GEMMA ANALYTICS

PROPOSAL

Modern Data Platform — Proof of Concept

Prepared for

Gregory's

2026-05-08

Valid for 7 days

Table of Contents

EXECUTIVE SUMMARY

Gemma Analytics proposes a **proof of concept** engagement to build Gregory's modern data platform end-to-end – from data ingestion through cloud warehousing and data modeling to interactive dashboards – for one or two selected reports that are representative of Gregory's broader reporting landscape.

The goal is to demonstrate, on a real and meaningful slice of the business, that a modern cloud-native stack can **match and exceed** what Targit and the current on-premise infrastructure deliver today: faster refreshes, richer visualizations, modular and independently runnable data pipelines, and a foundation that scales to the full reporting estate over time.

The technology stack – **Snowflake, dbt, Airflow, dlt, and Lightdash Enterprise** – is chosen to address every pain point Gregory's has identified: the monolithic nightly refresh, the limited visual repertoire, the single OLAP cube that binds everything together, and the absence of a clear platform roadmap.

This is not a throwaway prototype. The infrastructure, pipelines, and data models built during the proof of concept are production-grade and will serve as the foundation for a full migration.

Timeline: 6–12 weeks, depending on final scope.
Investment: EUR 25,000–40,000 (fixed price), depending on the reports and data sources confirmed during scoping.

GREGORY'S SITUATION

THE BUSINESS

Gregory's is the number one food service brand in Greece, operating approximately 367 stores across Greece, Germany, Romania, and Cyprus through a franchise model. The organization serves over 250,000 customers daily, employs more than 2,500 people, and has been growing continuously for over 50 years.

CURRENT DATA & BI STACK

Gregory's has built a substantial reporting capability over the past seven to eight years on an on-premise SQL Server and Targit stack. The architecture follows a well-structured layered approach:

- **ERP:** Microsoft Business Central (recently migrated from on-premise to cloud; data available via OData and REST APIs)
- **Ingestion:** SSIS packages (Visual Studio) and Azure Data Factory for Business Central cloud data
- **Data Warehouse:** On-premise SQL Server with bronze (stage tables), silver (source views and model tables), and gold (final dimension and fact views) layers feeding an OLAP cube
- **Orchestration:** SQL Agent jobs with approximately 200 steps in total, running as a nightly batch
- **BI Tool:** Targit – reports, scheduled PDF and Excel email exports, dynamic date filtering

The data warehouse spans roughly 760 GB across over 900 tables and 800 views, supporting reports across more than 20 departments – from sales and marketing to supply chain, finance, operations, and human resources.

WHAT WORKS WELL

Not everything needs replacing. Several aspects of the current setup are valuable and should be preserved or improved upon in a new platform:

- **Dynamic date handling** – Targit's built-in period comparisons (year-to-date, month-to-date, prior year, rolling periods) are heavily used and expected by report consumers
- **Centralized metrics and dimensions** – the OLAP cube provides a single source of truth where measures and dimensions are defined centrally and reused across reports
- **Scheduled email deliveries** – automated PDF and Excel reports sent to management on a schedule
- **Internal adoption** – stakeholders trust the numbers and have built workflows around the existing reports
- **Data lineage awareness** – the layered warehouse architecture provides clear traceability from source to report

