



id: SI_006

model: deepseek/deepseek-v3.2

seeds:

04_risiko: sårbarhet

03_relasjon: tillit

05_handling: vent

01_perspektiv: strategisk

02_tidhorisont: kortsiktig

Instans SI_006

Seeds: strategisk | kortsiktig | tillit | sårbarhet | vent

Arkitektur-valg: Forkast Copilot Studio for orkestrering – det blir for stivt. Bruk Azure Durable Functions (Python) som «svermmoderator». Det er en **strategisk** investering som bevarer Microsoft-investment, men åpner for hybride modeller (OpenAI + nylig kunngjorte Phi-3-mini). Dette håndterer **sårbarhet** rundt vendor lock-in.

Implementeringskompleksitet: For Case 1 (100 agenter): Ikke lag 100 unike roller fra dag én – det er **kortsiktig** kaos. Start med 5 kjerne-perspektiver (f.eks. regulatorisk, klinisk, finansielt, operasjonelt, pasientsikkerhet). Bruk en «perspektiv-seeding»-metode: hver hovedagent får en unik kontekstsnippet fra SharePoint-listen din og en rolle-definerende system-prompt. De genererer så 5-10 under-spørsmål som tildeles «under-agentene» (de resterende 95) via en oppgave-kø. Dette skalerer **tillit** i output ved å strukturere divergens.

For Case 2 (inbox): Bruk Azure Service Bus som jobb-kø. En orchestration-funksjon (moderator) leser en batch, vurderer kompleksitet (f.eks. via metadata/ordtelling), og tildeler cases til agent-pools med forskjellige modeller eller temperatur-innstillinger. Alle agenter i samme pool deler en system-prompt-mal, men får sin unike case-data som bruker-melding. Dette balanserer konsistens og fleksibilitet.

Kritisk sikkerhet: All data må bli i din Azure-tenant. Bruk Azure OpenAI on your data eller Bring Your Own Data (BYOD)-mønsteret med Azure AI Search for indeksering av SharePoint/Dataverse. Agentene berikes kun via søk og henting, med strenge RBAC via Entra ID. Ingen rå data kopieres ut. Monitorer token-forbruk per agent-«sess» i Application Insights og sett automatiske knekkpunkter for å stoppe uventet kostnadseksplasjon.

Failure mode: Sverm er ikke riktig for enkle, lineære Q&A-oppgaver hvor et enkelt Copilot-svar er nok. Det er heller ikke riktig hvis du ikke kan **vente** på batch-kjøring –



real-time swarm krever massiv overhead for koordinering. Den største risikoen er å miste oversikten i 100 parallelle trender.

Operasjonelt:

1. Fase 1 (8 uker): Bygg Durable Functions-orchestrator med 5 perspektiv-agenter på et intern test-case. Integrer med AI Search for 1 dokumentkilde.
2. Fase 2 (6 uker): Skaler til inbox-modell for et enkelt, godt definert case-type (f.eks. pasienthenvendelse-klassifisering). Implementer full logging og kostnadssporing.
3. Fase 3 (Løpende): Utvid rollepaletten og datakildene gradvis basert på innsikt fra fasene.

Neste steg:

1. Sett opp en liten Proof-of-concept med Azure Durable Functions og én Azure OpenAI-instans for å simulere en 3-agenter «mini-sverm» på et sikret test-datasett.
2. Kartlegg og klassifiser 10 konkrete dokumenter fra SharePoint i et skjema som kan brukes til perspektiv-seeding (f.eks. «Dokument X er mest relevant for regulatorisk og finansielt perspektiv»).
3. Definere klare knekkpunkter i kostnads- og latens-mål for når en batch-kjøring skal avbrytes og eskalere til en menneskelig operator.