



id: SI_001

model: openai/gpt-5.4-mini

seeds:

04_risiko: sårbarhet

03_relasjon: tillit

05_handling: utforsk

01_perspektiv: strategisk

02_tidhorisont: langsiktig

Instans SI_001

Seeds: strategisk | langsiktig | tillit | sårbarhet | utforsk

Min linse er at agentisk KI ikke først og fremst er et AI-problem, men et **styrings- og tillitsproblem i et sårbart systemlandskap**. Derfor bør dere ikke starte med “hva kan agenten gjøre?”, men med “hva tåler organisasjonen at en agent gjør, og hvordan oppdager vi avvik før skade blir varig?”.

Arkitekturprinsipper

1. **Containment før autonomi:** Del agentenes verden i soner med eksplisitte “blast radius”-grenser. Start med leseoperasjoner og lavrisiko-handlinger, ikke endringer i kjernefagsystemer.
2. **Policy ved kjøring, ikke bare design:** Innfør runtime-beslutning for hvert steg. OPA/Gatekeeper/Kyverno er nyttig, men dere trenger en felles policy-gateway foran alle agentkall, også mot legacy.
3. **Execution graph som førstestats-objekt:** Hver agent må ha et sporbarhetskart: mål → plan → delsteg → kall → respons → sideeffekt. Dette er grunnlaget for revisjon og rollback.
4. **Fail closed, men med operasjonell sikkerhetsventil:** Ved tvetydighet skal agenten stoppe, ikke “prøve litt til”.

IAM og tilgang

- Bruk **ephemeral credentials** med varighet på 5–15 minutter, ikke statiske nøkler.



- Modellér tilgang som **dynamisk, kontekstuell og oppgavebasert**: system, datasett, tidspunkt, risikonivå.
- Del autorisasjon i to: **planleggeren** kan foreslå, **executor** kan utføre bare innenfor smale scopes.
- Koble agentidentitet til tjenestekonto + kortlevd token + signert oppgave-ID.

API- og integrasjonsrisiko

Legacy-API-er uten idempotens er en hovedfare: én feil kan bli ti. Legg inn **rate limiting, circuit breakers, duplikatdeteksjon og “dry-run” der det finnes**. Hvis et API kan endre tilstand uten transaksjonell garanti, må agenten få svært begrenset rettighet.

Logging og hendelser

Bygg en **append-only audit pipeline** for prompts, beslutninger, kall, responser og avvik. Distribuert logging er ikke nok; dere trenger et konsolidert revisjonsløp, selv om det er enkelt i starten.

Blindsoner

Undervurder ikke:

- latent coupling mellom systemer,
- loops med “nesten vellykkede” kall,
- datakvalitet som får agenten til å handle riktig på feil grunnlag,
- emergent behavior når små feil får kumulativ effekt.

Min uvanlige anbefaling

Ikke gi agenten bred autonomi i pilotvinduet. Gi den **smal autonomi med høy observabilitet**. Det bygger tillit over tid og reduserer sårbarhet mer enn ambisiøs fullautomatisering.

Neste steg

1. Definér 3 risikoklasser for agenthandlinger og knytt dem til eksplisitte policy-regler.
2. Etabler en minimal policy-gateway + audit-pipeline rundt én pilotprosess.



3. Velg ett legacy- og ett moderne system og test “blast radius”, rollback og stoppmekanismer før bred utrulling.