



id: SI_006

model: openai/gpt-5.4-mini

seeds:

04_risiko: mulighet

03_relasjon: tillit

05_handling: utforsk

01_perspektiv: operasjonell

02_tidhorisont: kortsiktig

Instans SI_006

Seeds: operasjonell | kortsiktig | tillit | mulighet | utforsk

For en **200-personers, strengt regulert helseorganisasjon** ville jeg valgt en

Microsoft-first, kontrollert swarm-arkitektur:

Azure OpenAI + Python/Functions + Azure Service Bus + Azure AI Search + Entra ID + Purview.

Dette preserverer M365-investeringen og gir reell multi-agent-skalerting. **Copilot Studio + Power Automate** er bra for enkel orkestrering, men blir fort for rigid for 100-agents parallell analyse og batch-distribusjon. Proprietære swarm-rammeverk kan gi fart, men øker leverandør- og compliance-risiko; i deres case er det et dårligere tillitsvalg kortsiktig.

1) Arkitekturvalg

- **Best praksis:** Azure OpenAI som modell-lag, Python-orkestrator som kontrollplan.
- **Copilot Studio:** bruk til front-end/workflow entry, ikke som motor for swarm.
- **Hvorfor:** du trenger deterministisk kontroll over routing, retries, quotas, logging, og datagrenser.

2) Agent-spesialisering

Ikke dupliser samme prompt 100 ganger. Lag **rolle-matrise + task-slicing**:

- 10-15 faste roller: regulatory, cost, ops, risk, evidence, counter-argument, summarizer, etc.



- Hver agent får:
 1. rolleprompt
 2. smal kontekstpakke
 3. eksplisitt vurderingsrubrikk
 4. output-format (JSON)
- Aggregér via **jury/arbiter-agent** + scoring, ikke “stemmer”.
Eksempel: hver agent returnerer `{finding, confidence, evidence, risk}`; arbiter rangerer, dedupliserer, og bygger konsensus + avvik.

3) Small-batch processing

Bruk **queue-first**:

- Inngang: DB/Dataverse → Service Bus queue
- Worker-farm: Azure Functions/Container Apps
- Agent-allokering: ledig worker henter case, velger template etter kompleksitet (rules/metadata)
- Consistency: én **system template**, flere **task templates** per case-type.
Variabel kompleksitet styres med “depth budget” (f.eks. 1, 3 eller 5 analysepass).

4) Data og sikkerhet

- Hold data i **tenant boundary**; bruk **Managed Identity**, private endpoints, RBAC, DLP.
- Hent SharePoint/Dataverse via Microsoft Graph; send kun nødvendige utdrag til modellen.
- Bruk **Purview** for klassifisering og audit.
- Unngå direkte eksterne AI-API-er for sensitiv pasientnær info.

5) Observerbarhet og kost

- Logg per agent: tokens, latency, fail-rate, retry, model-version.
- Bruk Application Insights + Log Analytics.
- Kostoptimering: caching av kontekst, batching av små tasks, billig modell til første-pass, dyr modell kun for arbiter/edge cases.



6) Transition

0-6 uker: 1 use case, 5-10 agenter, batch-only, strikt logging.

6-12 uker: 25-50 agenter, queue-based routing, jurymodell.

3-6 mnd: 100-agent perspektivanalyse + small-case pipeline i produksjon.

Når swarm ikke passer: hvis beslutningen krever lav kompleksitet, høy presisjon, eller data ikke kan fragmenteres uten tap av klinisk/operasjonell mening.

Neste steg:

1. Velg én pilot-case med tydelig output-skjema.
2. Design 10 roller + JSON-kontrakt + arbiter.
3. Sett opp Azure landing zone: Entra, Purview, Service Bus, App Insights.