



id: SI_004

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

seeds:

04_risiko: mulighet

03_relasjon: avhengighet

05_handling: vent

01_perspektiv: strategisk

02_tidhorisont: kortsiktig

Instans SI_004

Seeds: strategisk | kortsiktig | avhengighet | mulighet | vent

Analysen avdekker en kritisk ubalanse mellom ambisjon og infrastruktur. Dere planlegger å slippe en autonom aktør inn i et landskap preget av deterministisk kaos.

Arkitektur og kontroll: Strategisk begrensning Glem fullstendig Zero Trust i piloten; dere har ikke ressursene. Implementer i stedet en **“Gateway-Proxy-modell”**. Agenten skal aldri snakke direkte med legacy-systemer. Alle kall må gå gjennom en kontrollert proxy som fungerer som en Policy Enforcement Point (PEP).

- **Execution Graph:** Modeller agentens handlinger som en rettet asyklisk graf (DAG). Hvert steg må valideres mot en definert “intent-kontrakt” før utførelse.
- **Risiko:** Uten OPA/Styra er dere avhengige av koden i proxyen. Dette er en kortsiktig sårbarhet.

IAM og API: Avhengighet av kontekst Dagens RBAC er utilstrekkelig. Dere må skape en **“Just-in-Time (JIT) Scoping”**-modell.

- Bruk Short-lived tokens (TTL < 15 min) som er strengt bundet til en spesifikk oppgave-ID.
- **API-risiko:** Legacy-systemer uten idempotens er en tidsinnstilt bombe. Hvis agenten gjør et dobbelt-kall pga. timeout, kan data korrumpes. Implementer en Idempotency Layer i proxyen som mapper agent-ID + Task-ID til unike nøkler.

Logging og Blindsoner: Mulighet og fare Deres største blindsoner er **“Latent Coupling”**: Agenten endrer data i System A, som trigger en uforutsett batch-jobb i System B, som krasjer System C.

- **Audit:** Dere trenger ikke bare logger, dere trenger en Reasoning Log. Hver handling må knyttes til en `prompt_id` og en `confidence_score`.



- **Emergent Behavior:** Dette kan ikke modelleres fullt ut, men kan begrenses via Circuit Breakers. Hvis agenten utfører >X endringer i minuttet, eller endrer data med høy integritetsverdi, må systemet gå i “Hard Kill”-modus.

Konklusjon: Vent Pilotvinduet på 6-9 måneder er for aggressivt gitt manglende sentralisert policy og audit-pipeline. Dere risikerer å bygge en “autonom teknisk gjeld” som er umulig å rulle tilbake.

Neste steg:

1. **Etabler en “Intermediary Proxy Layer”** som fungerer som den eneste inngangsporten til legacy-systemer (skaper kontroll uten å fikse legacy).
2. **Definer “Blast Radius” per agent:** Begrens agentens tilgang til kun 2-3 spesifikke, ikke-kritiske datasett i pilotfasen.
3. **Bygg en “Kill-switch” mekanisme** som kan terminere alle aktive sesjoner på tvers av API-er umiddelbart.