



id: SI\_008

model: openai/gpt-5.4-mini

seeds:

04\_risiko: mulighet

03\_relasjon: makt

05\_handling: vent

01\_perspektiv: strategisk

02\_tidhorisont: kortsiktig

## Instans SI\_008

**Seeds:** strategisk | kortsiktig | makt | mulighet | vent

Agentisk KI bør ikke behandles som «enda en integrasjon», men som en ny aktør med delvis egen vilje. Det strategiske grepet er derfor å begrense **makt** før man utvider **mulighet**. I en offentlig virksomhet med kort pilot (6–9 mnd) og svakt IAM/støyete logging, er den største feilen å gi agenten bred les/skriv-tilgang og så håpe på kontroll.

## Arkitekturprinsipper

### 1. Minste handlingsrom per oppgave

Gi agenten oppgavespesifikke tillatelser, ikke systemvidde. Bruk “capability tokens”/scoped credentials per workflow, med TTL på 5–30 minutter.

### 2. Menneske i løkken for irreversible handlinger

Alt som kan endre data permanent, utløse vedtak, sende eksterne kall eller påvirke økonomi/personvern, skal kreve eksplisitt godkjenning.

### 3. Kontrollplan utenfor agenten

Policy må ligge i en ekstern beslutningsmotor eller orkestrator, ikke i prompten. Agenten foreslår; policy-laget avgjør.

### 4. Vent som designprinsipp

Innfør tvungen “pause” ved usikkerhet, lav tillit, avvik fra mønster, eller når data er sensitiv. En agent som venter er bedre enn en som improviserer.

## Nødvendige sikkerhetsmekanismer

- **RBAC + tjenestekontoer per agent/oppgave**, aldri delte nøkler.



- **Policy-sandkasse:** allowlist av systemer, API-metoder og datatyper; blokker alle andre.
- **Rate limiting og kvoter** per agent, per time, per datasett.
- **Isolasjon:** egen runtime/container, nettverkssegmentering, ingen direkte tilgang til interne admin-grensesnitt.
- **Nøkkelrotasjon:** korte levetider, automatisk rotasjon, audit på bruk.
- **DLP/innholdsfiltrering** før eksterne API-kall for å hindre lekkasje.

---

## Blindsoner

Eldre fagsystemer har ofte skjulte sideeffekter: et “read” kan trigge batchjobber, et “update” kan skrive ufullstendig. Typisk undervurderte risikoer er:

- indirekte eskalering via integrasjonskontoer,
- fragmentert logging som gjør hendelser uetterprøvbare,
- rollback som ikke finnes fordi systemene er ikke-transaksjonelle,
- agenten lærer uheldige snarveier og utnytter dem raskt.

---

## Kontrollmodell

Bygg en **todelt modell**:

- **Planlagte handlinger:** agent foreslår sekvens + risikoestimat.
- **Utførelse:** orkestrator validerer policy, logger, og gjennomfører med minimal privilegium.

Dette er mindre “autonomi”, men mer **mulighet** på sikt fordi det skaper trygg drift og læring.

## Neste steg

1. Definer 5–10 konkrete agent-oppgaver og klassifiser dem: les, foreslå, utfør, eller krev godkjenning.
2. Lag en pilot-arkitektur med egen sandbox, scoped credentials og policy-gate før noen integrasjon bygges.
3. Test tre feilscenarier: feilskriv, datalekkasje og uautorisert eksternt kall — og mål om rollback faktisk virker.