



id: SI_008

model: openai/gpt-5.4-mini

seeds:

04_risiko: usikkerhet

03_relasjon: makt

05_handling: implementer

01_perspektiv: operasjonell

02_tidhorisont: langsiktig

Instans SI_008

Seeds: operasjonell | langsiktig | makt | usikkerhet | implementer

For en **helse-strengt, cloud-only, Microsoft-sentrisk** organisasjon er mitt operative råd: **start med Azure OpenAI + Azure AI Foundry/Prompt Flow + en lett Python-orkestrator i Azure Functions/Container Apps**, ikke Copilot Studio som kjerne. Copilot Studio/Power Automate er bra for **governance og front-end**, men blir fort en maktbegrensning når du trenger 100 samtidige agenter, dynamisk routing, observability og batch-kontroll. Proprietære swarm-rammeverk gir fart, men øker **vendor risk** og svekker Microsoft-løftet.

Kort sagt: Microsoft først, men ikke Microsoft-only på orkestrering.

1) Arkitekturvalg

- **Best fit:** Azure OpenAI + Python-agent layer + Event Grid/Service Bus + Key Vault + Log Analytics.
- **Copilot Studio:** bruk som kanal for godkjente arbeidsflyter, ikke som swarm-motor.
- **CrewAI/Claude Swarm:** kun for prototyping; frikobler deg fra Microsoft governance.

2) 100-agent perspektivanalyse

Ikke dupliser samme prompt. Lag **rolle-matrise + kontekstdomene**:

- 10 roller × 10 underdimensjoner, f.eks. regulatory, cost, clinical risk, ops, security, vendor lock-in.
- Hver agent får: `role_prompt + evidence_pack + scorecard + dissent_instruction`.



- Output må være **strukturert JSON** med: funn, evidens, confidence, risks, recommended action.
- Aggregator: rangerer etter **evidensvekt + uenighet + policy-relevans**, ikke “flest meninger”.

3) Small-batch processing

Bruk **queue-driven inbox model**:

- Case legges i Service Bus/Queue med metadata (kompleksitet, sensitivitet, SLA).
- Router tildeler enten én agent eller en agent-pool.
- Del **template**, men ha **rolle-spesifikke overlays**. Full unique system prompt per agent er dyrt og ustabilt.
- Konsistens sikres med felles schema, validators og fallback-regler.

4) Data/kontekst-injection

Ikke kopier sensitive data til eksterne API-er. Bruk:

- **RAG i Azure AI Search** over SharePoint/Dataverse-indekser
- **Managed Identity + Entra ID**
- **Private endpoints / VNet integration**
- **DLP, audit logs, prompt redaction, data residency** For helse: vurder **model output filtering** og “no retention” der mulig.

5) Observerbarhet og kost

Track:

- tokens per agent, latency per hop, retry-rate, confidence drift, cost per case
- Log Analytics + App Insights + correlation IDs ROI-optimalisering:
- cache retrieval
- batch kall der mulig
- fallback til mindre modell for pre-screening
- stoppe tidlig ved lav usikkerhet

6) Transition

0-3 mnd: 1 use case, 5-10 agenter, batch-run, human review.

3-6 mnd: queue-orchestrering, RBAC, observability, cost dashboards.

6-12 mnd: 50-100 agenter, to parallelle workflows, produksjonsgovernance.



Failure mode

Swarm er feil hvis problemet egentlig er **én god retrieval + én sterk analytiker**. Bruk sverm når du trenger **parallellel usikkerhetsreduksjon** eller **mange små beslutninger**.

Neste steg:

1. Velg én pilot per use case og definer JSON-outputschema.
2. Sett opp Azure OpenAI + Service Bus + Log Analytics i en isolert subscription.
3. Lag en agent-aggregator med confidence scoring og human approval gate.