



id: SI\_005

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

seeds:

04\_risiko: mulighet

03\_relasjon: tillit

05\_handling: vent

01\_perspektiv: strategisk

02\_tidhorisont: langsiktig

## Instans SI\_005

**Seeds:** strategisk | langsiktig | tillit | mulighet | vent

Analysen av agentisk KI i deres kontekst krever et paradigmeskifte: Vi må slutte å se på agenter som “verktøy” og begynne å behandle dem som “digitale ansatte” med begrenset fullmakt.

**Arkitekturprinsipper og sikkerhet** Dagens deterministiske sikkerhetsarkitektur er utilstrekkelig. I stedet for tradisjonell RBAC, må dere implementere **ABAC (Attribute-Based Access Control)** kombinert med en **Policy Decision Point (PDP)** som fungerer som en “digital vaktmester”.

- Sandboxing via “Shadow Execution”:** Før en agent får skriveadgang til eldre fagsystemer, må alle handlinger kjøres i en digital tvilling eller en logisk sandkasse for å validere utfall.
- Mikro-segmentering av identitet:** Hver agent må ha sin egen unike maskin-identitet (Workload Identity), ikke bare dele en service-konto. Dette er avgjørende for **tillit** og etterrettelighet.
- Intervensjons-logikk:** Implementer en “Human-in-the-loop” (HITL) trigger for alle handlinger merket med høy risiko (f.eks. endring av bankkontonummer eller sletting av arkivdata).

**Tekniske blindsoner og risiko** Den største faren er **“Semantic Drift”** i eldre systemer. En agent forstår ikke nødvendigvis de implisitte reglene i et 20 år gammelt fagsystem. Hvis agenten sender en forespørsel som er syntaktisk korrekt, men semantisk destruktiv, vil ikke dagens logging fange det opp før skaden er skjedd.

En ofte undervurdert risiko er **“Cascade Failure”**: En agent prøver å løse en feil, trigger en API-rate-limit, som igjen trigger en feilmeldinger som agenten tolker som et behov for å “starte på nytt” – en autonom loop som lammer infrastrukturen.



**Strategisk vurdering** Selv om muligheten for effektivisering er stor, dikterer den **langsiktige** stabiliteten at dere må **vente** med full autonomi i produksjon. Ikke la piloten (6–9 mnd) bli en “rask og uoversiktlig” implementering. Bygg kontrollaget først, ikke som et påbygg.

**Konkrete neste steg:**

1. **Etabler en “Policy Engine”:** Velg en teknologi (f.eks. Open Policy Agent - OPA) for å definere granulære regler som er uavhengige av de gamle fagsystemene.
2. **Kartlegg “High-Blast Radius” API-er:** Identifiser de 5 mest kritiske endringspunktene i systemlandskapet som må ha obligatorisk menneskelig godkjenning.
3. **Design en “Circuit Breaker”:** Utvikle en teknisk mekanisme som automatisk deaktiverer agentens tilgang hvis logganalyse viser uvanlige mønstre (anomali-deteksjon).