PAIN POINTS

- **Nightly refresh takes 13–15 hours** – cube processing alone accounts for approximately 8 hours. A failure mid-run means losing the entire day's refresh, with no ability to rerun individual components
- **No incremental or partial refreshes** – the monolithic OLAP cube forces an all-or-nothing approach. Individual tables or subject areas cannot be refreshed independently. Multiple refreshes per day are not feasible
- **Limited visualizations** – Targit's visual repertoire is narrow: tables, basic bar and column charts, and waterfall charts. No histograms, no custom visuals, no decomposition or metric trees. Certain report types are difficult or impossible to build
- **Single point of failure** – Petros is currently the only person in the organization building reports. The entire reporting operation depends on one person's availability and knowledge
- **Cube binding effect** – the OLAP architecture constrains how data can be modeled and queried, making it difficult to add new subject areas or change existing structures without affecting the entire cube
- **Descriptive analytics only** – the current stack supports traditional reporting but offers no path to predictive analytics, data science, or AI-driven insights
- **Platform stagnation** – Targit updates infrequently and lacks a forward-looking platform roadmap. Technology is evolving rapidly, and Gregory's wants to be on a platform that evolves with it
- **Vendor and integrator dependency** – the entire A-to-Z solution (SQL scripts through Targit dashboards) depends on a single external integrator. There are few alternative Targit integrators in the market, creating concentration risk
- **Limited data source connectivity** – adding new data sources requires SSIS expertise and manual integration work. Departments like quality control still rely on Excel files uploaded to SharePoint, creating data silos
- **No real-time or intra-day reporting** – real-time data is only possible for small datasets using relational models. The OLAP cube, which powers most reporting, refreshes once daily at best

GOALS

Gregory's has articulated a clear vision for what comes next:

1. A **cloud-based, future-proof data warehouse** that replaces the on-premise SQL Server and OLAP cube
2. **Rich visualizations** – decomposition trees, metric trees, and the ability to create custom visuals
3. **Modular data pipelines** that can be run and refreshed independently, breaking the monolithic nightly batch
4. **AI capabilities** – conversational analytics, predictive statistics, and data science use cases
5. A **data lakehouse** that can accommodate both structured and unstructured data
6. A platform with a **clear roadmap, frequent updates, and an ecosystem of integrators**
7. **Scheduled deliveries** that match or exceed Targit's current email job capabilities

This proof of concept addresses the first three goals directly and lays the groundwork for the rest.

PROOF OF CONCEPT: OUR APPROACH

NARROW BUT DEEP

Rather than attempting to scope the full migration upfront, we propose a focused proof of concept that rebuilds **one or two representative reports** end-to-end in the new stack. The reports will be selected jointly in the scoping call and should be complex enough to be representative of the broader reporting landscape – ideally including the sales invoiced dataset that powers Gregory's core sales reporting.

This approach has three advantages:

1. **Tangible proof before commitment** – Gregory's sees a working end-to-end chain on real data before investing in a full migration
2. **Risk reduction** – the proof of concept surfaces integration challenges, data quality issues, and stakeholder expectations early, when they are cheapest to address
3. **Production-grade foundation** – the infrastructure, pipelines, data models, and BI configuration built during the proof of concept are not disposable; they become the base layer for the full migration

FULL VALUE CHAIN

The proof of concept covers every layer of the modern data stack – not just the BI tool or just the warehouse. This is deliberate: Gregory's pain points span the entire chain (ingestion speed, cube rigidity, visualization limitations), and a proof of concept that only addresses one layer would not answer the real question.

Data Sources → **dlt + Airflow** → **Snowflake** → **dbt** → **Lightdash Enterprise**

Each layer is connected: data flows from the source systems through ingestion pipelines into the cloud warehouse, is modeled and tested in dbt, and surfaces as interactive dashboards in Lightdash. Scheduled deliveries (email, PDF) are configured where relevant.

SCOPE

The engagement is structured into three work streams. Together they form a single, fixed-price deliverable.

Work Stream 1: Infrastructure & Data Loading

Cloud data platform setup from A to Z: Snowflake warehouse, data ingestion pipelines (dlt), Airflow orchestration, dbt project scaffolding, Lightdash Enterprise configuration, CI/CD, and all supporting infrastructure. Connects the confirmed data sources and lands data in the warehouse.

Focus: Platform setup, pipeline development, data landing
Runs: First half of the engagement

Work Stream 2: Analytics Engineering

Data modeling in dbt (staging, intermediate, and mart layers) for the selected report scope. Semantic layer definitions (metrics and dimensions) that provide the governed, reusable building blocks for all downstream reporting and AI use cases.

