



id: SI_003

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

seeds:

04_risiko: mulighet

03_relasjon: konkurranse

05_handling: forhandle

01_perspektiv: juridisk

02_tidhorisont: kortsiktig

Instans SI_003

Seeds: juridisk | kortsiktig | konkurranse | mulighet | forhandle

Denne analysen forkaster “fullstendig sikkerhet” som et uopnåelig ideal innenfor 6–9 måneders pilotvinduet. I stedet fokuserer vi på å **forhandle** frem en kontrollert risikoeksponering som sikrer virksomhetens **konkurranseevne** uten å utløse **juridisk** ansvar.

Arkitektur: Fra Zero Trust til “Constrained Autonomy”

Glem en total Zero Trust-implementering nå; det tar for lang tid. Bruk en **kortsiktig** strategi basert på “**Proxy-based Guardrails**”. Plasser agentene bak en dedikert API-gateway (f.eks. Kong eller Tyk) som fungerer som en tvungen policy-enhet.

- **Prinsipp:** Agenter får aldri snakke direkte med legacy-systemer. De snakker med en “Mediator API” som inneholder de nødvendige transaksjonelle garantier og idempotens-logikk som de gamle systemene mangler.
- **Execution Graph:** Modeller agentens handlinger som en DAG (Directed Acyclic Graph) i sanntid. Hvis en node i grafen bryter med forhåndsdefinerte mønstre (f.eks. uventet rekursjon), må systemet utføre en hard kill.

IAM: Dynamisk delegasjon

Nåværende RBAC er utilstrekkelig. Vi må **forhandle** frem en modell med “**Just-in-Time (JIT) Scoped Tokens**”.

- Agenten opererer med en identitet som kun har “read” tilgang. Ved behov for handling, må den be om et kortlevd (minutter) token via en kontroll-instans som verifiserer handlingens intensjon mot en policy-motor (OPA). Dette minimerer det **juridiske** ansvaret ved eskalering.



Risiko og Blindsoner: Den største muligheten

Den største **muligheten** ligger i å automatisere det som i dag er manuelle flaskehalser, men den største blindsonen er **“Semantic Drift”**: At agenten tolker en svak API-kontrakt korrekt teknisk, men feil logisk, og skaper kjedereaksjoner i legacy-monolitter.

- **Konkurransefortrinn:** Ved å bygge en audit-pipeline som logger ikke bare hva som skjedde, men hvorfor (prompt + kontekst + beslutning), skaper dere et unikt datasett for kontinuerlig forbedring.

Konkrete neste steg:

1. **Etabler en “Mediator Layer”:** Ikke koble agenten til legacy; bygg et moderne, deterministisk API-lag mellom dem i løpet av de neste 3 månedene.
2. **Implementer “Circuit Breakers” på tvers av domener:** Definer harde grenser for antall endringer per time/bruker/system som trigger umiddelbar isolering.
3. **Juridisk Sandbox:** Definer et avgrenset datasett og systemområde (lav risiko) hvor agenten får operere med autonomi, fremfor å rulle ut bredt.