true,
    "actionability_type": "reply_required",
    "required_actions_count": 2,
    "waiting_status": "waiting_on_me",
    "reason": "The sender is asking you to confirm pricing and
schedule a call."
  },

  "classification": {
    "primary_intent": "reply",
```

```
    "secondary_intents": ["schedule_meeting"],
    "category": "work",
    "topic_tags": ["sales", "pricing", "renewal"],
    "priority": "high",
    "urgency": {
      "urgency_level": "deadline_this_week",
      "deadline_text": "by Friday",
      "parsed_deadline": "2025-12-12T23:59:59Z"
    },
    "email_type": "human_direct"
  },

  "counterpart": {
    "name": "Jane Doe",
    "email": "jane@acme.com",
    "organization": "ACME Corp",
    "role_guess": "customer"
  },

  "tasks": [
    {
      "task_id_hint": "reply_acme_pricing_2025-12-09",
      "action_type": "reply",
      "title": "Reply to ACME with updated pricing",
      "instruction": "Review the proposed pricing tiers, finalize the discount, and reply to Jane at ACME with the updated pricing sheet.",
      "priority": "high",
      "category": "work",
      "suggested_due_at": "2025-12-10T17:00:00Z",
      "suggested_scheduled_for": "2025-12-10",
      "is_today_focus_candidate": true,
      "metadata": {
        "provider_thread_id": "provider-thread-id-999",
        "provider_message_id": "provider-msg-id-123",
        "to": ["jane@acme.com"],
        "cc": [],
        "subject": "Re: ACME renewal pricing",
        "reply_style_hint": "detailed"
      }
    },
    {
      "task_id_hint": "schedule_call_acme_2025-12-09",
      "action_type": "schedule_meeting",
      "title": "Schedule renewal call with ACME",
```

```
      "instruction": "Find a 30-minute slot this week and schedule a
renewal call with Jane from ACME.",
      "priority": "normal",
      "category": "work",
      "suggested_due_at": "2025-12-13T23:59:59Z",
      "suggested_scheduled_for": "2025-12-11",
      "is_today_focus_candidate": false,
      "metadata": {
        "participants": [
          { "name": "Jane Doe", "email": "jane@acme.com" }
        ],
        "duration_minutes": 30,
        "location_hint": "Zoom"
      }
    }
  ],

  "filtering": {
    "is_noise": false,
    "noise_type": null,
    "should_suppress_from_today": false,
    "surface_level": "top_focus",
    "pin_candidate": true
  },

