



id: SI_007

model: openai/gpt-5.4-mini

seeds:

04_risiko: usikkerhet

03_relasjon: samarbeid

05_handling: vent

01_perspektiv: strategisk

02_tidhorisont: historisk

Instans SI_007

Seeds: strategisk | historisk | samarbeid | usikkerhet | vent

Kort posisjon: I en Microsoft-sentrisk, helsestrengt organisasjon ville jeg ikke startet med «ekte» sverm via eksterne rammeverk. Strategisk sett er beste vei å bygge en **Azure OpenAI + orkestrering i Azure Functions/Durable Functions + Power Automate/Logic Apps**-stack, og bruke Copilot Studio bare der det er brukergrensesnitt og enkel prosessflyt. Historisk sett er dette den tryggeste evolusjonen fra M365-Copilot til multi-agent uten å rive opp governance.

1) Arkitekturvalg

- **Azure OpenAI + Python/Functions:** best for 100 agenter, parallelisme, kontroll, logging, Entra ID, private endpoints, DLP. Høyest operasjonell frihet.
- **Copilot Studio + Power Automate:** raskt for business workflows, men svakt for ekte swarm: begrenset koordinering, vanskeligere agent-spesialisering og observability.
- **CrewAI/Claude Swarm:** god idé-modell, men dårligere fit for strengt Microsoft-governance; øker risiko for dataflyt utenfor kontroll. **Anbefaling:** bruk Microsoft-native orkestrering som «motor», ikke rammeverket som ideologi.

2) Case 1: 100 perspektiver

Ikke dupliser prompt. Gi hver agent:

- **rolle**
- **mål**



- **kontekstpakke**
- **vurderingsrubrikk**
- **budsjett** (tokens/tid)

Eksempel:

```
agent = {  
  "role": "regulatory_reviewer",  
  "context": retriever.get(["policy_docs", "incident_logs"]),  
  "rubric": ["compliance_risk", "evidence", "recommendation"],  
}
```

Samling: bruk **2-lags synthese**: først cluster svar i temaer, så en meta-agent som skriver beslutningsnotat med konfliktmarkeringer. Unngå «100 meninger» ved å kreve at hver agent leverer **claim + evidence + confidence**.

3) Case 2: micro-case inbox

Bruk **queue-first**: Service Bus / Azure Queue → worker pool → result store (Dataverse/SQL). Hver worker kan dele template, men få **case-type modifiers**. Variabel kompleksitet håndteres med **routing agent** som klassifiserer case før processing. Vent på throughput, ikke real-time.

4) Data og sikkerhet

Bruk **RAG innen tenant**: SharePoint/Dataverse/e-post via Microsoft Graph, indeksér i **Azure AI Search** eller direkte retrieval. Kritisk: **Entra ID, managed identity, private endpoints, customer-managed keys, audit logging, DLP, sensitivity labels**. Ingen eksterne API-er med rå helse-data.

5) Observability og kost

Track per agent: tokens, latency, retries, failure class, retrieval-hit-rate. Bruk **Application Insights + Log Analytics**. Kost-ROI: **batching, caching av retrieval, small-model-first, fallback til større modell kun ved lav confidence**.

6) Transition

Realistisk: **4-6 uker** for pilot, **8-12 uker** til kontrollert produksjon. Quick wins: 1) 10-agent perspektivanalyse på ett governance-problem, 2) 50-case inbox med auto-routing og SLA-måling.



Når ikke swarm? Når oppgaven krever én korrekt beslutning, lav variasjon, eller når data-kobling er mer risikofyllt enn gevinsten.

Neste steg:

1. Velg ett pilotdomene og definer rubric + confidence-format.
2. Sett opp Azure OpenAI + Service Bus + App Insights i privat nett.
3. Kjør en 2-ukers benchmark mot Copilot-basert manuell prosess.