



id: SI\_008

model: openai/gpt-5.4-mini

seeds:

04\_risiko: sårbarhet

03\_relasjon: makt

05\_handling: utforsk

01\_perspektiv: strategisk

02\_tidhorisont: langsiktig

## Instans SI\_008

**Seeds:** strategisk | langsiktig | makt | sårbarhet | utforsk

Agentisk KI bør behandles som en **ny maktfaktor i produksjon**, ikke som et vanlig automasjonsverktøy. Den største risikoen er ikke “feil svar”, men **akkumulert handlekraft** i et komplekst landskap der små feil kan forplante seg. Derfor bør dere starte med et prinsipp: **ingen agent skal ha direkte produksjonsmakt uten en eksplisitt, maskinlesbar begrensning av tid, rom og effekt.**

### Arkitektur og kontroll

Bygg et **guardrailed execution layer** mellom agent og systemer:

- Agenten foreslår; en policy-motor avgjør; en isolert executor utfører.
- Del handlinger i 4 klasser: **lese, foreslå, transaksjonell endring, irreversibel endring**. Bare de to første bør være åpne i pilot.
- Bruk **runtime-isolasjon**: separate service accounts, nettverkssegmentering, egress-kontroll, og per-oppgave sandkasse.
- Modellér execution graph som en **DAG med attestering per node**: prompt, beslutning, policy-avgjørelse, API-kall, sideeffekt, kompensasjon. Uten dette mister dere sporbarhet.

### IAM og tilgang

For autonome agenter er klassisk RBAC for grovt. Bruk **dynamisk, kontekstuell minste privilegium**:

- Ephemeral credentials med TTL på **5-30 minutter**
- Scope per system, per dataset, per operasjon
- Delegert autorisasjon med “just-in-time” godkjenning for høy-risiko handlinger



- Nøkkelrotasjon automatisk og hyppig; ingen langlivede secrets i prompt eller runtime

## Integrasjoner og API-risiko

Legacy-API-er uten idempotens er **sårbarhetsmultiplikatorer**. Anta dobbelkall, partial failure og skjulte sideeffekter.

- Sett **rate limits** per agent og per mål-API
- Bruk circuit breakers og “dry-run” der det finnes
- Krev idempotency keys på alle operasjoner som støtter det
- For ikke-deterministiske systemer: tillat bare **read-only** i første fase

## Logging, revisjon og hendelser

Bygg en audit-pipeline som fanger:

- prompt, system prompt, tool call, policy decision, output, sideeffekt, rollback
- tidsstempler og korrelasjons-ID per agent-oppgave
- lagring i append-only format, minst 90 dager

Ved avvik: **auto-isolér agenten**, frys tokens, stopp egress, og trigge menneskelig review.

## Blindsoner

Undervurdert risiko er **latent coupling**: agenten kan treffe flere “trygge” systemer som samlet gir utrygg effekt. Worst case må modelleres som:

1. feil input
2. legitim policy-godkjent handling
3. uventet kjedereaksjon
4. irreversibel endring

## Neste steg

1. Velg **én lavrisiko brukscase** og begrens agenten til read-only + forslag.
2. Definér en **policy-matrise** for handlingstyper, systemer og maksimal skade.
3. Etabler minimal audit og token-styring før noe går i produksjon.