  "thread_view": {
    "thread_summary": "ACME is negotiating their Q1 renewal and needs
updated pricing and a call to review details.",
    "unread_summary": "Jane asked you to confirm the final pricing and
schedule a call this week.",
    "open_questions": [
      "Can you confirm the final pricing?"
    ],
    "commitments_i_made": [
      "You said you would send revised pricing by Wednesday."
    ],
    "things_others_owe_me": []
  }
}
```

Your backend would then:

- Turn each entry in `tasks` into a row in the tasks table.

- Mark `email_messages.is_actionable = is_actionable`.
- Fill `tasks.metadata` with the LLM’s `metadata` object.
- Use `classification` and `filtering` fields to drive dashboard KPIs and views.

TASK

Nice, this is exactly the kind of “LLM contract” question that makes Drakko powerful.

Below is a **taxonomy of main email action classifications** – the “verbs” your LLM should be able to attach to an email or thread.

Think of these as **canonical action types**. In practice you’ll map many real-world phrasings (“circle back”, “ping them”, “send deck”) into one of these.

I’ll group them by category and give a short explanation per.

Also Need/Have: Package that reads PDF documents, receive data from OMAR as a HTML data with tags

The AI server is different from the NODE server. Build locally and deploy to Github. AI itself will have 2 separate servers, one for ChatGPT and one for a personally trained model. There, we can get it up and deployed in my own space.

Server has an endpoint taking in HTML and providing responses as JSON objects according to below and [reference to above first {1.} actionable tab]

Pull out keywords from the following: How are you planning to use Drakko?

---

1. Core communication actions

1. **REPLY**
   - **What it is:** You need to write a response to this email.
   - **Examples:** Questions from a client, requests for info, “Can you confirm…?”
   - **Why it matters:** Generates a “Reply to X” task and can power in-app reply drafting.
2. **FOLLOW_UP**
   - **What it is:** You previously replied or acted; now you need to check back / nudge them.

- **Examples:** “If I don’t hear back, let’s revisit next week”, “Can you follow up on this?”
- **Why it matters:** Drives your “Waiting on” list and follow-up reminders.

3. **FORWARD / SHARE**
   - **What it is:** You need to pass this email or content to someone else.
   - **Examples:** “Can you share this with Legal?”, “Loop in the team on this update.”
   - **Why it matters:** Might create a task like “Forward to Finance and add a note”.
4. **ACKNOWLEDGE / CONFIRM RECEIPT**
   - **What it is:** Just confirm you saw it; content doesn’t need deep work.
   - **Examples:** “Please confirm you received this”, “A quick ‘got it’ is fine.”
   - **Why it matters:** Often low-effort but time-sensitive; quick-win tasks.

---

2. Decisions & approvals

5. **APPROVE / REJECT**
   - **What it is:** You have to formally approve or decline something.
   - **Examples:** Contracts, budgets, proposals, designs, PRs.
   - **Why it matters:** High-impact; often blocking others; can be a top priority task.
6. **CHOOSE_OPTION**
   - **What it is:** Decide between given choices.
   - **Examples:** Pick time for a meeting, choose pricing tier, pick vendor A vs B.
   - **Why it matters:** A lot of emails are “pick one of these”; it’s a simple but critical action.

7. **PRIORITIZE / RANK**
    - **What it is:** Decide order or priority for items.
    - **Examples:** “Can you prioritize these 5 features?”, “Rank these candidates.”
    - **Why it matters:** Leads to tasks affecting roadmap, hiring, etc.

---

3. Scheduling & time management

8. **SCHEDULE_MEETING**
    - **What it is:** Book a meeting based on the email.
    - **Examples:** “Let’s find 30 minutes next week”, “Can we meet Tuesday afternoon?”
    - **Why it matters:** Produces a task that might later create a calendar event.
9. **RESCHEDULE_MEETING**
    - **What it is:** Update an existing meeting’s time/date.
    - **Examples:** “Can we move our call to Thursday?”, “Something came up, can we reschedule?”
    - **Why it matters:** Often time-sensitive; may impact multiple people.
10. **CONFIRM_MEETING**
    - **What it is:** Confirm an already proposed or tentative time.
    - **Examples:** “Does 3pm tomorrow work?”, “If that time is fine, we’ll lock it.”
    - **Why it matters:** Quick action, but delays slow everything down.
11. **ADD_CALENDAR_BLOCK / FOCUS_TIME**
    - **What it is:** Add a block of time to work on something.

- **Examples:** “Please spend an hour reviewing this by Friday”, “We’ll need half a day to debug.”
- **Why it matters:** Helps Drakko recommend blocking time.

12. **ATTEND / RSVP**
    - **What it is:** Accept or decline an event, invite, or webinar.
    - **Examples:** “Register for this event”, Accept invites.
    - **Why it matters:** Generates simple tasks: “RSVP yes/no”.

---

4. Document & content work

13. **REVIEW_DOCUMENT**
    - **What it is:** Read and review a document or attachment.
    - **Examples:** PRD, spec, contract, deck, report.
    - **Why it matters:** Heavy cognitive work; should show as “deep work” tasks.
14. **EDIT / COMMENT_DOCUMENT**
    - **What it is:** Make edits or leave comments.
    - **Examples:** “Can you leave suggestions?”, “Give feedback in the doc.”
    - **Why it matters:** Different from just reading; more effort, usually more urgent.
15. **DRAFT_DOCUMENT**
    - **What it is:** Create a new document from scratch.
    - **Examples:** “Draft a proposal”, “Write a post-mortem”, “Create a PRD.”
    - **Why it matters:** Big tasks that might spawn subtasks or focused time blocks.
16. **UPDATE_DOCUMENT**

- **What it is:** Update an existing doc with new info.
- **Examples:** “Can you update the roadmap with Q2 items?”, “Add those notes.”
- **Why it matters:** Common recurring action for product/leaders.

17. **SUMMARIZE / SYNTHESIZE**
    - **What it is:** Create a summary or synthesis of info.
    - **Examples:** “Send me a summary of that thread,” “Summarize user feedback.”