Focus: Data models, semantic layer, testing, documentation
Runs: Middle of the engagement, overlapping with WS1

Work Stream 3: Analysis & Reporting

Interactive dashboards in Lightdash that recreate and improve upon the selected Targit reports. Scheduled deliveries where agreed. Validation against existing reports to confirm correctness and demonstrate improvements.

Focus: Dashboards, scheduled reports, validation
Runs: Second half of the engagement

WORK STREAM 1 – INFRASTRUCTURE & DATA LOADING

Snowflake

- Cloud data warehouse setup: database, schemas, roles, and access controls
- Environment separation (production and development)
- Warehouse sizing appropriate for the proof of concept scope

Data Ingestion (dlt + Airflow)

- **Airflow** as the orchestration layer – DAG-based scheduling, retry logic, failure notifications, and monitoring
- **dlt** (data load tool) pipelines for each confirmed data source
- Incremental loading where the source API supports it – demonstrating the shift from monolithic nightly batch to modular, independently runnable pipelines
- Pipeline documentation: load strategy, schema handling, and refresh cadence

The confirmed data sources for the proof of concept will be finalized during the scoping call. Business Central is the expected primary source. Depending on which reports are selected, additional sources (e.g. POS system, delivery platforms) may be in scope.

dbt Project Scaffold

- Best-practice project structure: staging, intermediate, and mart layers
- Development and production profiles targeting appropriate Snowflake environments
- Naming conventions, base model templates, and package configuration
- **Staging models for all ingested sources** – standardized rename, cast, and deduplication layer that validates pipeline output

Lightdash Enterprise

- Lightdash Enterprise instance connected to Snowflake
- Project configuration, user setup, and initial space structure

CI/CD & Developer Experience

- GitHub-based workflow: all code in version control, changes via pull requests with review
- Automated testing on pull request (dbt slim CI, pipeline validation)
- Deployment pipeline: merge to main triggers production builds

WORK STREAM 2 – ANALYTICS ENGINEERING

dbt Data Models

- Staging, intermediate, and mart models for the selected report scope
- Comprehensive tests per layer (schema tests, data quality, referential integrity)
- Documentation generation for all models and columns

Semantic Layer

- Metric and dimension definitions in the dbt semantic layer, synced to Lightdash

- Consistent business logic: metrics are defined once in code and reused across all dashboards, queries, and future AI agents
- Dynamic date handling that matches Targit's built-in period comparisons (year-to-date, month-to-date, prior year, rolling periods) – implemented as reusable logic in the semantic layer

WORK STREAM 3 – ANALYSIS & REPORTING

Dashboards & Reports

- Interactive dashboards in Lightdash that **recreate and improve upon** the selected Targit reports
- Emphasis on the visual and analytical improvements that the new stack enables: richer chart types, interactive filtering, drill-down capabilities
- Scheduled email deliveries of dashboards and reports (PDF, PNG, or CSV) where applicable

Validation

- Side-by-side comparison of rebuilt reports against existing Targit outputs to confirm data correctness
- Walkthrough with Gregory's team to demonstrate new capabilities and gather feedback

AI READINESS

A key advantage of the modern stack is that it lays the foundation for AI-driven analytics – not as a future aspiration, but as a capability that becomes available as soon as the semantic layer is in place.

LIGHTDASH AI AGENTS

Lightdash Enterprise includes **AI agents** that operate inside the Lightdash interface and via Slack or Microsoft Teams. Users can ask questions in natural language – for example, "What were our top 10 stores by sales last month?" – and receive accurate, governed answers. The answers are accurate because the AI queries through the **semantic layer**, where metric definitions, joins, and filters are defined in code rather than guessed by the model.

Published benchmarks show that text-to-SQL applied directly to raw warehouse tables reaches around 65% accuracy – not reliable enough for business decisions. When the same models query through a semantic layer, accuracy rises substantially. The semantic layer built during this proof of concept is the prerequisite that makes AI on analytics production-ready.

