clusy.io: Design Document

Eldar Hasanov[1], Ju Lin[2], Raffaelli Ispas[3], Thomas Carruthers[4]

I. Problem Focus

Problem: Compute-Energy Convergence in a DEG World

AI training and inference demand is rising faster than affordable, low-carbon compute supply. Providers now span conventional cloud GPUs, private enterprise clusters, emerging TPU fleets, and decentralised networks. At the same time, energy pricing and carbon intensity vary by region and time, especially in a distributed energy grid (DEG). Users therefore face a multi-objective decision problem: selecting *where* to run workloads and *when* to run them to balance cost, speed, and environmental impact. Existing tools largely optimise within a single provider, leaving cross-provider routing and carbon-aware scheduling under-served.

II. Solution Overview

clusy.io is an AI-agent orchestration platform that intelligently routes and schedules ML/AI workloads across **heterogeneous compute providers**, including cloud GPU clusters, on-prem enterprise clusters, **TPU pods**, and decentralised GPU networks such as **CUDOS**, optimising for cost, performance, CO_2 intensity, and timing. Beyond placement, clusy.io introduces **carbon- and price-aware scheduling**: it surfaces windows where workloads will be cheaper or lower-carbon (e.g., off-peak electricity or peak-renewable periods) and enables deferred execution.

The clusy.io agent interacts with all providers through **Beckn-compliant discovery and order flows**, retrieving GPU/TPU wattage estimates, local and regional CO_2 intensity, and provider pricings. It recommends multiple pathways—cheapest, fastest, greenest, or optimal-time—and can automatically defer jobs to align with energy-flexible periods.

In enterprise deployments, clusy.io can track carbon-optimised compute usage and integrate with **carbon-credit** and **flexibility-market** incentives. In a DEG world, clusy.io serves as a unified layer bridging distributed compute markets with grid-aware optimisation.

III. Technical Architecture

Fig. 1 shows the end-to-end system architecture. clusy.io implements a Beckn Application Platform (BAP) front end and a backend orchestration stack:

- **User Interface (UI):** Collects a natural-language workload request and an optional dataset upload.
- **Dataset Parser:** Extracts a dataset-structure JSON describing schema, size, modality, and storage location.
- **Task Parser LLM:** Converts user intent into task JSON (model type, training regime, resource needs, constraints).
- **Intent Builder LLM:** Fuses task JSON, dataset JSON, and the retrieved Beckn schema into a Beckn-compliant request.
- **Intent Supervisor LLM:** Validates the request; on failure, it emits an error-aware repair prompt. After a second failure, the UI requests clarification from the user.
- **Cluster Quoting Tool:** Queries providers for pricing, availability, hardware specs, and carbon signals.
- **Aggregation Service:** Normalises and aggregates quotes and grid/market signals into comparable metrics.
- **Quote Formulator LLM:** Packages quotes with derived signals (cost, latency, CO_2 intensity, confidence).
- **Ranking Supervisor LLM:** Validates ranking inputs and outputs; if valid, returns final ranked options to the UI.

IV. Agent Workflow

Fig. 1 details the agentic loop:

1) **Parse inputs:** Generate dataset-structure JSON and task JSON from the upload and natural-language request.
2) **Construct intent:** Build a Beckn request JSON constrained by the provider schema.
3) **Supervise and repair:** Validate intent; repair once automatically, otherwise request user clarification.
4) **Quote retrieval:** Fetch multi-provider quotes and energy/carbon signals via Beckn flows.
5) **Aggregate metrics:** Compute normalised cost, expected runtime, carbon intensity, and timing feasibility.
6) **Rank options:** Produce ranked pathways (cheapest/fastest/greenest/optimal-time).
7) **Return results:** Present ranked options and recommended execution windows; optionally schedule deferred jobs.

V. Business Model & Impact

clusy.io creates value for cloud GPU providers, decentralised compute networks (e.g., **CUDOS**), TPU pod operators, enterprise data centres, and organisations seeking lower-cost, lower-carbon compute. By routing and scheduling workloads across heterogeneous hardware, clusy.io improves provider utilisation while reducing cost and emissions for users.

The business model mirrors a **Skyscanner-style marketplace**:

- **Public clusters (GPU, TPU, decentralised):** clusy.io earns a **commission per routed job**.

[1] Backend/Infra Engineer, Department of Computing, Imperial College London. Email: eldar.hasanov@imperial.ac.uk. Discord: superslonik079.

[2] Agent Architect/Frontend Engineer, Department of Earth Science and Engineering, Imperial College London. Email: ju.lin25@imperial.ac.uk. Discord: thorkee.

[3] AI/ML Engineer/Energy Analyst, Department of Computing, Imperial College London. Email: raffaelli.ispas25@imperial.ac.uk. Discord: rafa8676.

