



id: SI\_006

model: anthropic/claude-opus-4.6

seeds:

04\_risiko: sårbarhet

03\_relasjon: tillit

05\_handling: implementer

01\_perspektiv: strategisk

02\_tidhorisont: langsiktig

## Instans SI\_006

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

### Kjerneanalyse: Tillit er arkitekturen

Dissens først: Mainstream-tilnærmingen er å starte med sandboxing og tekniske kontroller. Mitt perspektiv er at den primære **sårbarheten** ikke er teknisk — den er at organisasjonen mangler en tillitsmodell for ikke-menneskelige aktører. Uten denne kollapser alle tekniske tiltak til ad hoc-brannslukking.

### Strategisk tillitsarkitektur

**Graduated Trust Model (GTM)** — implementer agenter som nyansatte, ikke som systemer:

- **Nivå 0 (uke 1-8):** Kun lesing. Agenten foreslår, menneske utfører. Alle handlinger logges til dedikert audit-stream.
- **Nivå 1 (måned 3-4):** Reversible skriveoperasjoner med automatisk rollback-vindu på 15 minutter.
- **Nivå 2 (måned 5-7):** Utvidede rettigheter på spesifikke domener, men med hard rate-limiting (maks 50 muterende kall/time).
- **Nivå 3 (måned 8+):** Autonomi innenfor definerte policy-korridorer.

Hvert nivåskifte krever dokumentert tillitsevaluering — ikke bare teknisk test, men gjennomgang av faktiske handlingsmønstre.



---

## De undervurderte sårbarhetene

1. **Eldre fagsystemer mangler idempotens.** Agenten kaller et SOAP-endepunkt to ganger ved timeout — og oppretter to vedtak. Implementer en **command-gateway** (f.eks. basert på Temporal.io) som garanterer exactly-once-semantikk foran alle legacy-systemer.
2. **Kontekstkollaps i IAM.** RBAC gir agenten samme rettigheter uavhengig av hvorfor den handler. Implementer en lettvekts policy-motor (Cedar/OPA) som evaluerer intensjon + kontekst, ikke bare rolle. Eksempel: agenten har skrive tilgang til saksbehandlingssystem, men policy-motoren blokkerer hvis handlingen trigges av eksternt API-respons uten validert kilde.
3. **Informasjonslekkasje via prompt/kontekst.** Agenten kan sende sensitive persondata som kontekst til eksterne LLM-API-er. Implementer en **egress-filter** som scanner utgående kall for fødselsnumre, helseopplysninger og gradert informasjon — før de forlater nettverket.

---

## Langsiktig kontrollmodell

Ikke bygg for piloten — bygg for **år 3**, der dere har 15 agenter. Det betyr:

- Sentralisert **agent-register** med livsløpsstyring (opprettelse, rettighetsendring, deaktivering)
- Event-sourced audit-log som er uavhengig av agentene selv — de kan ikke skrive til sin egen logg
- Dedikert **eskaleringskanal** med SLA: agenten pauser og varsler innen 200ms ved policy-brudd

---

## Konkrete neste steg

1. **Uke 1-2:** Implementer OPA/Cedar som policy-gateway foran de 3 mest kritiske fagsystemene. Start med deny-by-default og whitelist spesifikke handlinger. Konkret leveranse: policy-as-code i Git.
2. **Uke 3-4:** Bygg command-gateway med idempotens-lag og 15-minutters rollback-vindu. Test mot legacy-systemet med høyest risiko (typisk økonomi/vedtak).
3. **Uke 5-6:** Etabler agent-registeret og den uavhengige audit-loggen. Definer GTM-nivåene formelt og knytt dem til målbare tillitskriterier — ikke tidsplan alene.