

Technical Report — “Chief AI” Discord Orchestrator (OpenClaw + Discord + Enterprise Dashboard)

Team:

Simon Karan, Hannes Vogel, Max Lehl, Ivan Naranjo

1) Overview

We built a professional AI orchestrator (“Chief AI”) inside a Discord server using OpenClaw. The system continuously monitors channels, extracts actionable items (including from meeting transcripts), routes work to specialized agents, and triggers operational follow-ups (DMs, tasks, and email summaries). The design emphasizes reliability, traceability, and explicit rules for external actions. **Primary goals**

- Convert chat and transcripts into structured tasks and actions
- Route work to the correct specialist agent
- Notify responsible people directly (Discord DM)
- Send emails when required (AgentMail)

2) Architecture

2.1 Core Components

- **OpenClaw Gateway** — runtime, routing, tool access, cron scheduling
- **Discord plugin** — message ingestion and output
- **Supervisor Agent** — main coordinator; scans channels and transcripts
- **Specialist Agents**
 - **Task-Manager** — creates tasks via the hosted API
 - **Historian** — summarization and memory
 - **Critic** — quality checks and contradictions
 - **Graph-Builder** — builds org/knowledge graphs
 - **Org-Graph-Manager** — maintains org directory and comms graph
-

2.2 Key Files

- `SUPERVISOR.md` — rules for message scanning, transcript parsing, routing
- `TASK-MANAGER.md` — API schema and task creation rules
- `ROUTING_POLICY.md` + `ROUTING_MAP.md` — action → agent mapping
- `ORG_DIRECTORY.md` — names, Discord IDs, emails, employee IDs
- `EMAIL_NOTIFICATIONS.md` — when to email

- `ON_BEHALF_EMAIL.md` — consent flow for sending on behalf
 - `WEEKLY_DIGEST.md` — executive weekly digest rules
 - `DECISION_TRACKER.md` — decision detection + logging
 - `TEAM_SUMMARIES.md` — team-specific summary rules
-

3) Workflow

3.1 Message & Transcript Flow

- 1) Supervisor scans all Discord channels on a 15-minute cron.
 - 2) If a message has `.txt/.md` attachment, it is treated as a meeting transcript.
 - 3) Action items are extracted and mapped to specialist agents.
 - 4) Task-Manager creates tasks via
`https://hacknation-openai-challenge.onrender.com/api/tasks`.
- 5) Supervisor DMs responsible people and posts an **inline summary** in the same channel.

3.2 Email Flow (AgentMail)

- All emails are sent via `agentmail_send.sh` (script-only policy).
 - Direct emails are sent automatically when a transcript explicitly requests it.
 - On-behalf emails require **explicit DM approval** from the sender.
-

4) Security & Governance

4.1 External Action Safety

- Email is allowed **only** via `agentmail_send.sh` (no direct SMTP).
- On-behalf email requires explicit approval.
- Sensitive operations are centrally controlled by OpenClaw config.

4.2 Explicit Rules

- Supervisor posts **no admin-chat messages**.
 - Max 20 items per run.
 - Neutral tone; no forced mentions unless required.
-

5) Tooling & Integrations

5.1 Discord

- Reads all channels (allowlist configured).
- Sends DMs using `message.send(target="user:<id>")`.

5.2 Task System

- Hosted API base URL:
`https://hacknation-openai-challenge.onrender.com`
- Tasks created via `/api/tasks` with `assignee_id / reporter_id`.

5.3 AgentMail

- Sender inbox: `chief-ai@agentmail.to`
- API key stored in OpenClaw config

- Script wrapper: `/home/ubuntu/.openclaw/scripts/agentmail_send.sh`

6) What We Implemented

Completed:

- Multi-agent architecture and routing logic
- Transcript parsing (.txt / .md attachments)
- Inline action summaries (no threads)
- DM notifications to responsible people
- AgentMail integration via script-only policy
- On-behalf email policy with explicit approval
- Org directory with Discord IDs + emails

Validated:

- Thread creation was confirmed (then switched to inline summaries)
- DM routing works via `user:<id>`
- AgentMail emails sent successfully after config updates

7) Example Demo Sequence

- 1) Drop transcript into #general
- 2) Supervisor extracts action items
- 3) Task-Manager creates tasks
- 4) Responsible people receive DMs
- 5) Inline summary posted in #general
- 6) Required emails sent (AgentMail)

8) Executive Digest, Decision Tracker, Team Summaries

Executive Weekly Digest

- **Purpose:** Weekly executive overview and on-demand summaries
- **Triggers:** Scheduled Mondays 09:00 Europe/Berlin; also on phrases like “weekly summary” or “What happened today?”
- **Output:** 5–10 bullet executive brief with Highlights, Decisions, Risks/Blockers, Next Actions

Decision Tracker

- **Purpose:** Detect explicit decisions (approvals, commitments, policy changes)
- **Behavior:** Logs decisions to `DECISIONS.md` (append-only) and DMs the owner when identified
- **Source:** Discord messages + transcript attachments

Team-Specific Summaries

- **Purpose:** Provide summaries tailored to a specific team on request
- **Triggers:** “summary for engineering”, “GTM summary”, “team summary: marketing”, etc.

- **Output:** Team-focused highlights, decisions, risks/blockers, next actions

9) Hackathon-Ready Enhancements (Optional)

- **Cross-channel decision tracker with reminders**
- **Accountability radar** for overdue tasks
- **Stakeholder-aware summaries** for different teams
- **External partner mode** with on-behalf approval flow

--- If you want a **1-page slide**, **system diagram**, or **live demo script**, I can generate those too.

I have incorporated the requested information into the "Completed" list in section 6 of your document.

Here is the updated "**6) What We Implemented**" section:

6) What We Implemented

Completed:

- Multi-agent architecture and routing logic
- Transcript parsing (.txt / .md attachments)
- Inline action summaries (no threads)
- DM notifications to responsible people
- AgentMail integration via script-only policy
- On-behalf email policy with explicit approval
- Org directory with Discord IDs + emails
- **Frontend centralized dashboard for employee access and management, which is also managed by the agents**

Validated:

- Thread creation was confirmed (then switched to inline summaries)
- DM routing works via user:<id>
- AgentMail emails sent successfully after config updates

System Architecture:

