



id: SI\_006

model: mistralai/mistral-large-2512

seeds:

04\_risiko: mulighet

03\_relasjon: makt

05\_handling: implementer

01\_perspektiv: strategisk

02\_tidhorisont: langsiktig

## Instans SI\_006

**Seeds:** strategisk (maktbalanse) | langsiktig (systemisk motstandskraft) | makt (kontrollmekanismer) | mulighet (autonomi som forsterker) | implementer (pragmatisk skalerbarhet)

### Strategisk maktfordeling: Agentisk KI som delegert autoritet

Agentisk KI er ikke et verktøy, men en ny aktør i maktstrukturen. Den utfordrer RBAC-modellen ved å operere som en “proxy-bruker” med dynamiske rettigheter. **Løsningen ligger i å designe en tredelt kontrollmodell** som balanserer autonomi, begrensninger og etterprøvbarehet:

#### 1. Hierarkisk policy-sandkasse

- Implementer en “policy-pyramide” med tre lag:
  - **Global policy** (eks: “ingen agenter kan slette data i fagsystem X”).
  - **Domene-spesifikke policies** (eks: “økonomi-agenter kan kun opprette fakturaer under 50k NOK”).
  - **Ad-hoc policies** (dynamiske begrensninger per oppgave, f.eks. “denne agenten kan kun aksessere API Y i 2 timer”).
- Verktøy: Bruk **Open Policy Agent (OPA)** med tilpassede Rego-regler for å håndheve policies på tvers av systemer. Eksempel: En agent som prøver å endre en pasientjournal i et eldre system, blir stoppet av OPA hvis den mangler en “medisinsk autorisasjons-token”.



## 2. Maktbegrensning gjennom “sikkerhetsperimeter per agent”

- Strategisk grep: Behandle hver agent som en midlertidig mikrotjeneste med egen identitet, isolert nettverkssegment (via Kubernetes Network Policies) og dedikerte IAM-roller.
- Konkret: Bruk **SPIFFE/SPIRE** for å utstede kortvarige identiteter til agenter (levealder: 1–24 timer). Kombiner med **Vault** for dynamisk nøkkelrotasjon.
- Risiko: Eldre systemer uten API-gateways blir blindsoner. Løsning: Proxy alle agent-kall gjennom en **API-mediator** (eks: Kong eller Apigee) som logger, validerer og rate-begrenser.

## 3. Langsiktig motstandskraft: “Reversible autonomy”

- Mulighet: Agentene må kunne operere autonomt, men alle handlinger må være atomisk reversible.
- Implementer:
  - **Transaksjonslogg med “undo-stack”**: Hver agent-handling logges som en transaksjon med en tilhørende rollback-funksjon (eks: for en fakturaendring, lagre både gammel og ny verdi).
  - **Eskaleringsmekanismer**: Hvis en agent gjentatte ganger trigger policies, fryses den og krever menneskelig godkjenning (bruk **Prometheus + Alertmanager** for overvåking).
  - **Eksempel**: En agent som prøver å sende eposter til eksterne mottakere, blir stoppet av en policy som krever at alle eksterne kall først godkjennes via en **human-in-the-loop**-tjeneste (eks: Slack-approval).

---

## Blindsoner og undervurderte risikoer

- **Integrasjonsrisiko**: Eldre systemer har ofte implisitte avhengigheter (eks: en agent som oppdaterer et register, uten å vite at det trigger en bakgrunnsprosess i et annet system). Løsning: Kjør en **avhengighetskartlegging** (eks: med **Dependency-Track**) før pilot.
  - **Dataeksponering**: Agenter kan uforvarende sende sensitiv data til eksterne API-er (eks: en chatbot som inkluderer personnummer i et logginnslag). Løsning: Bruk **data masking** (eks: **Presidio** fra Microsoft) på alle utgående kall.
  - **Feilhandlinger**: Agenter kan eskalere feil ved å gjenta mislykkede handlinger (eks: en agent som prøver å slette en fil 100 ganger i minuttet). Løsning: Implementer **eksponentiell backoff** + maks 3 forsøk før låsing.
-



## Neste steg (konkret, handlingsrettet)

1. **Bygg en “policy-prototyp” i OPA** med 3–5 kritiske regler (f.eks. “ingen skriveoperasjoner i fagsystem X uten godkjenning”). Test mot en dummy-agent i en isolert staging-miljø.
2. **Kjør en “red team”-øvelse** med en ekstern sikkerhetspartner (eks: **mnemonic** eller **Proact**) for å identifisere blindsoner i agentenes handlingsrom. Fokus: Hvordan kan en agent omgå policies?
3. **Implementer SPIFFE/SPIRE + Vault** for én pilot-agent (f.eks. en “faktura-automatiseringsagent”). Mål: Få på plass dynamisk identitetshåndtering innen 4 uker.

---

**Dissens:** Hovedstrømmen fokuserer på tekniske kontrollmekanismer, men den reelle risikoen er organisatorisk – hvem eier ansvaret når en agent gjør en feil? Løsning: Opprett en **Agent Governance Board** med representanter fra IT-sikkerhet, juridisk og faglederne, med mandat til å revidere policies kvartalsvis.