



id: SI_003

model: openai/gpt-5.4-mini

seeds:

04_risiko: usikkerhet

03_relasjon: tillit

05_handling: forhandle

01_perspektiv: operasjonell

02_tidhorisont: kortsiktig

Instans SI_003

Seeds: operasjonell | kortsiktig | tillit | usikkerhet | forhandle

Min lesning er operasjonell og kortsiktig: ikke bygg “full agentplattform” først; bygg et **forhandlet handlingsrom** der agenten kun kan utføre det dere eksplisitt tåler å reparere. I et landskap med legacy, svake kontrakter og begrensede ressurser er hovedrisikoen ikke “ond AI”, men **ukontrollerte sideeffekter i marginale systemer**.

Arkitektur og kontroll

- Bruk **deny-by-default** med tre soner:
 1. **Observe**: agent leser, foreslår, ingen writes.
 2. **Bounded execute**: kun forhåndsgodkjente handlingstyper, tidsvindu og datasett.
 3. **Escalate**: alt utenfor rammen krever menneskelig godkjenning.
- Innfør en **execution graph** per agentoppdrag: noder = beslutning, verktøyskall, datauttak, sideeffekt; kanter = årsak/avhengighet. Dette gir sporbarhet og stoppunkt.
- Kjør agenten i **isolert runtime** (container/sandbox) med nettverksutgang kun via proxy/gateway.

IAM og tilgang

- Tildel **ephemeral credentials** per oppdrag, ikke per agent.
- Token-scoping må være **handling + ressurs + varighet**; f.eks. “read customer-status i system X i 15 min”.



- Bruk **delegert autorisasjon** med kort levetid og automatisk rotasjon. RBAC alene er for grovt; ABAC brukes bare på de få høy-risiko handlingene.

Integrasjoner og API-sikkerhet

- Anta at 30-50 % av interne API-er mangler idempotens eller korrekt kontrakt. Da må agenten ha **idempotency keys**, replay-beskyttelse og “dry-run” der mulig.
- Sett **rate limits per agent og per ressurs**, pluss circuit breaker ved 3-5 feil på rad eller uvanlig sekvenslengde.

Logging og hendelser

- Audit-pipeline må fange: prompt, kontekst, verktøykall, respons, beslutning, sideeffekt, korrelasjons-ID.
- Lag en **kill switch** som automatisk isolerer agenten ved policybrudd, anomaliteter eller loop-indikatorer.

Blindsoner / worst case

- Undervurdert risiko: **latent coupling** mellom gamle systemer, der én “harmløs” write trigger batchjobber, varsler eller økonomiske prosesser.
- Modellér worst case som: agent gjør feil i et høy-integritets system, feil oppdages sent, og rollback er delvis umulig. Da må første mål være **begrenset blast radius**, ikke maksimal autonomi.

Neste steg

1. Velg 1-2 use cases med lav integritetsrisiko og målbar rollback.
2. Etabler en enkel policy-gateway og execution graph før pilot.
3. Definer stoppkriterier: hvilke 5 handlinger agenten aldri får gjøre uten menneskelig godkjenning.