



# Debrief — google/gemma-4-26b-a4b-it

---

- Instanser: 9
  - Tokens inn: 15,048
  - Tokens ut: 7,533
  - Kostnad: \$0.0038
  - Kjørt: 2026-04-16 00:49:05.686657+00:00
- 

## Sverm-debrief: Agentisk KI i komplekse systemlandskap

---

### Konsensus

Svermen er enstemmig om at nåværende sikkerhetsarkitektur (RBAC og deterministisk logging) er fundamentalt utilstrekkelig for autonome agenter. Følgende punkter er kritiske for suksess:

- **Fra RBAC til ABAC/PaC:** Man må implementere en sentralisert policy-motor (f.eks. **Open Policy Agent - OPA**) som validerer handlinger basert på kontekst og attributter, ikke bare identitet.
  - **Intermediary Proxy Layer:** Agenter må aldri kommunisere direkte med fagsystemer. En “Agent Gateway” eller “Semantic Proxy” er nødvendig for å fungere som kontrollpunkt, protokoll-konverter og regulator.
  - **Human-in-the-loop (HITL) ved risiko:** Autonomi må begrenses av terskelverdier. Handlinger med høy risiko (irreversible endringer, store beløp, sensitiv data) krever asynkron menneskelig godkjenning.
  - **Observability-first:** Det må etableres en uforanderlig (immutable) logg som kobler agentens resonnement (hvorfor) med systemets handling (hva).
- 

### Dissens

Det er en strategisk konflikt mellom **implementering** og **venting**:

- **Implementerings-fløyen:** Ser muligheten i å bygge kontrollaget parallelt med piloten for å lære gjennom kontrollert eksponering.



- **Vent-fløyen:** Advarer sterkt mot å rulle ut agenter i et fragmentert legacy-miljø før kontrollaget er fullstendig modent. De frykter “silent corruption” og systemisk kollaps som følge av for rask utrulling.

---

## Blindsoner avdekket

Svermen identifiserte risikoer som går utover tradisjonell IT-sikkerhet:

- **Semantisk eskalering/drift:** Agenter kan utføre tilsynelatende legitime, men logisk destruktive handlinger ved å kombinere flere API-kall på måter utviklere ikke har forutsett.
- **Kaskadefeil via autonomi:** En agent som forsøker å “fikse” en feil i et eldre system kan trigge en uendelig loop av feilmeldinger og automatiske forsøk, som resulterer i en DoS-tilstand (Denial of Service) i fagsystemene.
- **Kontekst-lekkasje:** Risikoen for at sensitiv informasjon eksponeres gjennom agentens resonneringsprosess mot eksterne LLM-tjenester.

---

## Anbefalinger

1. **Bygg en “Policy Proxy” (POC):** Implementer OPA mellom en agent og ett enkelt, ikke-kritisk API for å teste Policy-as-Code i praksis.
2. **Definer “Blast Radius”:** Kartlegg alle irreversible handlinger i legacy-systemene og innfør harde tekniske sperrer (Kill-switches) for disse.
3. **Etabler “Shadow Mode”:** Kjør piloten i en “observe-only”-modus i de første 3-6 månedene, hvor agenten foreslår handlinger uten å utføre dem, for å validere policy-motoren.
4. **Implementer “Intent-Logging”:** Tving alle agent-interaksjoner gjennom en sentralisert logg som korrelerer agentens `intent_id` med systemets `transaction_id`.