MODEL CONTEXT PROTOCOL (MCP)

Lightdash Enterprise exposes an **MCP server** – a standardized interface that allows external AI tools to query the data platform through the governed semantic layer. This means tools like Claude, ChatGPT, Cursor, or custom-built agents can access Gregory's data with the same accuracy guarantees as the built-in Lightdash AI agents.

The MCP server is the foundation for more advanced AI use cases:

- Custom AI agents tailored to specific business functions (e.g. a supply chain assistant that answers logistics questions)
- Integration of analytics into existing tools and workflows (e.g. AI-assisted analysis in spreadsheets or presentations)
- Agentic workflows that generate dashboards, detect anomalies, or produce scheduled narrative reports

AI-ASSISTED DEVELOPMENT

The dbt and Lightdash ecosystem is optimized for AI-assisted development. Metric definitions, model scaffolding, and dashboard configuration can be generated and maintained with AI support – reducing the burden on Petros as the sole report builder and making the platform more accessible to other team members.

TECHNOLOGY STACK

Snowflake Cloud Data Warehouse	**dbt Core** Data Modeling & Testing	**dlt** Data Ingestion	**Apache Airflow** Orchestration
Lightdash Enterprise BI & Dashboards	**GitHub Actions** CI/CD	**Docker** Containerization	

Why This Stack

- **Snowflake** replaces the on-premise SQL Server and OLAP cube. It is a fully managed cloud data warehouse that separates storage from compute – meaning queries and refreshes do not compete for resources, and scaling is elastic. There is no cube to process: dbt models materialize as tables and views that Snowflake serves directly. The 13–15 hour nightly refresh becomes a set of independent pipeline runs, each completing in minutes.
- **dbt** (data build tool) replaces the layered SQL scripts and SSIS transformations. Models are modular, version-controlled, and testable. Each model can be run independently or as part of a directed acyclic graph – the direct answer to "breaking the big cube into blocks that can be run separately." dbt also provides the **semantic layer** that defines metrics and dimensions once, in code, for use across all downstream consumers.
- **dlt** (data load tool) handles data ingestion – extracting data from source systems (Business Central, POS, and others) and loading it into Snowflake. dlt pipelines are pure Python, support incremental loading natively, and are designed to be operated by analytics engineers rather than requiring dedicated platform engineering.
- **Apache Airflow** orchestrates the pipelines: scheduling, dependency management, retries, alerting, and monitoring. Each pipeline is a separate DAG (directed acyclic graph) that runs independently – replacing the monolithic nightly batch with a modular, observable system where individual pipelines can be scheduled at different frequencies.
- **Lightdash Enterprise** replaces Targit as the BI and visualization layer. It is natively integrated with dbt's semantic layer, which means metric definitions, joins, and filters are governed in code and reused consistently across every dashboard, query, and scheduled delivery. Enterprise-specific capabilities relevant to Gregory's include:
  - **Custom data apps** – code-first custom visualizations (including decomposition trees and metric trees) powered by the Vega grammar and AI-assisted generation
  - **AI agents** – conversational analytics inside the Lightdash interface and via Slack, querying through the semantic layer for governed, accurate answers
  - **Embedding** – dashboards can be embedded into existing internal portals (e.g. a franchisee operations app), extending reach beyond direct Lightdash users
  - **Scheduled deliveries** – email, Slack, and Microsoft Teams delivery of dashboards, charts, and reports on a schedule – matching Targit's current scheduled job capabilities
  - **Pre-aggregates** – pre-computed metrics served from cache for fast dashboard response times at scale

TIMELINE

The proof of concept is expected to run for **6 to 12 weeks**, depending on the reports and data sources confirmed during scoping. The work follows a phased structure with natural overlap between work streams.

Pre-Engagement Week 0

Scoping finalized, legal documents signed, source credentials provisioned, Snowflake and Lightdash accounts set up.

