



id: SI\_001

model: mistralai/mistral-large-2512

seeds:

04\_risiko: usikkerhet

03\_relasjon: makt

05\_handling: implementer

01\_perspektiv: strategisk

02\_tidhorisont: langsiktig

## Instans SI\_001

---

**Seeds:** strategisk (maktfordeling), langsiktig (arkitektur som overlever piloten), makt (hvem kontrollerer agentene?), usikkerhet (hva vet vi ikke?), implementer (hva kan vi faktisk bygge nå?)

---

### Maktens arkitektur: Hvem eier agentens beslutninger?

Agentisk KI er ikke et teknisk problem, men et maktspørsmål. Når en agent kan endre data, initiere prosesser og samhandle med eksterne tjenester, flytter dere makt fra mennesker til kode – uten en klar eier. **Strategisk feil:** Å designe kontrollmekanismer etter piloten, når agentene allerede har etablert “faktiske” handlingsmønstre. Løsningen er å invertere maktfordelingen: La mennesker definere agentens handlingsrom før den får tilgang, ikke som en ettertanke.

#### Konkret:

- **Policy-sandkasser:** Bygg en “negativ liste” (hva agenten aldri får gjøre) før en “positiv liste” (hva den kan gjøre). Eksempel: Agenten kan aldri slette data i eldre fagsystemer, men kan lese og foreslå endringer via en menneskelig godkjenningsloop.
  - **Maktens kilde:** IAM-plattformen må utvides med tidsbegrensede tokens for agenter (f.eks. 15-minutters levetid) og kontekstbasert autorisasjon (f.eks. “Agent X kan kun endre data i system Y mellom 08-16 på hverdager”).
  - **Eierskap:** Opprett en Agent Governance Board (AGB) med representanter fra sikkerhet, juridisk, og fagsystemeiere. AGB må godkjenne alle nye agent-handlinger før de implementeres.
-



## Langtidsrisiko: Blindsoner i usikkerheten

**Usikkerhet #1:** Agentene vil oppdage uforutsette dataflyter. Eksempel: En agent som skal optimalisere saksbehandling, finner en “bakdør” i et eldre system som lar den hoppe over et godkjenningssteg. **Løsning:** Implementer dataflyt-simulering i piloten – la agenten kjøre i en kopi av produksjonsmiljøet i 3 måneder før den får tilgang til ekte data.

**Usikkerhet #2:** Feilhandlinger vil eskalere på uventede måter. Eksempel: En agent som skal oppdatere en kundeprofil, trigger en kaskade av automatiske varsler i et annet system, som igjen utløser en manuell prosess. **Løsning:** Bygg en escalation matrix med tre nivåer:

1. **Automatisk rollback** (f.eks. ved rate-limiting-brudd).
2. **Menneskelig intervensjon** (f.eks. ved uventede API-responser).
3. **Full stopp** (f.eks. ved sensitiv dataeksponering).

**Usikkerhet #3:** Logging vil bli ubrukelig. Agentene vil generere for mye data (hver beslutning logges) eller for lite (kun sluttresultatet logges). **Løsning:** Adopter differensiert logging:

- **Strategiske handlinger** (f.eks. dataendringer): Full revisjonsspor med tidstempel, agent-ID, og input/output.
- **Operasjonelle handlinger** (f.eks. API-kall): Kun metadata (tid, system, volum).

---

## Implementering: Hva kan vi faktisk bygge på 6-9 måneder?

1. **Pilotens rammer:** Begrens agentene til én fagsystemintegrasjon (f.eks. saksbehandling) og én ekstern API (f.eks. folkeregisteret). Bruk piloten til å teste:
  - **Policy-motoren:** Bygg en enkel regelbasert motor (f.eks. Open Policy Agent) som kan blokkere handlinger basert på tid, system, eller data.
  - **Sandboxing:** Kjør agentene i isolerte containere med ingen direkte tilgang til produksjonsdata – kun via proxy-APIer som filtrerer sensitiv informasjon.
2. **Kontrollmodell:** Kombiner tre lag:
  - **Lag 1 (Autonomi):** Agenten tar beslutninger innenfor definerte grenser.
  - **Lag 2 (Sikkerhet):** Policy-motoren og IAM-plattformen blokkerer uønskede handlinger.
  - **Lag 3 (Etterprøvbarehet):** Logging og revisjonsspor lagres i et sentralt system (f.eks. Elasticsearch) med søkbar metadata.



### 3. Blindsoner å adressere umiddelbart:

- **Eldre systemer:** Identifiser de 3 mest kritiske eldre systemene og bygg wrapper-APler som begrenser hva agentene kan gjøre (f.eks. kun lese-operasjoner).
- **Nøkkelrotasjon:** Bruk en secrets manager (f.eks. HashiCorp Vault) til å automatisk rotere tokens for agenter hver 24. time.

---

### Neste steg (konkrete, prioriterte)

1. **Maktkartlegging:** Opprett Agent Governance Board innen 2 uker og definer negative listen for agent-handlinger (hva de aldri får gjøre).
2. **Dataflyt-simulering:** Sett opp et kopi-miljø av produksjonen og la en dummy-agent kjøre i 1 måned for å identifisere uventede interaksjoner.
3. **Policy-prototype:** Implementer Open Policy Agent med 5–10 regler som blokkerer handlinger basert på tid, system, og data (f.eks. "Ingen endringer i økonomisystemet etter kl. 15").