



id: SI_001

model: google/gemma-4-26b-a4b-it

seeds:

04_risiko: mulighet

03_relasjon: konkurranse

05_handling: implementer

01_perspektiv: strategisk

02_tidhorisont: kortsiktig

Instans SI_001

Seeds: strategisk | kortsiktig | konkurranse | mulighet | implementer

Arkitektonisk Analyse: Fra Copilot til Swarm-infrastruktur

For en organisasjon i helsesektoren med streng governance, er **strategisk** valget å unngå Copilot Studio for massiv skalering. Copilot Studio er for rigid og kostnadskrevenende for 100+ agenter. **Konkurransefordelen** ligger i å bygge en custom orchestration layer på Azure.

1. Arkitektur & Implementering

Valg: Azure OpenAI Service + Python (LangGraph/AutoGen) deployert på Azure Container Apps.

- **Hvorfor:** Copilot Studio/Power Automate vil knele under latency og kostnad ved 100 parallelle kall. Python-basert orkestrering gir kontroll over state management og token-budgeting.
- **Implementer:** Bruk **LangGraph** for å definere sykliske grafer. Dette muliggjør kontrollert feedback-loops som er umulige i en lineær Power Automate-flyt.

2. Spesialisering & Case-behandling (Case 1 & 2)

For å unngå "kaos av 100 meninger" (Case 1), må du implementere en **Hierarkisk Swarm**:

- **Worker Agents:** 100 spesialiserte instanser med unike system-prompts (f.eks. `role: "HIPAA-compliance-auditor"`).



- **Synthesizer Agent:** En “Chief Critic” som aggregerer input via en Map-Reduce tilnærming.
- **Case 2 (Inbox):** Bruk en **Azure Service Bus** som kø. En Dispatcher Agent leser meldingen, klassifiserer kompleksitet, og trigger en spesifikk agent-konfigurasjon.

3. Data Governance & Sikkerhet (Kritisk)

Strategisk sikring: Bruk **Azure AI Search** med Semantic Kernel.

- **Injection:** Ikke kopier data. Bruk RAG (Retrieval-Augmented Generation) hvor agenter kun henter relevante chunks via Entra ID-autentiserte API-kall.
- **Sikkerhet:** Data forblir i din Azure-tenant. Bruk Managed Identities for at agenter skal aksessere SharePoint/Dataverse uten lagrede credentials.

4. Cost & Observerbarhet

Kortsiktig risiko er “token-burn”.

- **Implementer:** Bruk **Azure Monitor + Application Insights** for å tracke `total_tokens_per_run`.
- **Optimering:** Bruk GPT-4o for “Synthesizer” og “Dispatcher”, men rull ut billigere modeller (GPT-4o-mini) til de 100 worker-agentene.

Failure Modes

Ikke bruk swarm hvis oppgaven er lineær. Swarm øker kompleksitet og feilrate eksponentielt hvis orkestreringslogikken er svak.

Neste steg:

1. **PoC (2 uker):** Sett opp en LangGraph-instans i Azure som kjører 5 spesialiserte agenter mot et begrenset SharePoint-datasett.
2. **Benchmark:** Test latency vs. kostnad ved å bytte mellom GPT-4o og GPT-4o-mini for worker-agenter.
3. **Governance-audit:** Valider at Entra ID-overføring av kontekst mellom agenter følger helsesektorens krav.