



id: SI\_007

model: anthropic/claude-sonnet-4.6

seeds:

04\_risiko: sårbarhet

03\_relasjon: tillit

05\_handling: aksepter

01\_perspektiv: strategisk

02\_tidhorisont: langsiktig

## Instans SI\_007

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

### Tillit som arkitekturprinsipp — ikke som mål

Det strategisk viktigste grepet er å **slutte å behandle kontroll som en teknisk sjekkliste** og begynne å designe for tillitsbygging over tid. Agentisk KI i offentlig sektor er ikke et infrastrukturproblem — det er et tillitsregnskapsproblem.

### Aksepter sårbarhet som utgangspunkt

Mainstream-rådet er å løse alle sikkerhetsgap før pilot. Det er feil. Med fragmentert logging, svak IAM og eldre fagsystemer uten integrasjonsstøtte, **er full kontroll en illusjon i 6-9 måneder**. Aksepter dette eksplisitt.

Konsekvensen er ikke å utsette — det er å designe for kontrollert synlig feil, ikke for feilfrihet.

#### Konkret:

- Definer maksimalt 3 handlingsdomener for pilot (f.eks. kun lesing + én skrive-operasjon i isolert system)
- Bruk “circuit breaker”-mønster: agent stoppes automatisk ved >5 uventede API-responser per time
- Logg alt til én sentralisert, append-only tjeneste (f.eks. Azure Monitor + immutable storage) — ikke perfekt, men god nok



---

## Langsiktig IAM-strategi starter nå

RBAC er utilstrekkelig for autonome agenter. Men i stedet for å vente på finmasket policy-motor, **implementer en proxy-lag-arkitektur**:

- Alle agentkall går gjennom én kontrollert gateway (f.eks. Kong eller Azure APIM)
- Gatewayen håndhever: rate-limiting (maks 100 kall/minutt per agent), scope-begrensning, og nøkkelrotasjon hver 24. time via Vault eller Azure Key Vault
- Dette gir sporbarhet nå, og legger grunnlag for OPA-basert policy-motor i fase 2

---

## Den undervurderte risikoen

**Eskaleringsblindhet:** Agenter som ikke kan eskalere korrekt, improviserer. I hybride systemlandskap betyr dette at en agent kan forsøke alternative API-paths og utilsiktet eksponere data. Design eksplisitt **dead-end-tilstander** — agenten skal feile høylytt, ikke stille.

---

## Neste steg

1. **Uke 1-2:** Kartlegg og frys agentens handlingsrom til tre veldefinerte domener — skriv dette som en formell *capability boundary*-spesifikasjon
2. **Uke 3-4:** Implementer proxy-gateway med logging til append-only lager og automatisk circuit breaker
3. **Måned 2:** Gjennomfør en “red team”-øvelse internt der én person forsøker å misbruke agenten — bruk funnene til å justere boundary-spesifikasjonen, ikke til å stoppe piloten