    - **Why it matters:** Perfect for AI assistance; yields tasks like “Summarize and send to team.”
18. **CREATE_NOTE / LOG**
    - **What it is:** Capture notes or log info somewhere.
    - **Examples:** “Log this in the CRM”, “Take notes from the call and share.”
    - **Why it matters:** Connects email to other systems (CRM, Notion, etc.).

---

5. Financial & admin actions

19. **PAY_INVOICE**
    - **What it is:** Pay a bill or invoice.
    - **Examples:** “Invoice attached, due by X”, “Please process payment.”
    - **Why it matters:** High-stakes, often urgent; ties to finance workflows.
20. **APPROVE_EXPENSE / REIMBURSE**
    - **What it is:** Approve reimbursement or expenses.
    - **Examples:** “Can you approve this expense report?”
    - **Why it matters:** Manager workflows.

21. **SUBMIT_EXPENSE**
    - **What it is:** Submit your own reimbursement.
    - **Examples:** “Attach receipts and submit in Expensify.”
    - **Why it matters:** Personal admin; recurring tasks.
22. **UPDATE_BILLING_DETAILS**
    - **What it is:** Update payment info, address, or subscription details.
    - **Examples:** “Your card is expiring”, “Update billing address.”
    - **Why it matters:** Avoids service interruptions.
23. **RENEW / CANCEL_SUBSCRIPTION**
    - **What it is:** Decide to continue or cancel a service.
    - **Examples:** “Your trial ends soon”, “Last chance to renew.”
    - **Why it matters:** Strategic and financial impact.

---

6. People & process actions

24. **RECRUITING_ACTION**
    - **What it is:** Do something related to hiring.
    - **Examples:** Schedule candidate interview, send take-home, review candidate, give feedback.
    - **Why it matters:** Feeds a “hiring” lane; might map to more granular actions.
25. **ASSIGN / DELEGATE**
    - **What it is:** Assign this request to someone else.
    - **Examples:** “Can you have your team handle this?”, “Please delegate to X.”

- **Why it matters:** Drakko could support internal assignment later.

26. **CREATE_TICKET / ISSUE**
    - **What it is:** Turn the email into a ticket (Jira, Linear, Zendesk, etc.).
    - **Examples:** Bug reports, support requests, feature requests.
    - **Why it matters:** Common workflow at teams; can be used to drive integrations.
27. **UPDATE_STATUS / REPORT_PROGRESS**
    - **What it is:** Provide an update on ongoing work.
    - **Examples:** “Can you share where we are on this?”, “What’s the status?”
    - **Why it matters:** Leads to tasks like “Send status update to X”.
28. **ESCALATE**
    - **What it is:** Raise priority or involve higher-level people.
    - **Examples:** “Escalate this issue to management.”
    - **Why it matters:** Marks tasks as urgent and may tag the thread specially.

---

7. Information management & organization

29. **SAVE_REFERENCE / FILE**
    - **What it is:** Store info for later reference.
    - **Examples:** Important credentials, instructions, legal notifications.
    - **Why it matters:** Might become “File important info into knowledge base / folder”.
30. **TAG / LABEL**
    - **What it is:** Apply labels/tags/categories.

- **Examples:** “Label all invoices from X as ‘Invoices’.”
- **Why it matters:** Drives organization, can be automated later.

31. **INBOX_CLEANUP**
    - **What it is:** General tidying: archive, move, mute threads.
    - **Examples:** “Archive this long-resolved thread,” “Clean up old notifications.”
    - **Why it matters:** Good for “batch tasks” like “Clean up GitHub notifications.”
32. **UNSUBSCRIBE**
    - **What it is:** Stop receiving a certain sender or newsletter.
    - **Examples:** Marketing blasts, newsletters that are no longer useful.
    - **Why it matters:** Classic high-value little action Drakko can suggest.
33. **SET_RULE / AUTOMATION**
    - **What it is:** Create a rule for future emails.
    - **Examples:** “Always archive this type of email”, “Auto-label as Finance.”
    - **Why it matters:** Perfect for future “automations” tab.

---

8. Learning, reading & awareness

34. **READ / REVIEW_INFO**
    - **What it is:** You should read this information, but no response is required.
    - **Examples:** Weekly updates, strategy docs, announcements.
    - **Why it matters:** Creates “reading tasks” that are not reply-based.
35. **RESEARCH / INVESTIGATE**

- **What it is:** Look into something before responding/deciding.
- **Examples:** “Can you look into why this is happening?”, “Investigate this bug.”
- **Why it matters:** More than just reading – requires further work.

36. **LEARN / TRAINING**
    - **What it is:** Training, courses, onboarding material.
    - **Examples:** “Complete this training by Friday.”
    - **Why it matters:** Often de-prioritized; Drakko can surface it intentionally.

---

9. Social & personal actions

37. **THANK / APPRECIATE**
    - **What it is:** Send a thank you / kudos / congrats.
    - **Examples:** “Please thank them for the introduction”, “Congrats emails.”
    - **Why it matters:** Keeps relationships healthy; easy quick tasks.
38. **SEND_INTRO / CONNECT_PEOPLE**
    - **What it is:** Introduce people to each other.
    - **Examples:** “Can you intro me to X?”, “Happy to connect them.”
    - **Why it matters:** Relationship-building actions.
39. **PERSONAL_LIFE_TASK**
    - **What it is:** Personal or family task triggered by an email.
    - **Examples:** Travel plans, family events, doctor appointments.
    - **Why it matters:** Helps Drakko be “life OS”, not just work.

---

10. Meta / system actions

40. **CONFIGURE_SETTINGS**
    - **What it is:** Change settings in another system.
    - **Examples:** “Update your Slack notification settings”, “Change password.”
    - **Why it matters:** Security & productivity; may spawn external links.
41. **SECURITY_ACTION**
    - **What it is:** Security-related response.
    - **Examples:** Password reset, suspicious login, phishing warnings.
    - **Why it matters:** High priority; may override normal filters.
42. **NO_ACTION / IGNORE**
    - **What it is:** Explicitly do nothing.
    - **Examples:** Pure FYI, noise, spam.
    - **Why it matters:** Let the LLM mark things as safely ignorable.