



id: SI_003

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

seeds:

04_risiko: sårbarhet

03_relasjon: makt

05_handling: vent

01_perspektiv: juridisk

02_tidhorisont: kortsiktig

Instans SI_003

Seeds: juridisk | kortsiktig | makt | sårbarhet | vent

Analysen avbryter den naive troen på “agenter som assistenter”. I en helsekontekst med streng governance er en sverm ikke et verktøy, men en utvidet maktstruktur som krever ekstrem kontroll.

1. Arkitektur: Maktbalansen mellom kontroll og fleksibilitet

Å velge Copilot Studio er et **kortsiktig** valg som ofrer agens for enkelhet. For 100 agenter er det utilstrekkelig.

- **Anbefalt:** Azure OpenAI Service (AOAI) + Python (LangGraph/AutoGen) i Azure Kubernetes Service (AKS).
- **Hvorfor:** Du beholder **juridisk** kontroll via Azure Virtual Networks og Managed Identities. Du bruker Microsoft-investeringen som en sikkerhetsbuffer (Entra ID), men flytter orkestreringen ut av “black box”-Copilot til en kontrollerbar Python-runtime.

2. Spesialisering og Aggregering (Case 1)

Du unngår “kaos” ved å innføre en **Hierarkisk Swarm**.

- **Design:** Ikke 100 flate agenter. Bruk 10 “Domain Experts” som hver leder 9 “Sub-agents”.
- **Kontekst:** Bruk Dynamic Prompt Injection. En “Regulatory Reviewer” får ikke bare en prompt, men en RAG-pipeline som injiserer spesifikke lovtekster (f.eks. Pasientjournalloven) via Azure AI Search.
- **Syntese:** En dedikert “Synthesizer Agent” mottar ikke rådata, men strukturerte JSON-objekter fra sub-agentene for å unngå hallusinert støy.



3. Distribusjon og Sikkerhet (Case 2 & Data)

For Case 2: Bruk en **Message Queue (Azure Service Bus)**.

- **Sårbarhet:** Den største risikoen er "Context Leakage".
- **Løsning:** Bruk **Semantic Kernel** for å kontrollere hvordan data hentes fra SharePoint/Dataverse. Agentene skal aldri "se" rådata; de skal operere på embeddings eller filtrerte metadata via en kontrollert API-gateway.

4. Observerbarhet og Kostnad

Uten streng monitorering vil en sverm eskalere kostnader eksponentielt.

- **Implementering:** Azure Monitor + OpenTelemetry. Du må spore "Token-per-task".
- **Strategi:** Bruk GPT-4o for "Orchestrator" og "Synthesizer", men tving de 90 sub-agentene over på GPT-4o-mini eller mindre modeller for å bevare ROI.

5. Transition & Failure Modes

Vent. Ikke rull ut alt samtidig.

- **Fase 1:** Single-agent RAG (veker).
- **Fase 2:** Orchestrated batch processing (måneder).
- **Når det feiler:** Hvis problemstillingen krever dyp kausal forståelse fremfor mønstergjenkjenning, er sverm-arkitektur en sårbarhet, ikke en styrke.

Neste steg:

1. Sett opp en **Azure Landing Zone** med isolerte vnet for AI-workloads.
2. Bygg en **PoC med LangGraph** som kjører 5 spesialiserte agenter mot et begrenset, syntetisk helsedatasett.
3. Definer **juridiske grensesnitt** (Audit Logs) for hver agent-interaksjon.