Infrastructure & Data Loading Weeks 1–4

Snowflake configuration, Airflow and dlt pipeline development for confirmed data sources, dbt project scaffold and staging models, Lightdash Enterprise setup, CI/CD.

Analytics Engineering Weeks 3–6

dbt intermediate and mart models for selected reports, semantic layer definitions. Iterative refinement with Gregory's team.

Analysis & Reporting Weeks 5–8

Dashboard development, scheduled deliveries, validation against existing Targit reports. Final QA, documentation review, walkthrough with Gregory's team. Phase 2 scoping.

INVESTMENT

The proof of concept is delivered as a **fixed-price engagement**:

Proof of Concept — Fixed Price	EUR 25,000 – 40,000

The final price within this range depends on the number and complexity of reports to be rebuilt and the data sources required, which will be confirmed during the scoping call.

The fixed price covers all Gemma Analytics consulting effort: infrastructure setup, pipeline development, data modeling, dashboard development, project management, and documentation.

What Is Not Included

- **Snowflake usage costs** – Snowflake offers a 30-day free trial with USD 400 in credits. After the trial period, warehouse and storage costs are billed directly by Snowflake to Gregory's. For a proof of concept of this scope, monthly costs are expected to be modest (in the range of USD 50–150/month).
- **Lightdash Enterprise subscription** – Lightdash has made an Enterprise instance available to Gregory's for the duration of the proof of concept at no cost. Should Gregory's proceed to a full engagement, the Enterprise subscription would be contracted directly with Lightdash.
- **Airflow hosting** – Airflow requires a compute environment (a Linux VM or container service). Hosting costs are minimal (EUR 10–30/month for a small cloud VM) and billed directly by the cloud provider.

All prices are net of VAT. Gemma Analytics GmbH invoices upon completion of the proof of concept. Travel expenses, if any, are passed through at cost.

PREREQUISITES

The following should be in place before the engagement starts.

- **Source system credentials** – API keys and access credentials for Business Central and any additional data sources confirmed during scoping.
- **Snowflake account** – Gregory's Snowflake account provisioned (Gemma can assist with setup).
- **Lightdash Enterprise** – access to the Gregory's sandbox instance at gregorys.lightdash.cloud.
- **GitHub access** – Gemma added to a shared repository with write access for pipeline and dbt code.
- **Technical contact** – a designated contact on Gregory's side for data source questions and report validation during the engagement.

WAYS OF WORKING

Area	Approach

Communication	Shared communication channel (Slack, Teams, or email); direct access to the engineering team
Sync Cadence	Weekly call to review progress, validate outputs, and steer priorities
Team	Project Leader, Analytics Engineer, Analyst
Code Review	All work via pull requests with review before merge
Documentation	All pipelines, models, and infrastructure documented to handoff standard
Validation	Gregory's team validates report outputs against existing Targit reports to confirm correctness

PHASE 2 & BEYOND

The proof of concept establishes the full technical foundation. Phase 2 scope will be defined based on proof of concept results and evolving business priorities. Potential areas:

FULL REPORT MIGRATION

- Systematic migration of Gregory's reporting estate from Targit to Lightdash – department by department, starting with the highest-value reports
- Onboarding of additional data sources (POS, delivery platforms, aggregators, B2B CRM, loyalty scheme) as reports require them
- Expansion of the dbt semantic layer to cover all business domains

METRIC TREES & CUSTOM DATA APPS

- **Decomposition trees and metric trees** – the visualization pattern Gregory's has identified as a priority. Lightdash Enterprise's custom data apps (powered by the Vega visualization grammar and AI-assisted generation) are the direct answer to this requirement
- **Branded, interactive data applications** – custom-built analytical tools that go beyond standard dashboards, tailored to Gregory's specific analytical workflows

AI & CONVERSATIONAL ANALYTICS

- **AI agents** inside Lightdash and via Slack – enabling users across the organization to ask questions in natural language and receive governed, accurate answers grounded in the semantic layer
- **Model Context Protocol (MCP) server** – allowing external AI tools (Claude, custom agents) to query Lightdash through the semantic layer
- The semantic layer built during the proof of concept is the foundation that makes AI on analytics accurate and production-ready, rather than a demo

