



id: SI\_008

model: deepseek/deepseek-v3.2

seeds:

04\_risiko: mulighet

03\_relasjon: makt

05\_handling: vent

01\_perspektiv: juridisk

02\_tidhorisont: langsiktig

## Instans SI\_008

**Seeds:** juridisk | langsiktig | makt | mulighet | vent

**Arkitekturvalg:** Forbindelsen mellom **juridisk** ansvar og langsiktig kontroll (**makt**) over data dikterer et rent Azure OpenAI-arkitekturvalg. Unngå Copilot Studio for swarm – det låser deg i lav fleksibilitet. Bygg et Python-basert orkestreringslag på Azure Container Instances som snakker med Azure OpenAI. Dette bevarer investeringen, gir full kontroll og unngår proprietære swarm-rammeverk som kan kompromittere helsedata. Muligheten ligger i å standardisere på én plattform, men **vent** med å låse deg til én LLM – design for modellbytte.

**Implementering for Case 1 & 2:** Dette er din kjernemulighet. For perspektivanalyse (Case 1), spesialisér 100 agenter via distinkte system-prompts og kontekst-fragmenter fra Dataverse/SharePoint (f.eks., «For denne analysen, fokuser kun på kapittel 4 i retningslinje X og kostnadslinje Y»). Sammenstill via et «syntese-agent»-lag som bruker et tre for å trekke ut konsensus og dissens, ikke bare en tekstsammenføyning. For micro-case (Case 2), bruk en Azure Service Bus-kø. Hver agent henter en case, men alle deler en master system-prompt med parametriserte instruksjoner (f.eks., `{case_type: 'klage', urgency: 'høy'}`). Dette sikrer konsistens.

**Kritisk Kompleksitet:** Den **juridiske** risikoen ved data injection er størst. All kontekst må hentes on-demand via Azure Synapse eller sikrede API-kall til Microsoft 365. Aldri send rå helsedata til LLM-API-et. Bruk strenge Entra ID-scopes. Kostnadssporing må være innfødt – instrumenter hver agent-kall med Log Analytics for token-forbruk. Swarm er ikke rett løsning for enkle, lineære oppgaver eller når deterministiske svar er et **juridisk** krav.

**Neste konkrete steg:**

1. Sett opp et pilot-prosjekt i en isolert Azure-resourcegruppe med 5 spesialiserte agenter for én test-case fra Case 1. Mål: valider kontekst-injection og syntesemetode.
2. Utfør en juridisk og sikkerhetsgjennomgang av den foreslåtte dataflyten med IT-governance-teamet. Få godkjenning for prinsippet før du skalerer.
3. Bygg det grunnleggende kø- og agent-orchestration-verktøyet for Case 2 ved å bruke Azure Functions og Service Bus, og test med syntetiske data.