



id: SI\_001

model: qwen/qwen3.6-plus

seeds:

04\_risiko: mulighet

03\_relasjon: makt

05\_handling: vent

01\_perspektiv: operasjonell

02\_tidhorisont: kortsiktig

## Instans SI\_001

**Seeds:** operasjonell | kortsiktig | makt | mulighet | vent

Din utfordring er ikke teknisk, den er arkitektonisk. Mainstream svarer med nye rammeverk; vi svarer med en **operasjonell** sløyfe som kjøres på din infrastruktur. Kutt støyen. Bygg et 3-agent swarm på serveren din (Ubuntu/Python), styr via OpenRouter:

1. **Kilder:** yt-dlp + RSS + X via offisielt API. Kjør whisper.cpp (local, q4\_K\_M) for transkripsjon.
2. **Filter:** En llama-3.1-70b scorer innhold (0-100) mot dine nøkkelord. Kun >75 går videre.
3. **KB & Levering:** Agent skriver Markdown direkte til ~/ObsidianVault/swarm-intel/. En FastAPI -dashboard gir admin-oversikt. Postmark sender digest til raymond@sverm.ai. Null manuell overføring.

**Kortsiktig** prioritering: 14-dagers sprint. Bruk LangGraph for tilstand, men hold agentene deterministiske. Kjør 04:00–06:00. Rapport før frokost.

**Makt** ligger i datakontroll. Unngå sky-avhengige orkestratorer. Din server er din suverenitet. Bruk OpenRouter kun som modell-router. Lag lokal SQLite med vector extensions for søkbar historikk.

**Vent** er nøkkelen. Ikke jag releases. La systemet samle signal i 3 uker. Juster terskler manuelt annenhver uke. Mainstream hypen om “autonome agenter” er støy; din fordel er kontrollert asynkronitet.

**Mulighet:** Din nisje er ikke orkestrering, men signal-resiliens. Dokumenter sløyfen. Det du bygger internt, kan lisensieres til team som kveles i overlast.

Neste steg:

1. Deploy yt-dlp + whisper.cpp + OpenRouter-summarizer i dag. Test 5 kanaler.



2. Opprett `docker-compose` med agent-swarm, SQLite og FastAPI. Kjør første filsystem-sync til Obsidian.
3. Sett 10-dagers “vent-periode” uten logikk-endringer. Mål: <15 min daglig vedlikehold, >80% relevant dekning.