**Architecture Requirements**

*Identify architecture requirements by analyzing constraints (technical, operational,regulatory).*

**I. Business Requirements (BR) -**

* BR-1: Multi-format Data Collection

*The platform must allow users to securely upload their accounting documents in PDF, CSV, and Excel (.xlsx) formats.*

* BR-2: Automated Processing

*The system must automatically extract, validate, and structure relevant data from the uploaded documents for further analysis.*

* BR-3: Financial Indicators Calculation

*The system must reliably compute a predefined set of key financial indicators that support strategic decision-making.*

* BR-4: Data Visualization

*The platform must display these indicators through interactive and intuitive dashboards designed for business users.*

**II. Technical Constraints -**

Mandatory Technologies

As specified in the project requirements, the following stack components are mandatory:

* Containerization: Docker
* Orchestration: Kubernetes
* Workflow Automation: Apache Airflow
* Monitoring: Grafana

Team Technology Choices

* Cloud Provider: Google Cloud Platform (GCP)

*Rationale: GCP was selected for the robustness and simplicity of its managed Kubernetes service (Google Kubernetes Engine – GKE), which significantly streamlines deployment and scaling operations. Its data- and AI-oriented ecosystem also provides strong growth potential for Quantis (e.g., predictive modeling). Additionally, GCP’s Identity and Access Management (IAM) offers granular and secure access control — a critical factor for ensuring data protection.*

* Programming Language & Framework: Python

*Rationale: Python is the de facto standard for data processing, offering a mature and extensive ecosystem (Pandas, NumPy, etc.) that accelerates development and improves maintainability.*

* Databases: PostgreSQL

*Rationale: PostgreSQL extracted financial data is highly structured and relational (linked to companies, fiscal years, etc.). PostgreSQL is the reference open-source relational database, known for its reliability, robustness, and strong performance — ideal for FinTech applications.*

**III. Operational Requirements -**

Scalability

* Initial Target: Support 100 client companies in the first year.
* Scalability Goal: The architecture must scale seamlessly to 5,000 clients within three years.
* Workload: The system must handle up to 200 documents per day, with predictable activity peaks during month-end accounting closings.

Performance (SLOs – Service Level Objectives)

* Processing Time: 95% of uploaded documents (including extraction and computation) must be processed within 60 seconds.
* API Latency: The 95th percentile (P95) of API response times for dashboard endpoints must be below 200 ms.

Availability

* Uptime Objective: The client-facing service must guarantee 99.9% availability.
* Technical Implications:
  + Implement readiness and liveness probes in Kubernetes.
  + Deploy all critical microservices (e.g., API) with a minimum of two replicas to ensure redundancy and high availability.

**IV. Regulatory and Security Constraints -**

Data Privacy (GDPR Compliance)

* Data Isolation: Strict logical isolation will be enforced at the application level. Every database query must include a client\_id filter to prevent data leaks between tenant accounts. No data should ever be accessible without this filter.

Security Measures

* Encryption:
  + In Transit: All traffic must be secured using TLS 1.2 or higher (HTTPS).
  + At Rest: Data stored in PostgreSQL, Redis, and Cloud Storage will be encrypted by default using GCP’s native mechanisms.
* Authentication:
  + API access will be secured through short-lived JSON Web Tokens (JWTs).
* Document Storage:
  + Uploaded files will be stored in a private Google Cloud Storage bucket.
  + Access will be granted exclusively via short-lived, single-use signed URLs generated by the backend.
  + Clients will never have direct access to the bucket — this is a core security design principle.