

Teknisk Rapport: Mail Agent för Home Assistant

Projekt: Mail Agent Custom Component

Version: 0.16.0

Datum: 2025-12-17

Arkitektur: Modulär Event-driven AI Orchestrator

Roll: Home Assistant Expert / Senior Utvecklare

1. Sammanfattning

Mail Agent är en avancerad integrationsplattform för Home Assistant, designad för att fungera som en intelligent, autonom sekreterare. Dess primära syfte är att automatisera hanteringen av inkommande ostrukturerad information (e-postkallelser, bokningsbekräftelser, bilagor) och konvertera detta till strukturerade kalenderhändelser och handlingsbara notiser.

I en värld där viktig information ofta kommer i ostrukturerat format (brödtext i mail eller bifogade PDF-kallelser) överbryggar Mail Agent gapet mellan e-postkorgen och det smarta hemmet. Genom att kombinera traditionella kommunikationsprotokoll (IMAP/SMTP) med modern generativ AI (Google Gemini), skapas en robust pipeline för informationshantering.

2. Designfilosofi

Systemet är byggt kring tre kärnprinciper som styr all kodutveckling och arkitekturval:

2.1 Defensive Coding & Graceful Degradation

Systemet är designat för att vara feltolerant. Det får aldrig krascha på grund av otydlig data från en extern källa (mail eller AI).

- **Scenario:** AI:n hittar en händelse men misslyckas med att tolka en exakt tidpunkt (`start_time: null`).
- **Aktion:** Kalenderbokning hoppas över för att skydda dataintegriteten, men en notifiering skickas *ändå* ut med den information som finns. Användaren hålls alltid informerad, men systemet gissar aldrig.

2.2 Semantisk Förståelse framför Regler

Istället för att använda bräckliga "Regular Expressions" (Regex) som lätt går sönder om en avsändare ändrar sin mail-mall, använder Mail Agent semantisk analys via LLM. Den "läser"

mailet som en människa och förstår kontexten, oavsett format eller layout.

2.3 Total Kontroll över Dataflödet (Sovereignty)

För att garantera att bifogade filer (t.ex. PDF-kallelser) levereras korrekt till slutanvändaren, förlitar sig komponenten inte på externa beroenden för e-postutskick. Den implementerar en egen SMTP-klient för att säkerställa MIME-korrekthet, leveranssäkerhet och kontroll över headers.

3. Systemarkitektur och Komponenter

Systemet har i version 0.16.0 migrerat till en **Modulär Arkitektur** baserad på **Strategy Pattern**.

3.1 Kärnkomponenter

1. **Transportlagret (IMAP/SMTP):** Hanterar säker, krypterad kommunikation med e-postservern.
2. **Intelligenslagret (AI/LLM):** Använder Google Gemini API (via google-genai SDK) för multimodal analys (Text + Bild/PDF).
3. **Orkestreringslagret (Scanner & Processors):** Binder samman data med handlingar.

3.2 Strategy Pattern (Nytt i v0.14+)

Istället för en monolitisk kodbas, delegerar MailAgentScanner (Dispatcher) arbetet till specialiserade processorer.

- **Dispatcher:** `__init__.py` ansvarar endast för polling, uppkoppling och trådsäkerhet.
- **Processor:** `kallelse_processor.py` (KallelseProcessor) innehåller all logik för prompts, kalenderbokning och notifiering specifik för "Kallelser".
- **Fördel:** Detta gör det trivialt att lägga till stöd för t.ex. "Faktura-tolkning" i framtiden genom att bara skapa en `faktura_processor.py`, utan att riskera att förstöra existerande logik.

3.3 Trådsäkerhet: Global Scanning Lock (Nytt i v0.15+)

För att hantera asynkrona "Race Conditions" där långsamma AI-anrop orsakade överlappande processer (dubletter), implementerades en strikt global låsning.

