



id: SI\_005

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

seeds:

04\_risiko: sårbarhet

03\_relasjon: tillit

05\_handling: implementer

01\_perspektiv: strategisk

02\_tidhorisont: kortsiktig

## Instans SI\_005

**Seeds:** strategisk | kortsiktig | tillit | sårbarhet | implementer

Analysen avdekker et fundamentalt gap mellom ambisjon og infrastruktur. Dere planlegger å slippe løse en autonom kraft inn i et landskap som er for skjørt til å tåle dens uforutsigbarhet.

**Strategisk risiko og sårbarhet** Deres største **sårbarhet** er ikke KI-en selv, men “latent coupling” i legacy-systemer. Når en agent utfører en handling i en monolitt uten idempotens, skaper den en irreversibel tilstandsendring. Uten transaksjonell kontroll er agentisk KI ikke et verktøy, men en “chaos monkey” som opererer med høy hastighet. **Strategisk** må dere skifte fokus fra “hva kan agenten gjøre” til “hva er det absolutte skadepotensialet per handling”.

**Kortsiktig implementering av kontroll** Siden dere mangler OPA og Zero Trust, er en full utrulling umulig innen 6–9 måneder. Dere må **implementere** en “Proxy-Gateway”-modell umiddelbart. Ikke gi agenter direkte API-tilgang; la dem snakke med en kontroll-proxy som fungerer som en menneskeskapt sandkasse.

- **IAM:** Bruk Ephemeral Service Tokens med en levetid på maks 15 minutter. Hver agent-sesjon må ha et unikt `correlation_id` som knytter prompt → plan → API-kall.
- **Arkitektur:** Bygg en “Circuit Breaker” for agenter. Hvis agenten trigger  $>3\%$  feilrater eller forsøker å endre data utenfor definerte JSON-skjemaer, må sesjonen termineres umiddelbart (Hard Kill).
- **Audit:** En audit-log er verdiløs hvis den bare lagrer tekst. Dere må logge Intent (hva agenten prøvde å oppnå) mot Effect (hva som faktisk skjedde i databasen).

**Tillit gjennom begrensning** Tillit i et autonomt system bygges ikke gjennom overvåking, men gjennom deterministiske grenser. Hvis et system mangler



kontraktshåndheving, må agenten operere i et “Read-Only” modus inntil en Policy Engine er på plass.

**Neste steg:**

1. **Definer “Blast Radius”:** Identifiser de 3 mest kritiske API-ene/tabellene og innfør manuell godkjenning (Human-in-the-loop) for alle endringer her i pilotfasen.
2. **Bygg Proxy-laget:** Implementer en sentralisert API-gateway som fungerer som agentens eneste inngangsport, med innebygd rate-limiting og skjemavalidering.
3. **Simuler feil:** Utfør en “Red Team”-øvelse der dere simulerer en agent i en loop som kaller et legacy-API 100 ganger i sekundet.