

Live data for people negotiating fuel prices

One of Europe's largest energy companies (Germany / Netherlands)

THE STAKES

Operators and traders negotiated input prices — gas, hard fuels, biomass — for power generation units using a tool with hardcoded values and long waits on direct data-warehouse queries. When the number on screen feeds a live commercial negotiation, stale data is money left on the table with every deal.

WHAT WE DID

The replacement began as a small internal Flask/Dash tool. Three years later it was the daily instrument of a large trading and operations team:

- **Live prices and live calculations** instead of hardcoded values
- **Answers in seconds:** replaced direct Snowflake dependence with a data agent pulling only the needed slices into a local PostgreSQL, transformed with Polars — interactive speed instead of warehouse waits
- Rebuilt Flask into **FastAPI + FastStream**, sync to async; split the monolith into services; unified frontend and backend on shared Pydantic contracts so the two cannot drift apart
- Ran it to production and kept it there: Kubernetes, Helm, CI/CD, load tests — plus the monitoring, alerting and traceability layer, built as part of the delivery

OUTCOMES

- User waits cut from long stalls to interactive speed; negotiation calculations moved from stale hardcoded inputs to live data — in commodity negotiation, data freshness is directly a commercial position.
- A small internal build became a large team's daily instrument.
- Three-year retained engagement — energy majors don't keep contractors for 33 months out of politeness.

STACK

Python · FastAPI · FastStream · Dash · Polars · Snowflake · PostgreSQL · Azure · Kubernetes · Helm

This is the engineering standard behind Alpine's fixed-price delivery audit: findings with file-and-line evidence, an implementation playbook, and an execution-ready backlog your team can run with. References on request.

Alpine LTD · contact@alpine-engineering.eu · alpine-engineering.eu · linkedin.com/company/138193958