



id: SI_008

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

seeds:

04_risiko: usikkerhet

03_relasjon: tillit

05_handling: implementer

01_perspektiv: operasjonell

02_tidhorisont: langsiktig

Instans SI_008

Seeds: operasjonell | langsiktig | tillit | usikkerhet | implementer

Analysen avdekker et fundamentalt gap mellom agentisk intensjon og eksisterende infrastruktur. Dere forsøker å implementere en “super-bruker” i et landskap som mangler grunnmuren for kontroll.

Arkitektur og Kontroll: Fra tillit til verifisering

Å basere kontroll på eksisterende RBAC i legacy-systemer er en **operasjonell** blindvei. Dere må implementere en “**Interventionist Proxy Architecture**”.

- **Prinsipp:** Agenter skal aldri snakke direkte med API-er. Alle kall må gå via en Agent Gateway som fungerer som en runtime-isolasjon.
- **Implementering:** Bruk OPA (Open Policy Agent) som en “sidecar” for hver agent-instans. Policy-as-code må definere ikke bare hva (GET/POST), men hvor mye (kvantitative grenser, f.eks. “maks 5 endringer per minutt”).
- **Execution Graph:** Modeller agentens plan som en rettet asyklisk graf (DAG) før utførelse. Hvis grafen inneholder sirkulære avhengigheter eller uforutsigbare hopp, blokkeres den umiddelbart.

IAM og Integrasjon: Dynamisk begrensning

Eksisterende IAM er statisk; agentisk drift krever **ephemeral identity**.

- **Strategi:** Bruk Workload Identity Federation. Agenter tildeles kortlivede (minutter) tokens med ekstremt snever audience og scope.
- **Risiko:** Manglende idempotens i legacy-API-er er den største kilden til **usikkerhet**. En agent som “re-tryer” en ikke-idempotent operasjon etter en timeout, kan skape duplikate transaksjoner og systemisk kaos. Implementer



Circuit Breakers på gateway-nivå som kutter forbindelsen hvis feilraten stiger over 2%.

Blindsoner og Emergent Behavior

Den største **langsiktige** risikoen er ikke bevisste angrep, men “latent coupling”. Agenter kan oppdage uformelle logiske koblinger mellom systemer som menneskelige arkitekter har oversett, og trigge kjedereaksjoner (f.eks. endring i lagerstatus som trigger en feilaktig faktureringsprosess).

Tillit bygges ikke gjennom fravær av feil, men gjennom deterministisk gjenoppretting. Dere mangler compensating transactions. Uten en audit-pipeline som knytter en spesifikk prompt til en spesifikk databaseendring, vil dere aldri kunne rekonstruere en hendelse.

Konkrete neste steg:

1. **Etabler en “Shadow Execution” pilot:** Kjør agenter i read-only modus mot de mest kritiske systemene i 3 måneder for å kartlegge deres naturlige “execution graphs”.
2. **Bygg Agent Gateway:** Ikke rør legacy-systemene direkte. Implementer en proxy som håndterer rate-limiting og OPA-evaluering.
3. **Definer “Kill-Switch” protokoll:** Utvikle en manuell og automatisert prosedyre for å isolere alle agent-identiteter på tvers av hele landskapet på under 30 sekunder.