- **Mekanism:** En `_is_scanning` flagga agerar som en semafor.
- **Logik:** Om en sökning pågår, avbryts alla nya försök från timern omedelbart.
- **Resultat:** Garanterar att ett mail **aldrig** bearbetas eller notifieras dubbelt, oavsett hur ofta sökindervallat triggas eller hur långsam AI:n är.

4. Detaljerat Dataflöde (Workflow)

Processen följer en cyklisk, händelsestyrd modell:

Fas 1: Inhämtning (Polling Layer)

1. **Trigger:** `async_track_time_interval` triggar `check_mail`.
2. **Låsning:** Systemet kontrollerar `_is_scanning`. Om ledigt -> Lås och fortsätt.
3. **Filtrering:** Hämtar endast mail markerade som UNSEEN.
4. **Separering:** Brödtext avkodas. Bilagor (PDF/Bilder) sparas temporärt i en lokal cache (`/config/www/mail_agent_temp`).

Fas 2: Delegering & Intelligens (Cognitive Layer)

1. **Delegering:** MailAgentScanner skickar data till vald processor (t.ex. KallelseProcessor).
2. **Temporal Context Injection:** Prompten injicerar dynamiskt "Nuvarande Tid" (t.ex. "Idag är 2025-12-17") för att ge AI:n en referensram.
3. **AI Analys:**
 - Filer laddas upp till Gemini.
 - Multimodal analys körs.
 - Svaret tvingas till strikt JSON.
4. **Sanering:** Koden hanterar om AI:n svarar med en lista [...] eller ett objekt {...}.

Fas 3: Exekvering (Action Layer)

1. **Kalender:** Om giltig tid finns -> `calendar.create_event` anropas.
2. **Mobil:** En "Actionable Notification" skickas (klick leder till kalendern).
3. **SMTP (E-post):**
 - **Dynamisk MIME:** Om bilagor saknas skapas ett rent text/html-mail (inget "gem"). Om bilagor finns skapas multipart/mixed.
 - **Avsändare:** Mailet skickas med det konfigurerade avsändarnamnet.
4. **Städning:** Temporära filer raderas och det globala låset släpps.

5. Funktioner och Finesser

5.1 Smart Tidshantering

AI-modeller saknar inbyggd tidsuppfattning. Mail Agent injicerar logik i prompten: "Om årtal saknas, välj det år som gör datumet kommande." Detta förhindrar att bokningar hamnar i dåtiden.

5.2 Direkt SMTP-implementering

Vi förlitar oss inte på Home Assistants inbyggda notify-tjänst för mail, då den ofta brister i hanteringen av lokala bilagor från dynamiska källor. Vår implementation garanterar att den bifogade PDF:en faktiskt kommer fram till mottagaren.

5.3 Konfigurerbarhet (Full UI)

Ingen YAML-redigering krävs.

- **Anslutning:** IMAP/SMTP serverinställningar.
- **Personalisering:** Anpassningsbart avsändarnamn.
- **Logik:** Val av tolkningstyp och AI-modell.
- **Integration:** Val av kalendrar och notifieringsenheter.

6. Systemkrav

För drift krävs följande:

1. **Home Assistant:** Version 2024.x eller senare (Stöd för Config Flow).
2. **Externa Konton:**
 - Google AI Studio konto (för API-nyckel).
 - E-postkonto med IMAP/SMTP (App-lösenord rekommenderas).
3. **Python-bibliotek:**
 - google-genai (Hanteras automatiskt).
 - Standardbibliotek: imaplib, smtplib, email.

7. Slutsats

Mail Agent (v0.16.0) är en mogen, produktions-klar lösning som transformerar Home Assistant från ett passivt styrsystem till en proaktiv assistent. Den robusta arkitekturen med global låsning och modulära processorer säkerställer att den fungerar pålitligt dygnet runt, och dess förmåga att förstå ostrukturerad data sparar tid och minskar den kognitiva belastningen för användaren.