



id: SI_004

model: deepseek/deepseek-v3.2

seeds:

04_risiko: sårbarhet

03_relasjon: tillit

05_handling: implementer

01_perspektiv: strategisk

02_tidhorisont: langsiktig

Instans SI_004

Seeds: strategisk | langsiktig | tillit | sårbarhet | implementer

Arkitekturtrade-off: Azure OpenAI + Python-agenter er strategisk valg. Copilot Studio er forbruksorientert og lukket; proprietære rammeverk skaper sårbarhet (vendor lock-in). Azure holder data innenfor Microsofts tillitsramme og skalerer langsiktig.

Implementer med Azure Container Instances for orchestration – billigere og enklere enn Kubernetes for 100 agent-batch.

Agent-spesialisering: For perspektivanalyse, definer 5-7 kjerneroller (f.eks. «GDPR-compliance agent» med Dataverse-tilgang, «patient-flow simulator»). Deretter permutér disse med 15-20 kontekst-filtre (region, tidsramme, scenario). Dette gir 100 unike kombinasjoner uten kaos. Aggregering via et «meta-analyst»-agent som trekker ut top-3 konsensuspunkter og top-5 dissensområder.

Inbox-modell: Azure Queue Storage + Logic App trigger. Hver agent henter én case, men bruker **samme system-prompt-template** med case-spesifikke data injiserte via placeholders. Consistency sikres med en valideringsagent som sjekker alle outputs mot et sett minimumskriterier før aggregering.

Data-injection & sikkerhet: Bruk Azure OpenAI's data grounding med SharePoint/ Dataverse via indekserte søk. Data kopieres **ikke** til API; referanser sendes. Største sårbarhet: agentens prompt kan lekke sensitiv kontekst via «few-shot examples». Mitiger med prompt-masking: erstatt konkrete patient-ID'er med «{PATIENT_REF}» før API-call.

Cost-optimering: Batch 100 agent-calls til én Azure Function, logg alle til Application Insights med custom dimensions (agent-type, tokens). ROI: Bruk GPT-4-turbo for analyse-agenter, men fallback til GPT-3.5-turbo for «transkriber»-roller. Cache grounding-data-søk i 10 minutter.



Transition: Kort sikt (3 måneder): Implementer 5-agent swarm for en spesifikk rapporteringsprosess (Case 2). **Middels sikt** (6 måneder): Utvid til 15-agent perspektivanalyse på strategiske dokumenter (Case 1). Bruk dette for å demonstrer verdi: sammenlign swarm-analyse vs. enkelt-Copilot på samme problem – vis **dissensdybde**.

Begrensning: Swarm-arkitektur er **ikke** riktig når: (1) problemet krever en enkelt, autoritativ svar (regulatory approval), (2) latency < 1 min er pålagt, (3) data-konteksten er så liten at 100 agenter blir redundant noise.

Neste steg:

1. Sett opp en PoC med Azure OpenAI, en Dataverse-tabell og 3 Python-agenter (regulatory, operational, cost) som analyserer én felles case – mål: se token-kostnad og output-forskjell.
2. Design en «agent-rolle-matrix» for din organisasjon: 5 kjerneveroller x 20 kontekst-filtre.
3. Book møte med IT-security for å godkjenne data-grounding-arkitekturen spesifikt for helsedata.