



Problemstilling: Agentisk KI - avansert styring, kontroll og...

1. Problemstilling: Agentisk KI - avansert styring, kontroll og risiko i komplekse systemlandskap

Vi er en mellomstor offentlig virksomhet som vurderer å innføre agentisk KI som kan utføre autonome handlinger på tvers av interne systemer, API-lag, dataplattformer og eksterne tjenester. Agentene vil kunne initiere prosesser, endre data, trigge hendelser, utføre skript og samhandle med tredjeparts-API-er uten eksplisitt menneskelig godkjenning for hvert steg.

Teknisk kontekst og begrensninger

- Systemlandskapet består av en blanding av moderne mikrotjenester, eldre monolitter og fagsystemer uten deterministiske API-kontrakter.
- IAM-plattformen støtter RBAC og delvis ABAC, men mangler finmasket policy-evaluering for autonome prosesser.
- Vi har ikke en sentralisert policy-motor (f.eks. OPA) eller en helhetlig Zero Trust-arkitektur.
- Logging er distribuert, uten konsolidert event-sourcing eller revisjons-pipeline.
- Det finnes ingen eksisterende mekanismer for “safe execution environments” for KI-drevne agenter.
- Ressurser til sikkerhetsarkitektur og DevSecOps er begrenset, og pilotvinduet er 6–9 måneder.

Uavklarte tekniske beslutninger

- Hvordan definere og håndheve autonomi-grenser for agentene (tillatte handlinger, systemer, datasett, API-kall, endringsnivå).
- Hvordan etablere policy-sandkasser med runtime-evaluering (f.eks. OPA/Styra, Kyverno, Gatekeeper) som stopper uønskede handlinger før de skjer.
- Hvordan sikre autentisering og autorisasjon for KI-agenter (service accounts, ephemeral credentials, nøkkelrotasjon, token-scoping).
- Hvordan designe audit-pipeline som fanger opp agentbeslutninger, mellomsteg, prompts, API-kall og sideeffekter.
- Hvordan implementere rollback-mekanismer, transaksjonelle garantier og “compensating actions” når agenter gjør feil.
- Hvordan sikre data-minimering, konfidensialitet og kontroll når agenter gjør eksterne API-kall.



- Hvordan overvåke og stoppe runaway agents, loops, eskaleringer og uforutsette kjedereaksjoner.
- Hvordan modellere risiko for emergent behavior i systemer som ikke er designet for autonome aktører.

Spørsmål: Arkitektur og kontroll

- Hvilke arkitekturprinsipper bør ligge til grunn for å kontrollere autonome KI-agenter i et heterogent systemlandskap med både moderne og legacy-komponenter.
- Hvordan bør vi kombinere Zero Trust, policy-as-code, least privilege og runtime-isolasjon for å hindre uønskede handlinger.
- Hvordan bør vi modellere agentenes “execution graph” for å sikre sporbarhet og kontroll.

IAM og tilgangsstyring

- Hvordan bør vi designe en IAM-modell for autonome agenter som krever dynamiske, kontekstuelle og tidsbegrensede tilganger.
- Hvordan bør vi håndtere token-scoping, nøkkelrotasjon, ephemeral credentials og delegert autorisasjon.

Integrasjoner og API-sikkerhet

- Hvilke risikoer oppstår når agenter får tilgang til API-er som ikke er designet for autonom interaksjon, inkludert manglende idempotens, uforutsigbare sideeffekter og svak kontraktshåndheving.
- Hvordan bør vi implementere rate-limiting, throttling, circuit breakers og observability for agent-initierte kall.

Logging, revisjon og hendelseshåndtering

- Hvordan bør vi bygge en audit-pipeline som fanger opp agentens beslutningsgrunnlag, mellomsteg, prompts, handlinger og sideeffekter.
- Hvordan bør vi designe hendelseshåndtering for feilhandling, eskalering og automatisk isolering av agenten.

Blindsoner og risiko

- Hvilke tekniske blindsoner oppstår når autonome agenter opererer i systemer med svak datakvalitet, uforutsigbare API-kontrakter eller manglende transaksjonelle garantier.
- Hvilke risikoer knyttet til emergent behavior, uforutsette kjedereaksjoner og “latent coupling” undervurderes typisk.
- Hvordan bør vi modellere worst-case-scenarier når agenten får tilgang til systemer med høy integritetsrisiko