



id: SI_008

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

seeds:

04_risiko: resiliens

03_relasjon: konkurranse

05_handling: forhandle

01_perspektiv: strategisk

02_tidhorisont: langsiktig

Instans SI_008

Seeds: strategisk | langsiktig | konkurranse | resiliens | forhandle

Analysen skifter fokus fra “hvordan implementere” til “hvordan overleve implementeringen”. I et landskap med legacy-systemer er agentisk KI ikke et verktøy, men en uregulert aktør i et eksisterende økosystem.

Arkitektonisk kontrollmodell: “The Intermediary Layer”

Mainstream foreslår integrasjon; jeg foreslår **isolasjon via en kontroll-proxy**. Du må ikke gi agenten tilgang til systemene, du må gi den tilgang til en Policy Enforcement Point (PEP).

- Strategisk arkitektur:** Innfør en “**Semantic Gateway**”. Dette er en mellomvare som oversetter agentens intensjon (LLM-output) til deterministiske API-kall. Dette hindrer at agenten “snakker” direkte med eldre fagsystemer.
- Sikkerhetsmekanismer:** Glem RBAC alene. Du trenger **ABAC (Attribute-Based Access Control)** kombinert med en “**Decision Guardrail**”. Hver agent-handling må valideres mot en policy-motor (f.eks. OPA - Open Policy Agent) som sjekker: Er handlingen innenfor kontekst? Er volumet (rate-limiting) normalt? Er verdien av endringen (impact) under en definert terskel?
- Resiliens gjennom “Circuit Breakers”:** Implementer autonome rollback-mekanismer. Hvis en agent endrer data i et legacy-system uten tilstrekkelig logging, må systemet automatisk kutte forbindelsen (kill-switch) ved avvik i anomalideteksjon.



Tekniske blindsoner og risiko

- **Konkurransen om ressurser:** Agenter kan skape “race conditions” i eldre systemer som ikke håndterer samtidighet fra ikke-menneskelige aktører.
- **Forhandlingsasymmetri:** Agenter som kommuniserer med eksterne API-er kan utilsiktet “forhandle” bort data eller priser gjennom uforutsette sekvenser av kall.
- **Eskalerings-loop:** Den største risikoen er ikke én feil, men en kaskade der agent A korrigerer feil gjort av agent B, noe som skaper en uendelig loop av systemendringer.

Konklusjon

Ikke bygg for autonomi; bygg for **kontrollert observabilitet**. Din konkurransefordel ligger ikke i hvor raskt agenten handler, men i hvor robust du kan rulle tilbake en autonom beslutning.

Neste steg:

1. **POC på Policy-Proxy:** Bygg en prototype av en “Semantic Gateway” som fungerer som et filter mellom agenten og ett enkelt legacy-API.
2. **Definer “Blast Radius”:** Kartlegg hvilke data som er irreversible. Begrens piloten til systemer med høy “recoverability”.
3. **Etabler “Human-in-the-loop” terskler:** Definer nøyaktig hvilke parametere (beløp, sletting, ekstern kommunikasjon) som krever manuell godkjenning før utførelse.