



id: SI_004

model: google/gemma-4-26b-a4b-it

seeds:

04_risiko: mulighet

03_relasjon: konkurranse

05_handling: vent

01_perspektiv: strategisk

02_tidhorisont: kortsiktig

Instans SI_004

Seeds: strategisk | kortsiktig | konkurranse | mulighet | vent

Arkitektur og Implementering

1. Arkitekturvalg: Den hybride veien Strategisk sett er Copilot Studio for begrenset for Case 1. For å bevare Microsoft-investeringer uten å låse seg til lav-kode, må du velge **Azure OpenAI Service + Python (LangGraph/Autogen)** på Azure App Service.

- **Pros:** Full kontroll over agent-logikk, Entra ID-integrasjon, og data-residency.
- **Cons:** Krever utviklingsressurser.
- **Konkurransfordel:** Ved å bruke LangGraph kan du bygge sykliske grafer (ikke bare lineære kjeder), som er kritisk for Case 1.

2. Case 1: Perspektiv-divergens vs. Konvergens Ikke dupliser prompts. Bruk en **Hierarkisk Orchestrator-modell**.

- **Spesialisering:** Definer agenter via en `AgentManifest` (JSON) der hver agent har en unik `SystemPersona` og `KnowledgeAccessScope`.
- **Syntese:** For å unngå kaos, bruk en "Synthesizer Agent" som kjører en `Consensus-weighted Aggregation`. Den ser ikke på alle 100 svar, men på de 10 mest divergerende og de 5 mest konsistente for å identifisere spenningsfelt.

3. Case 2: Inbox-modell og Skalering Bruk en **Azure Service Bus** som kø-mekanisme.

- **Arkitektur:** En "Dispatcher" leser fra køen → sender til en "Router Agent" → allokere til en spesialisert agent-instans.
- **Consistency:** Bruk en sentralisert `Schema Registry` for output. Hver agent må returnere JSON som valideres mot et Pydantic-skjema før lagring i Dataverse.



4. Sikkerhet og Data (Helse-kontekst) Strategisk risiko ved helsedata krever **RAG via Azure AI Search** med OneBox-tilnærming. Data forlater aldri din Azure-tenant. Bruk Managed Identities for all kommunikasjon mellom agenter og datakilder. Ingen API-nøkler i kode.

5. Observabilitet og Kostnad Kortsiktig risiko er “token-bleed”. Implementer **Azure Monitor + Application Insights** for å spore token-bruk per `CorrelationID` (som følger en case gjennom hele svermen).

- **Optimering:** Bruk GPT-4o for “Orchestrator/Synthesizer” og GPT-4o-mini for de 100 “Worker”-agentene.

6. Transition: Roadmap

- **Fase 1 (Quick Win):** Implementer Case 2 (Batch processing) for administrative oppgaver. Lav risiko, høy synlig verdi.
- **Fase 2:** Implementer Case 1 som et beslutningsstøtteverktøy for ledelsen.
- **Vent:** Ikke prøv å automatisere kliniske beslutninger i fase 1; hold svermen i “advisory mode”.

Neste steg:

1. Sett opp en Azure Landing Zone med Managed Identities og Azure OpenAI.
2. Prototype en LangGraph-struktur med 5 spesialiserte agenter (ikke 100) for å teste syntese-logikken.
3. Definer JSON-skjemaer for Case 2-output for å sikre dataintegritet.