



id: SI\_003

model: deepseek/deepseek-v3.2

seeds:

04\_risiko: mulighet

03\_relasjon: makt

05\_handling: implementer

01\_perspektiv: økonomisk

02\_tidhorisont: langsiktig

## Instans SI\_003

**Seeds:** økonomisk | langsiktig | makt | mulighet | implementer

**Arkitekturvalg:** For en økonomisk og langsiktig bevaring av Microsoft-makt, er hybrid-orkestrering nøkkelen. Implementer **Azure Logic Apps som orkestrator** – den abstraherer kompleksitet og holder alt innenfor Microsofts sikkerhetsperimeter. Bruk **Azure OpenAI** (GPT-4) for agentkjerne med **Python Azure Functions** for spesialisert logikk. **Unngå Copilot Studio** – det er for lav-kode og låser deg til simple workflows, ikke 100 parallelle agenter. Proprietære rammeverk (CrewAI) introduserer unødvendig risiko og kompleksitet i et strengt helsemiljø.

**Implementering av spesialisering (Case 1):** Makt ligger i meningsfull rollefordeling. Ikke dupliser prompts. Hver agent får en unik **“kontekstpakke”** fra SharePoint/Dataverse via søk med Azure AI Search (semantisk, sikret med Entra ID). Eksempel: “Regulatory reviewer”-agenten får kun dokumenter fra “Regulatory” SharePoint-mappe og en instruksjon om å analysere kun etter lovverk HPRL. **Sammenstilling** skjer via en “synthesizer”-agent som bruker konflikt med en fast metaprompt (“Finn de 3 høyest rangerte konsensuspunktene”).

**Data & Sikkerhet:** All kontekst injiseres via **Azure AI Search med sikker filtrering**. Data forblir i din tenant; kun søkevektorer og spørringer går til modellen. Dette er muligheten for swarm uten brudd på compliance.

**Failure mode:** Swarm er overkill for enkle Q&A eller dokumentgenerering. Det er en **makt-teknologi** for komplekse analyser med høy konfliktgrad eller distribuerte, uensartede cases.

### Neste steg:

1. **Implementer** en minimal Proof-of-Concept med 3 spesialiserte agenter (f.eks. regulatorisk, økonomisk, operasjonell) mot et test-datasett i Azure AI Search, orkestrert av én Logic App.



2. Etabler et enkelt cost-tracking-dashboard i Azure Monitor for å følge token-forbruk og latency per agent-type.
3. Kjør en parallell analyse av ett eksisterende problem med både standard Copilot og din 3-agent-sverm, og kvantifiser tidsbesparelse og analysedybde.