



id: SI_009

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

seeds:

04_risiko: mulighet

03_relasjon: makt

05_handling: utforsk

01_perspektiv: operasjonell

02_tidhorisont: kortsiktig

Instans SI_009

Seeds: operasjonell | kortsiktig | makt | mulighet | utforsk

Analysen identifiserer et fundamentalt gap mellom deterministisk arkitektur og agentisk uforutsigbarhet.

Arkitektonisk kontrollmodell

For å unngå systemisk kollaps må vi skifte fra tillitsbasert tilgang til kontrollert eksekvering. **Prinsipp:** "The Interceptor Pattern". Agenten skal aldri snakke direkte med fagsystemer. Vi må implementere en **Policy Enforcement Point (PEP)** mellom agenten og API-ene. Denne fungerer som en operasjonell vaktbikkje som validerer hver handling mot en definert policy-motor (f.eks. OPA - Open Policy Agent).

Sikkerhetsmekanismer og maktbalanse

Nåværende RBAC er utilstrekkelig. Vi trenger **Attribute-Based Access Control (ABAC)**.

- **Sandboxing:** Agenten opererer i et isolert miljø hvor den kun har tilgang til "Read-Only" replikaer av data, med unntak av spesifikke, pre-godkjente "Write-funksjoner".
- **Maktbegrensning:** Innfør Rate-Limiting på handlingstype, ikke bare forespørsler. Maks 5 endringer per time i kritiske fagsystemer før menneskelig eskalering kreves.
- **Nøkkelrotasjon:** Bruk kortlevde, dynamiske tokens (f.eks. via HashiCorp Vault) som er bundet til agentens spesifikke oppgave-ID, ikke en generell tjenestekonto.



Tekniske blindsoner og risiko

Den største kortsiktige risikoen er **“Semantic Escalation”**. En agent kan bruke legitime API-kall til å manipulere logikk på måter utviklerne ikke forutså (f.eks. ved å endre en status som trigger en utbetaling).

- **Blindsonen:** Eldre systemer mangler “intent-logging”. De logger at noe skjedde, ikke hvorfor. Uten kontekstuell logging blir revisjon umulig.
- **Undervurdert risiko:** “Feedback loops” hvor agenten endrer data som den selv bruker som grunnlag for neste beslutning, noe som fører til eksponentiell feilforplantning.

Mulighet: Den kontrollerte utforskningen

Vi må ikke bygge for perfekt sikkerhet (som kveler autonomien), men for **kontrollert feiling**. Design for Automated Rollback: Hver agentisk handling må pakkes i en transaksjon som kan reverseres umiddelbart hvis policy-motoren detekterer avvik.

Neste steg:

1. **Pilot-isolasjon:** Etabler en “Shadow API Gateway” som logger alle agent-forsøk uten å utføre dem, for å kartlegge uforutsett adferd.
2. **Policy-prototype:** Implementer en enkel OPA-instans som begrenser agentens skriveadgang til ett enkelt, ikke-kritisk API.
3. **Logg-standardisering:** Tving alle agent-interaksjoner gjennom en sentralisert “Intent-Logger” som kobler API-kall til agentens resonneringskjede.