



id: SI_003

model: anthropic/claude-sonnet-4.6

seeds:

04_risiko: mulighet

03_relasjon: makt

05_handling: implementer

01_perspektiv: bruker

02_tidhorisont: syklisk

Instans SI_003

Seeds: bruker | syklisk | makt | mulighet | implementer

Perspektiv: Brukeren som syklisk maktnode

Mainstream-tilnærming fokuserer på teknisk arkitektur. Min dissens: **brukeren** er den kritiske blindsonen – ikke systemet.

Agentisk KI redistribuerer makt syklisk: agenten handler → systemet endres → brukeren påvirkes → brukeren justerer agenten → gjenta. Denne syklusen er ukontrollert uten eksplisitt maktmodellering.

Konkret risikoanalyse

Maktkonsentrasjon i agentlaget er underkommunisert. En agent med RBAC-tilgang til 4-5 fagsystemer har akkumulert makt ingen enkeltbruker ville fått. Eksempel: agent som kan lese HR-data, trigge utbetalinger og kalle eksternt API – kombinasjonen er farligere enn delene.

Syklisk forsterkning: Svak datakvalitet → agent tar feil beslutning → data forverres → neste iterasjon amplifiser feilen. Uten transaksjonelle garantier i legacy-monolitter kan 3-4 sykluser skape irreversibel skade innen minutter.



Implementeringsanbefalinger (realistisk for 6-9 mnd)

Ikke start med OPA/Styra full-deploy. Det er for ambisiøst. Implementer i stedet:

1. **Agent-spesifikke service accounts** med 15-minutters ephemeral tokens (HashiCorp Vault eller Azure Managed Identity). Kostnad: lav. Effekt: umiddelbar sporbarhet.
2. **Maksimalt 3 handlingsklasser per pilot-agent** – les, skriv, kall-eksternt. Ingen agent krysser alle tre i v1. Dette er en mulighet til å bevise konseptet uten full risiko.
3. **Syklus-bryter i kode:** Hardkodet maksgrense på 10 iterasjoner per agentjobb + mandatory human-in-the-loop ved iterasjon 7. Implementeres på 2-3 dager.
4. **Brukerdrevet revisjonslogg** – ikke bare teknisk audit. Brukeren som eier datadomenet skal motta daglig digest av agenthandlinger i sitt system. Dette skaper distribuert overvåkning uten sentralisert SIEM.

Worst-case-modellering

Latent coupling er den reelle trusselen: to systemer som tilsynelatende er uavhengige deler faktisk en databaserad eller event-queue. Agenten ser ikke dette. Kartlegg eksplisitt hvilke 5-10 systempar som har skjulte avhengigheter før pilot.

Neste steg

1. **Uke 1-2:** Kartlegg alle systempar med latent coupling – intervju systemforvaltere, ikke bare arkitekter
2. **Uke 3:** Implementer ephemeral credentials + iterasjonsbryter som MVP-sikkerhetslag
3. **Uke 4:** Definer bruker-eierskap per datadomene og design digest-varsling før agenten berører systemet