

Project Summary

The client, a high-volume enterprise seller managing over 120K SKUs, faced major challenges in maintaining competitive pricing across GSA Advantage and other marketplaces. Manual price monitoring was inefficient, error-prone, and led to delayed updates, compliance risks, and lost revenue opportunities. We designed and developed an Automated Competitive Pricing Intelligence & Price-to-Beat Platform that continuously discovers, validates, and compares market prices across multiple sources, including Amazon and verified US vendors. The system applies strict rule-based matching, filters unreliable sellers, calculates true landed costs, and identifies the top competitive pricing options. It also integrates seamlessly with the client's ERP (Business Central) through automated CSV exports and S3/SFTP delivery. As a result, the client achieved real-time pricing visibility, significantly reduced manual effort, improved pricing accuracy, and strengthened their position as a lowest compliant vendor in a highly competitive marketplace.

Problem Statement

The client was managing a catalog of 120K+ SKUs without any automated pricing intelligence, relying heavily on manual checks across marketplaces like GSA Advantage. This process was slow, inconsistent, and incapable of keeping up with real-time market changes. As a result, pricing updates were delayed, leading to frequent loss of "lowest price" positioning and missed revenue opportunities. There was no standardized way to validate vendor credibility, calculate true landed costs, or ensure compliance with GSA pricing structures, increasing operational risk. Additionally, the lack of historical pricing data and automation made it impossible to scale operations as the catalog grew toward 1M SKUs. The client needed a scalable, automated solution to continuously monitor market prices, enforce compliance, and maintain competitive dominance with accurate, real-time pricing decisions.

Approach / Solution

smartData designed and built a scalable Competitive Pricing Intelligence & Price-to-Beat Platform using a distributed, API-first architecture. Instead of building complex custom scraping infrastructure, we integrated Bright Data managed APIs for Amazon, SERP, and deep vendor extraction—reducing engineering overhead while improving reliability and scalability. A rule-based matching and validation engine was implemented (instead of AI) to ensure deterministic, audit-friendly, and GSA-compliant pricing decisions. The system leverages Celery + Redis for parallel processing, enabling efficient handling of 120K–1M SKUs without performance bottlenecks. The platform includes automated SKU normalization, vendor filtering, landed cost calculation, and top-2 price selection, with seamless ERP integration (Business Central) via CSV exports and S3/SFTP delivery.

Technical Challenges

- Challenges

- ★ Scaling price discovery for 120K–1M SKUs without timeouts or sequential bottlenecks
- ★ Handling anti-bot restrictions and unreliable scraping across multiple vendor sources
- ★ Ensuring accurate product matching without AI while avoiding false positives

- How We Solved It

- ★ Implemented distributed processing using Celery + Redis with parallel API execution and batch partitioning
- ★ Replaced custom scraping with Bright Data managed APIs for stable, scalable, and compliant data extraction
- ★ Built strict rule-based validation using normalized MPN, manufacturer matching, and title verification for deterministic results

Learning

Early alignment on data formats, vendor rules, and ERP schemas is critical—delays here directly impact architecture decisions and timelines. Choosing managed data providers (like Bright Data) over custom scraping significantly reduces engineering complexity and long-term maintenance overhead. Avoid over-engineering with AI when deterministic, rule-based systems better suit compliance-driven use cases. Future pods should prioritize cost-control mechanisms (API usage, batching strategy) from day one to prevent scaling inefficiencies.

Screenshots

