

I built an analytics platform that reduced report creation turnaround from weeks to under a day.

Context

As part of our core service, we were obligated to deliver sophisticated analytics reports to clients. However, our reliance on Sigma Computing created a significant bottleneck. New report requests often took weeks to fulfill due to limited licensing and a rigid development queue. Furthermore, the outputs were static PDFs; clients had no way to filter or drill into their data. While we considered embedding Sigma directly into our portal to solve this, the prohibitive licensing costs made it unfeasible.

It became clear we needed a self-service reporting engine that empowered clients to pull real-time insights as needed. At the time, the business was paying a premium for a workflow that was fundamentally broken:

- **Data Silos:** Critical data lived in Databricks, isolated from the client-facing portal.
- **High Friction:** Creating a single report required tedious coordination between Data Engineering, the Sigma platform team, and Client Success.
- **Scalability Issues:** Customizing reports for individual renewable energy portfolios was nearly impossible at scale.
- **Missed Opportunity:** Leadership and Sales viewed analytics as a cost center rather than the market differentiator it could be.

To prove a path forward, I collaborated with the data engineering team to stand up a POC instance using **Lakebase**. I used this to build and integrate a live report into our portal, eventually evolving the architecture into a configuration-driven system that reduced report turnaround time from weeks to under a day.

Action

This was a cross-organizational transformation. I identified early on that the technical implementation would be straightforward compared to the challenge of alignment. The project required buy-in from Data Engineering, AppSec, Infosec, Marketing, and Executive Leadership—all of whom held different priorities.

Honing Requirements Across Disciplines

I partnered with **Data Engineers and Managers** to map out existing Databricks structures and transformations. Simultaneously, I worked with **Marketing and Sales** to identify the specific analytics capabilities that would resonate with prospects. By collaborating with **Leadership**, I framed the project not as a mere feature, but as a foundational platform investment. Crucially, I engaged **Infosec** early to bake data isolation and access patterns into the architecture from day one.

Designing the Data Pipeline

The core challenge was translating complex Databricks volumes into fast, flexible, client-facing reports. I architected a system focused on three pillars:

- * **Decoupled ETL:** Pipelines extracted data from Databricks into a dedicated reporting database, shielding the production warehouse from analytical load.
- * **Materialized Views:** These served as the "source of truth," pre-computing common aggregations to power both internal dashboards and external portals simultaneously.

* **Configuration-First Schema:** I designed a flexible schema where new reports are defined via configuration files rather than custom code, enabling us to deploy new views in hours.

Building for Customization at Scale

A primary requirement from Sales was per-client customization. Rather than building bespoke code for every user, I designed a system where reports are composed of reusable data components and layout templates. This allowed a single engineering team to support hundreds of unique client configurations without the linear scaling costs imposed by our previous tools.

Iterative Delivery & Alignment

I maintained momentum through regular syncs with all stakeholders—validating pipeline reliability with Data Engineering while ensuring the product narrative stayed aligned with Sales and Marketing. We delivered incrementally: first proving the pipeline with internal reports, then launching the client portal, and finally enabling full customization.

Result

The impact was both immediate and structural:

* **Velocity:** New report creation dropped from **weeks to less than 24 hours**.

* **Empowerment:** Customization became a configuration task for the Client Success team, removing Engineering as a bottleneck.

* **Competitive Edge:** The client portal evolved from a transactional tool into a robust analytics platform, becoming a key differentiator in sales cycles.

* **Data Integrity:** We eliminated the "two versions of the truth" problem by using a shared foundation for internal and external reporting.

* **Performance:** Materialized views provided predictable, high-speed performance regardless of the underlying data volume in Databricks.

Learning

- **Alignment is the bottleneck, not architecture.** The technical patterns (ETL, Materialized Views) were well-understood. The "hard" part was keeping seven different departments converged on a shared vision.
- **Design for the report you hasn't been requested yet.** Investing in a configuration-based system felt like an "extra" step during the POC, but it prevented a linear scaling crisis three months later.
- **Eliminate intermediaries, not just tools.** Our success wasn't just in replacing Sigma; it was in removing the human-in-the-loop dependencies that slowed down the business.
- **The power of clean boundaries.** Materialized views served as the essential contract between messy source data and the polished end-user experience.
- **Cross-functional communication prevents "Building the Wrong Thing."** Constant dialogue with Marketing and Sales ensured that our technical milestones actually moved the needle for the business.