



id: SI_001

model: openai/gpt-5.4-mini

seeds:

04_risiko: sårbarhet

03_relasjon: makt

05_handling: vent

01_perspektiv: strategisk

02_tidhorisont: langsiktig

Instans SI_001

Seeds: strategisk | langsiktig | makt | sårbarhet | vent

Agentisk KI bør behandles som en ny **maktaktor** i virksomhetsarkitekturen, ikke som “enda en integrasjon”. Den viktigste feilen er å gi agenter bred funksjonell tilgang og håpe at gode prompts og logger holder. Det gjør de ikke. I en offentlig virksomhet med fragmentert IAM og eldre fagsystemer er **venting** ofte den riktige sikkerhetsstrategien: start med begrenset autonomi, høy observasjonsevne og gradvis utvidelse.

Arkitekturprinsipper

1. **Minste handlingsrom per oppgave:** Ikke gi “agent-tilgang”, gi oppgavespesifikke capabilities, f.eks. les sak , foreslå vedtak , opprett utkast , men aldri iverksett i første fase.
2. **To-trinns utførelse:** Agenten kan foreslå og forberede, men irreversible handlinger krever menneskelig godkjenning eller sterk policy-gate.
3. **Kapslede soner:** Del opp i read zone, write sandbox og production bridge. Legacy-systemer skal kun nås via kontrollerte adaptere, ikke direkte.
4. **Kontroll før intelligens:** Policy-motor og transaksjonskontroll må ligge foran agenten, ikke etterpå.

Nødvendige sikkerhetsmekanismer

- **IAM:** Separate service accounts per agent og per oppgave, korte tokens, OIDC/OAuth2 med kort TTL, mTLS mellom komponenter.
- **Policy-motor:** OPA/liknende for beslutningsregler, med eksplisitt allowlist per handling og datakilde.
- **Sandboxing:** Isolert kjøring, nettverksbegrensning, egress-kontroll, rate limiting, og “dry-run” for alle skriveoperasjoner.
- **Nøkkelrotasjon:** Automatisk, minst hver 24-72 time for høyere risiko-bruk.



- **Logging:** Uforanderlige audit-logger med korrelasjons-ID for hver agentbeslutning, input, policyutfall og API-kall.

Blindsoner som ofte undervurderes

- Agenten kan kombinere “uskyldige” datakilder til sensitiv innsikt.
- Eldre fagsystemer mangler ofte idempotens; en dobbel handling kan bli en reell hendelse.
- Logging blir ofte fragmentert akkurat der dere trenger etterprøvnbarhet.
- Prompt injection via eksterne API-er eller dokumenter kan bli en ny angrepsflate.

Kontrollmodell Bygg en **gradert autonomi**: nivå 0 = lesing, nivå 1 = utkast, nivå 2 = begrensede skriveoperasjoner, nivå 3 = kun etter eksplisitt human approval. Hver overgang krever testet rollback og hendelseshåndtering.

Neste steg

1. Velg 2–3 lavrisiko brukstilfeller og definer eksplisitte capabilities og stoppekriterier.
2. Etabler en minimal policy-gate og audit-kjede før pilot.
3. Kartlegg legacy-systemer for irreversible operasjoner, manglende idempotens og svak logging.