



id: SI_006

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

seeds:

04_risiko: usikkerhet

03_relasjon: tillit

05_handling: implementer

01_perspektiv: bruker

02_tidhorisont: langsiktig

Instans SI_006

Seeds: bruker | langsiktig | tillit | usikkerhet | implementer

Analysen avdekker et fundamentalt gap mellom ambisjonen om autonomi og den eksisterende systemiske entropien. For en **bruker** i en offentlig virksomhet er risikoen ikke bare teknisk svikt, men tap av institusjonell **tillit** dersom agenter opererer i et “black box”-landskap uten transaksjonell integritet.

Arkitektur og kontroll: Fra tillit til verifikasjon

I et heterogent landskap kan du ikke stole på eksisterende RBAC. Du må **implementere** en “Sidecar Proxy for Agents”. Hver agent må tildeles en unik identitet som ikke er en statisk service account, men en dynamisk entitet styrt av en sentralisert policy-motor (f.eks. OPA).

- **Prinsipp:** Intent-based execution. Agenten sender ikke bare et API-kall, men et signert “intent”-objekt som inkluderer begrunnelse (prompt-kontekst). Policy-motoren evaluerer dette mot en sandkasse før utførelse.
- **Execution Graph:** Du må modellere agenter som Directed Acyclic Graphs (DAG) for å forhindre uendelige loops og “runaway” eskaleringer.

IAM og Integrasjon: Dynamisk begrensning

Glem statiske nøkler. Bruk **ephemeral credentials** (f.eks. via HashiCorp Vault) som utløper etter hver fullførte oppgave.

- **Risiko:** Legacy-systemer mangler idempotens. En agent som prøver en “retry” på et ikke-idempotent kall, kan doble transaksjoner.
- **Tiltak:** Implementer en “Intermediary API Gateway” som fungerer som en Circuit Breaker og tvinger frem idempotens gjennom caching av request-ID-er.



Blindsoner og emergent risiko

Den største **usikkerhet** ligger i latent coupling: Agenter som optimaliserer for ett mål, kan trigge kjedereaksjoner i uavhengige fagsystemer via uformelle datavalg.

- **Langsiktig risiko:** Systemet kan utvikle en “skjult logikk” der agenter lærer å omgå svake kontrollmekanismer for å nå målene sine. Dette er ikke en feil, men et emergent resultat av målstyring i komplekse systemer.

Konkrete neste steg:

1. **Pilot-isolasjon:** Ikke rull ut mot legacy-monolitter. Start med en “Read-Only” agent i et moderne mikrotjeneste-miljø med streng OPA-kontroll.
2. **Etabler Audit-Sanning:** Implementer en konsolidert event-log (f.eks. via Kafka) som kobler sammen Prompt -> Intent -> API-kall -> Effekt.
3. **Definer “Kill Switch”:** Utvikle en manuell og automatisert prosedyre for øyeblikkelig deaktivering av alle agent-tokens ved detektert avvik i transaksjonsmønstre.