[4] Energy Analyst/Business Analyst, Department of Earth Science and Engineering, Imperial College London. Email: thomas.carruthers25@imperial.ac.uk. Discord: dreadgator.

- **Private/enterprise clusters:** clusy.io provides a **licensed orchestration stack** for internal GPU/TPU environments.
- **Energy-aware enterprises:** Optional dashboards enable **carbon-optimised scheduling** and potential **carbon-credit or flexibility-market earnings**.

clusy.io scales with the expanding supply of cloud, edge, and decentralised accelerators. By shifting compute toward low-carbon, off-peak periods, it supports DEG objectives, reduces grid stress, and enables system-level sustainability gains.

Declaration

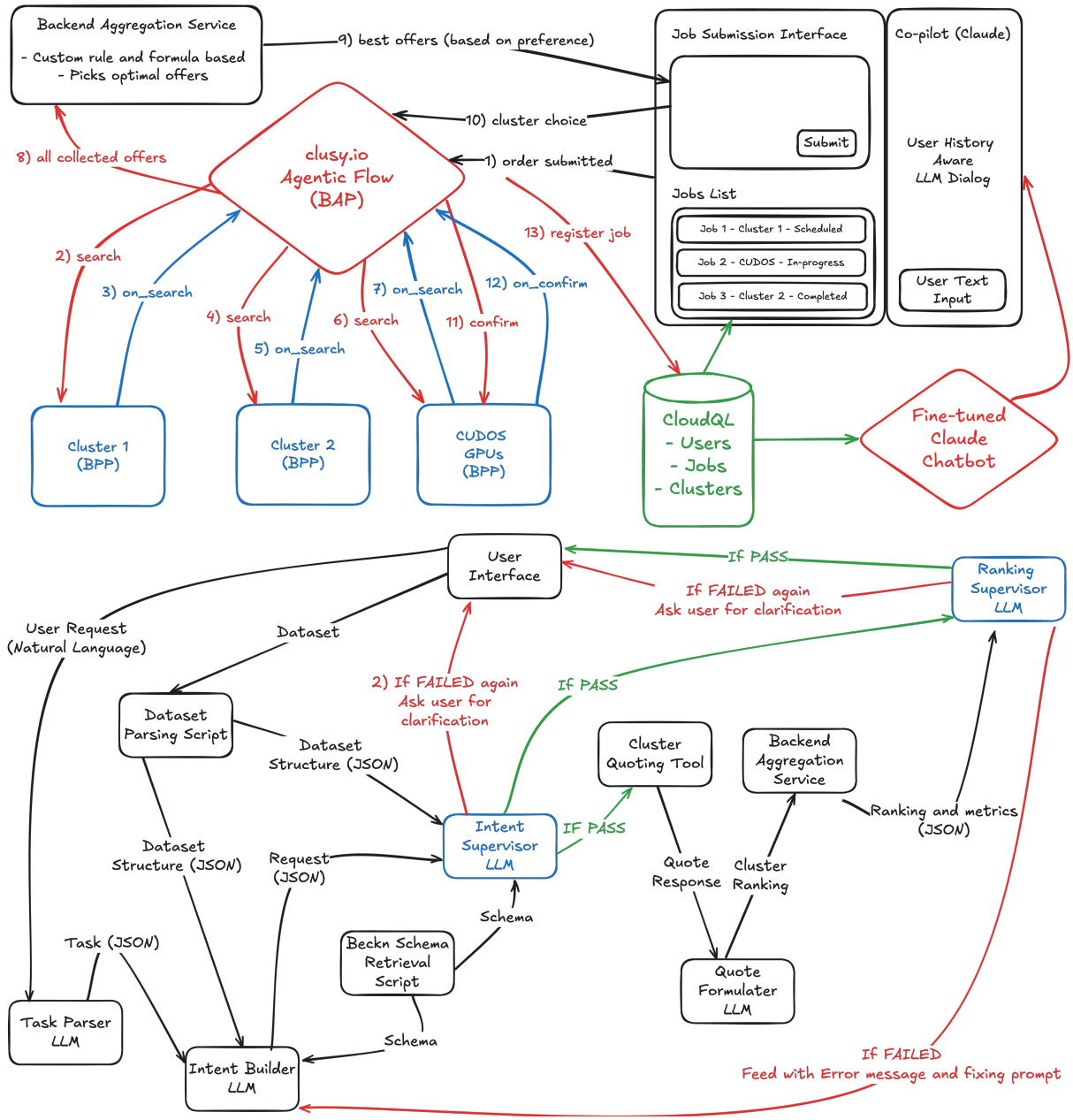

Fig. 1: clusy.io system architecture (top) and agent workflow (bottom).