



id: SI_004

model: openai/gpt-5.4-mini

seeds:

04_risiko: mulighet

03_relasjon: samarbeid

05_handling: implementer

01_perspektiv: strategisk

02_tidhorisont: kortsiktig

Instans SI_004

Seeds: strategisk | kortsiktig | samarbeid | mulighet | implementer

For en offentlig virksomhet med 6–9 måneders pilotvindu ville jeg **ikke** startet med “full autonomi”, men med en **begrenset operasjonsmodell**: agenten får gjøre små, reverserbare handlinger i utvalgte domener. Strategisk bør dere designe for kontroll først, og kortsiktig levere et mønster som kan skaleres.

Arkitektur og kontroll

- Del opp i tre soner: **observasjon**, **forslag**, **begrenset utførelse**. Agenten må aldri ha direkte bred tilgang til systemlandskapet.
- Bruk **policy-as-code** som et obligatorisk kontrollpunkt før hvert kall: OPA/ Gatekeeper er mer relevant enn “ren” app-logikk. Hvis dere ikke kan rulle ut sentral motor raskt, lag en **policy-gateway** foran kritiske API-er.
- Modellér agentens **execution graph** som en DAG med noder for: intent, plan, policy-check, kall, resultat, kompensasjon. Hver node får korrelasjons-ID og signerte metadata.

IAM og tilgang

- Bruk **ephemeral credentials** per oppgave, med TTL på minutter, ikke timer.
- Tildel **task-scoped service accounts** med eksplisitt system- og databegrensning.
- Delegert autorisasjon bør være **“just-in-time, just-enough”**: ingen permanente agent-nøkler, og rotasjon etter hver sesjon.



Integrasjoner og API-sikkerhet

- Legacy-API-er uten idempotens er høy risiko. Sett dem i “**manual approval only**”-klasse i første fase.
- Bruk **rate limiting, circuit breakers og idempotency keys** der det er mulig.
- Innfør en agent-spesifikk **egress-gateway** som logger, filtrerer og kan stoppe trafikk.

Logging, revisjon og hendelser

- Audit-pipeline må fange: prompt, plan, policyavgjørelse, API-kall, respons, sideeffekt og kompensasjon.
- Logg til en **append-only** kanal med tidsstempler og signering. Ikke stol på distribuert logging alene.
- Ved avvik: automatisk **freeze → revoke credentials → isoler agent → varsle**.

Blindsoner og risiko

- Den største blindsonen er ikke “hallusinasjon”, men **latent coupling**: små handlinger som treffer uforutsigbare avhengigheter.
- Worst case bør modelleres som **kjedefeil i høy-integritetsdata**: feil oppdatering + replikerte integrasjoner + treg oppdagelse.
- Emergent behavior bør behandles som et **operasjonelt sikkerhetsproblem**, ikke bare et KI-problem.

Neste steg

1. Velg 1-2 lavrisiko use cases og klassifiser alle handlinger i **tillat / begrensn / forby**.
2. Etabler en minimal **policy-gateway + audit-pipeline** før pilotstart.
3. Definer stoppkriterier: runaway-loop, policy-brudd, ukjent API-effekt, eller manglende sporbarhet.