



id: SI\_002

model: openai/gpt-5.4-mini

seeds:

04\_risiko: sårbarhet

03\_relasjon: tillit

05\_handling: implementer

01\_perspektiv: strategisk

02\_tidhorisont: langsiktig

## Instans SI\_002

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

Agentisk KI bør behandles som en **privilegert, men ustabil medarbeider**: nyttig for tempo, men aldri som en generell aktør med fri tilgang. Min vurdering er at den største feilen er å tenke “modell + API = automatisering”. For offentlig sektor må styringen være **komposisjonell, ikke modell-sentrisk**: kontrollen legges i et mellomlag av policy, godkjenning og sporbarhet, ikke i selve agenten.

### Arkitekturprinsipper

#### 1. Minste handlingsrom per oppgave

- Gi agenten **oppdragsbaserte rettigheter**, ikke varige breddefullmakter.
- Bruk korte, spesifikke tokens og tidsavgrensede “leases” (f.eks. 5–15 min).
- Del opp handlinger i **lese → foreslå → klargjøre → utføre**.

#### 2. Policy-sandkasse utenfor agenten

- Innfør en **policy enforcement point** mellom agent og systemer.
- All skriving til fagsystemer må gå via en **transaksjonsgate** med regler for beløp, sakstype, felt, og irreversible operasjoner.
- For eldre systemer: bruk “shadow actions”/simulering før faktisk kall.

#### 3. Tillit bygges gjennom etterprøvbarehet

- Én felles hendelseslogg med korrelasjons-ID på tvers av API-er.
- Logg alltid: input, verktøyvalg, policy-sjekk, resultat, avvik, menneskelig overstyring.



- Ikke stol på agentens egen forklaring; stol på **reproduserbar kjede av beslutninger**.

## Sikkerhetsmekanismer

- **RBAC + oppgavestyrte roller**; ingen direkte systembruker med “super-agent”.
- **Rate limiting** per agent, per datakilde, per tidsrom.
- **Isolerte nettverkskorridorer** mot eksterne tjenester; standard nekt på utgående data.
- **Krypterte, kortlivede nøkler** med automatisk rotasjon og tjenesteidentitet per workflow.
- **Datafiltrering/DLP** før eksterne API-kall, spesielt for persondata og skjermingsverdig info.

## Blindsone

Det undervurderes ofte at legacy-systemer kan bli **irreversible skadeflater**: ett feil kall kan opprette saker, sende brev eller trigge utbetalinger uten enkel rollback. Derfor må rollback designes som **kompensasjon**, ikke reversering.

## Kontrollmodell

Ha en “**autonomi-matrise**”:

- Grønn: les og foreslå
- Gul: klargjør med menneskelig godkjenning
- Rød: ingen direkte handling

## Neste steg

1. Kartlegg 5–10 konkrete agent-oppgaver og klassifiser dem i grønn/gul/rød.
2. Etabler en policy-gate og felles audit-logg for én pilotflyt.
3. Test rollback/kompensasjon på et legacy-system før pilot i produksjonsnær sone.