

Project Ariadne:
Multi-Curve Estimation & Calibration Service

Igor Melnikov

November 9, 2025

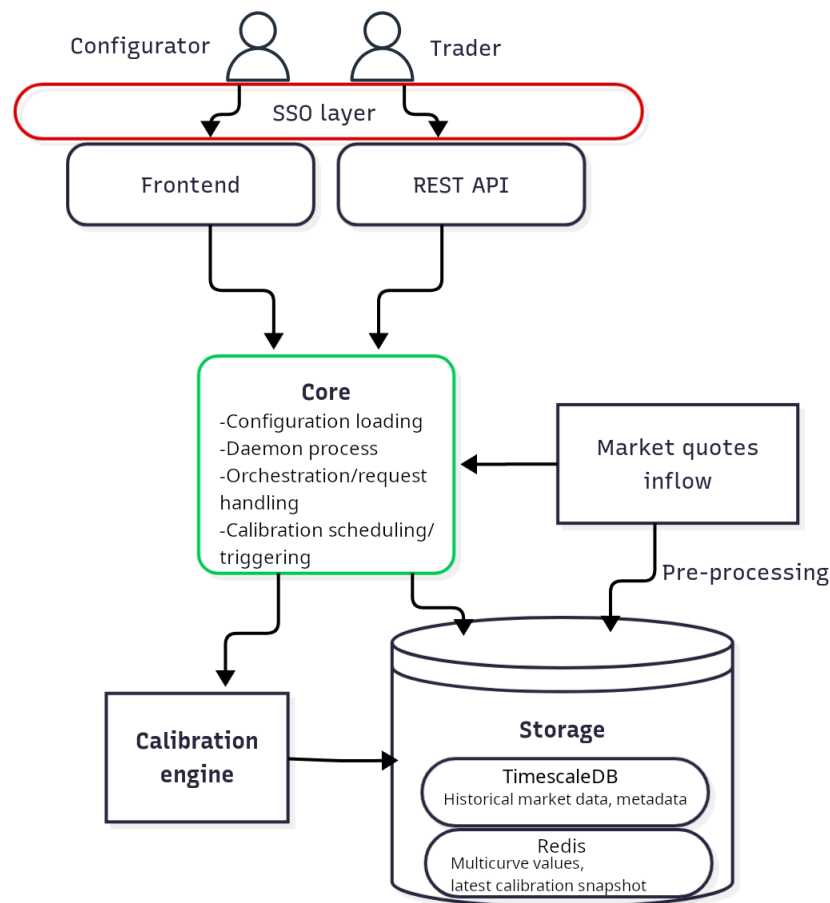
Executive Summary

The Ariadne project aims to design and implement a centralised, robust, and scalable service infrastructure for estimation and calibration of multiple interest rate curves (Multi-Curve) to market data. This system will serve as the single source of truth for risk-free rates (e.g., SOFR, €STR), forwarding curves, and basis curves, providing pricing and risk evaluation data for trading desks across Rates, FX, and Credit products.

System Architecture & Design

The system follows a modular architecture with a core process responsible for module communication, function calls, handling requests, logging and self-testing. By design, to ensure scalability, the core process should support multiple parallel calls to read the in-memory multicurve values and send them back with low latency. The core is responsible for calibration calls based on configured scheduling, market data trigger or manual request. In the calibration step, the core process feeds the latest pre-processed market quotes for the selected instruments to the calibration engine, resulting in a multicurve estimate, which is stored in the memory cache for fast access.

Component Diagram



Principal Components

1. Core Process

- **Purpose:** Orchestrate a well-coordinated workflow across components, handle requests, trigger calibrations

2. Data Ingestion Service

- **Purpose:** Consume, validate, and normalize raw market data
- **Inputs:** Live feed
- **Data Types:** Deposits, Treasury quotes, OIS Swaps, FRA, IRS, IR Futures, Basis Swaps, FX Forwards

3. Calibration Engine

- **Purpose:** Core logic for curve building process
- **Process:** Bootstrapping Discounting Curve -> Forward Curve Construction -> Basis Curve Construction

4. Data & Curve Storage

- **TimescaleDB:** Time-series database for storage of previous calibrations
- **Redis:** Curve and latest market snapshot caching

5. Service Layer secured with SSO authentication

- **REST API:** Programmatic access to curves and calibration
- **Web Dashboard:** Frontend for monitoring and analysis

Scheme of Usage

- **Quantitative Analysts:** Configure curves and analyse logs
- **Trading & Risk Operators:** Consume curves via API for low-latency real-time operations
- **Middle Office:** EOD batch calibration and verification

Functional Requirements

1. Curve definition management (CRUD operations)
2. Calibration execution (on-demand, scheduled, triggered)
3. Results delivery via REST API and file-based formats
4. Analysis & diagnostics with revaluation errors
5. Audit & logging with full calibration history

Non-Functional Requirements

1. Low Latency

- Intra-day calibration: < 10 seconds
- API response: < 100ms for cached curves

2. High Accuracy & Robustness

- Revaluation errors within bid-ask spread
- Algorithm convergence rate: >99.9%

3. Scalability & Availability

- Calibrate 50+ currency/curve combinations in < 5 minutes
- Support concurrent requests
- Redundant service instances

4. Data Audit

- Full traceability of curves to input data and model versions

5. Security

- SSO integration with role-based access control

Other Key Considerations

Technological Stack

- **Backend:** C++
- **Frontend:** Python, PyQt
- **Data:** Redis, TimescaleDB
- **Infrastructure:** Docker, Kubernetes

Project Evaluation Metrics

- Reduction in manual curve management time
- Elimination of curve inconsistencies across desks
- Reduction in number of failed trades due to curve mispricing
- Meeting the defined NFRs