DATA LAKEHOUSE & ADVANCED ANALYTICS

- Extension of the platform to accommodate **unstructured data** alongside structured reporting data
- **Predictive analytics and data science** – forecasting, scenario modeling, and automated anomaly detection
- Combination of public and online information with warehouse data

FRANCHISEE REPORTING & EMBEDDING

- **Embedded dashboards** inside existing franchisee portals or store operations applications – extending the reach of reporting to all 367 stores without requiring individual Lightdash logins
- Role-based access and row-level security to ensure each franchisee sees only their own data

OPERATIONAL INDEPENDENCE

- Knowledge transfer and documentation to enable Gregory's team to extend the platform independently
- Ongoing support models (retainer, on-demand) available as needed

LEGAL FRAMEWORK

The engagement is governed by the following contracts:

Document	Purpose
Framework Service Agreement (FSA)	Master terms governing the consulting relationship, liability, IP, and general obligations.
Non-Disclosure Agreement (NDA)	Mutual confidentiality agreement. Must be signed before any access is granted.
Data Processing Agreement (DPA)	Per GDPR requirements, covering any personal data processed during the engagement.
Legal Offer	A short, signable summary of the scope, deliverables, timeline, and pricing defined in this proposal.

ABOUT GEMMA ANALYTICS

Gemma Analytics is a Berlin-based data consultancy specializing in modern data platform implementation. We work with growth-stage companies to build production-grade data infrastructure – from warehouse setup and pipeline engineering through to KPI dashboards and team enablement.

Our team brings deep experience across the modern data stack, with 70+ completed data platform projects. We operate code-first and leverage AI-assisted tooling to deliver at speed without compromising on quality or maintainability.

Over these projects we have built extensive internal documentation, reusable patterns, and engineering best practices that carry over to every new engagement – accelerating development significantly from day one.

GEMMA ANALYTICS

Gemma Analytics GmbH – Chausseestraße 17, 10115 Berlin
gemmaanalytics.com

CONTACT

Bijan Soltani

Founder & CEO

bijan.soltani@gemmaanalytics.com

+49 179 7438926

Appendix: Open Questions for Scoping

The following questions should be resolved during the scoping call to finalize the proof of concept scope and pricing:

1. **Which reports?** – Which one or two Targit reports should the proof of concept rebuild? The sales invoiced report and its associated dimensions are a natural candidate. What is the second report, if any?
2. **Which data sources?** – Which source systems feed those reports? Business Central is expected. Does the selected scope also require data from the POS system, delivery platforms, or other sources listed in the architecture overview?
3. **Report complexity** – How many dimensions and metrics are involved in each selected report? The existing Targit reports have an extremely detailed dimension structure (folder hierarchies, many filter criteria). Should the proof of concept replicate this level of granularity in full, or is a pragmatic subset acceptable?
4. **Scheduled deliveries** – Should the proof of concept include scheduled email deliveries (PDF, Excel) of the rebuilt reports? This is straightforward to include but should be confirmed.
5. **Metric trees** – Gregory's has identified decomposition and metric trees as a high-priority visualization. Should the proof of concept include a metric tree as a deliverable (using Lightdash Enterprise's custom data apps capability), or should this be reserved for Phase 2?
6. **Dynamic date handling** – The current Targit reports rely heavily on dynamic date filters (year-to-date, MTD, prior year comparisons). These are fully supported in Lightdash and dbt. Should exact parity with Targit's date handling be a success criterion for the proof of concept?
7. **Validation process** – Who on Gregory's side will validate that the rebuilt reports match (or improve upon) the existing Targit reports? Is Petros the sole validator, or will other stakeholders be involved?
8. **Airflow hosting** – Does Gregory's have a preferred cloud provider (AWS, Azure, GCP) for hosting the Airflow instance, or should Gemma recommend one?