

Teknisk Rapport: Mail Agent för Home Assistant

Projekt: Mail Agent Custom Component

Version: 0.12.1

Datum: 2025-12-16

Arkitektur: Event-driven AI Orchestrator

1. Sammanfattning

Mail Agent är en avancerad integration för Home Assistant designad att fungera som en **intelligent 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.

Genom att kombinera traditionella kommunikationsprotokoll (IMAP/SMTP) med modern generativ AI (Google Gemini), överbryggas komponenten gapet mellan digital kommunikation och det smarta hemmet.

2. Designfilosofi

Systemet är byggt kring tre kärnprinciper som styr all kodutveckling:

2.1 Defensive Coding & Graceful Degradation

Systemet får aldrig krascha på grund av otydlig data. Om AI:n hittar en händelse men misslyckas med att tolka en exakt tidpunkt, ska systemet "degradera snyggt".

- **Scenario:** Ett mail säger "Vi ses någon gång nästa vecka".
- **Aktion:** Kalenderbokning hoppas över (för att undvika felaktig data), men en notifiering skickas *ändå* ut med den information som finns. Användaren hålls alltid informerad.

2.2 Semantisk Förståelse framför Regler

Istället för att använda strikta "Regular Expressions" (Regex) som lätt går sönder om en mail-mall ändras, använder Mail Agent semantisk analys. Den "läser" mailet som en människa och förstår kontexten, oavsett format.

2.3 Total Kontroll över Dataflödet

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 och leveranssäkerhet.

3. Systemarkitektur och Dataflöde

Komponenten opererar i en cyklisk process uppdelad i fyra faser:

Fas 1: Inhämtning (Polling Layer)

- **Protokoll:** IMAP över SSL.
- **Logik:** En bakgrundsprocess (MailAgentScanner) övervakar inkorgen med ett konfigurerbart intervall (default 60s).
- **Filtrering:** Hämtar endast mail markerade som UNSEEN för att minimera API-anrop och bandbredd.
- **Hantering:** Separerar brödtext från bilagor. Bilagor (PDF/Bilder) sparas temporärt i en lokal cache (/config/www/mail_agent_temp).

Fas 2: Intelligens (Cognitive Layer)

- **Motor:** Google Gemini (via google-genai SDK).
- **Input:** Systemet skickar en multimodal prompt bestående av:
 1. Maillets text.
 2. De fysiska bilagorna (uppladdade temporärt till Gemini).
 3. En dynamisk system-prompt som injicerar "Nuvarande Tid" för att ge AI:n en tidsreferens.
- **Output:** Svaret tvingas till ett strikt JSON-format för maskinläsbarhet.

Fas 3: Beslutsfattande (Logic Layer)

Här sker valideringen av AI:ns svar:

1. **Event Found?** Är event_found: true? Om nej, avsluta.
2. **Tidsvalidering:** Finns ett giltigt start_time i formatet YYYY-MM-DD HH:MM:SS?
 - Ja: Fortsätt till kalenderbokning.
 - Nej: Hoppa över kalender, gå direkt till notifiering.

Fas 4: Exekvering (Action Layer)

- **Kalender:** Anropar `calendar.create_event` mot Home Assistants interna API.
- **Mobilnotis:** Skickar en "Actionable Notification" via notify-tjänsten (klick leder till kalendern).
- **E-post (SMTP):** Bygger ett MIMEMultipart-meddelande. Bifogar de ursprungliga filerna fysiskt och skickar via den konfigurerade SMTP-servern.

4. Nyckelfunktioner

4.1 Temporal Context Injection (Tidsmedvetenhet)

AI-modeller saknar inbyggd tidsuppfattning. För att lösa problemet med att modellen gissar fel år (t.ex. tolkar "16 dec" som innevarande år trots att vi är i december), injicerar koden följande logik i prompten vid varje körning:

"Idag är det: 2025-12-16. Utgå alltid från detta datum. Om årtal saknas, välj det år som gör datumet kommande."

4.2 Direkt SMTP-implementering

I version 0.12.0+ implementerades en egen SMTP-motor.

- **Problem:** Home Assistants standard `notify.email` hanterar inte alltid lokala filbilagor korrekt vid vidarebefordran av inkommande mail-strömmar.
- **Lösning:** Genom att använda Python's `smtplib` direkt i komponenten har vi full kontroll över headers och MIME-typer, vilket garanterar att bilagan kommer fram.

4.3 Trådsäkerhet (Threading)

Systemet körs i Home Assistants `SyncWorker`-trådar för att inte blockera huvudloopen (Event Loop). I version 0.12.1 korrigerades en kritisk bugg genom att säkerställa att SMTP-anropen görs synkront inom den redan existerande arbetstråden, istället för att försöka skapa nya asynkrona jobb felaktigt.

5. Konfiguration (UI)

Komponenten använder Home Assistants moderna **Config Flow** och **Options Flow**. Inget behov av att redigera YAML-filer.

5.1 Initial Setup

- IMAP Server, Port, Användarnamn, Lösenord, Mapp.

5.2 Alternativ (Options)

- **SMTP:** Inställningar för att skicka svar/notiser.
- **AI:** API-nyckel för Gemini och modellval (t.ex. gemini-2.0-flash-exp).
- **Integrationer:**
 - Val av Kalendrar (Dropdown).
 - Val av Mobil-notifieringstjänster (Dropdown).
 - E-postmottagare.

6. Systemkrav och Beroenden

För drift krävs följande:

1. **Home Assistant:** Version 2024.x eller senare.
2. **Externa Konton:**
 - Google AI Studio konto (för API-nyckel).
 - E-postkonto med IMAP/SMTP aktiverat (App-lösenord rekommenderas för Gmail).
3. **Python-bibliotek:**
 - google-genai (Installeras automatiskt via manifest.json).

7. Slutsats

Mail Agent transformerar Home Assistant från ett passivt styr-system till en proaktiv assistent. Genom att hantera den mest komplexa formen av data – ostrukturerad text och bild – sparar den tid och minskar risken för missade möten. Den robusta arkitekturen säkerställer att den fungerar pålitligt i en produktionsmiljö.