



id: SI_003

model: qwen/qwen3.6-plus

seeds:

04_risiko: resiliens

03_relasjon: tillit

05_handling: implementer

01_perspektiv: strategisk

02_tidhorisont: langsiktig

Instans SI_003

Seeds: strategisk | langsiktig | tillit | resiliens | implementer

Din utfordring er ikke mangel på verktøy, men fraværet av en **strategisk** arkitektur. Å chase hver ny KI-plattform er en tapt kamp. Bygg i stedet en selvforsterkende protokoll der agentene ikke bare samler, men validerer og strukturerer.

Langsiktig kunnskap krever at du skiller ingest fra prosessering. Bruk en Redis-kø som buffer mot API-stopp eller rate-limits. Dette skaper **resiliens**: hvis OpenRouter eller X-API svikter, henger ikke hele svermen. Agentene kjører asynkront, kvitterer og gjenopptar uten tap av kontekst.

For å unngå manuelt lim mellom e-post og Obsidian, deploy en lett admin-side (Next.js + Supabase). Agentene pusher direkte til databasen via REST. Obsidian syncer automatisk via Obsidian Git. E-post til raymond@sverm.ai blir en ren output for commute-lytting, aldri en mellomlagring.

Implementer denne 4-trinns sverm-pipen:

1. Horizon Scout: Skanner RSS, YouTube API og X. Identifiserer nye noder via engagement-grafer, ikke bare nøkkelord.
2. Signal Filter: Vurderer relevans mot din KB-topologi. Kaster ~80 % av støyen. Gir en konfidensscore.
3. Synthesis Engine: Kjører lokal Whisper + OpenRouter-llm. Transkriberer og strukturerer etter fast skjema (Kilde, Kernepåstand, Implikasjon for orkestrering, Lenke).
4. KB Weaver: Pusher til Supabase, tagger, og trigger daglig digest.

Tillit bygges gjennom konsensus-kryssvalidering: La to uavhengige modeller vurdere samme kilde før lagring. Hvis avviket >30 %, flagges kilden for manuell review. Dette eliminerer KI-hallusinasjoner i din vault.



Dissens: Slutt å optimalisere for dekning. Optimaliser for dybde. En sverm på 100 instanser som jakter nyheter, er verdiløs uten en felles ontologi. Definer 5 kjerne-dimensjoner for ditt felt (f.eks. orkestreringsmønstre, feiltoleranse, kostnad/eval, sikkerhet, agent-koordinering) og la agentene kun indexere innhold som eksplisitt mappes til disse.

Neste steg:

1. Deploy Redis + Supabase + Next.js på din server. Konfigurer OpenRouter med fallback-ruting til minst 3 modeller.
2. Skriv 3 Python-agenter (Scout, Filter, Weaver) som kommuniserer via køer, ikke direkte kall.
3. Test konsensus-valideringen på 50 historiske kilder. Juster terskler til false-positive <5 % før fullskaladrift.