ECOMETER
PROJECT DOCUMENTATION

CLAPTAC - IWB25

Contents

1. Introduction

EcoMeter is a web application designed to help Sri Lankan households manage electricity consumption more effectively. It addresses the complexity of Sri Lanka's **Block Tariff (Domestic Type 1)** and **Time-of-Use (TOU, Domestic Type 2)** systems by offering transparent insights, cost-saving recommendations, and sustainability awareness. Unlike traditional approaches, EcoMeter preloads nationwide tariffs so users don't need to enter them manually.
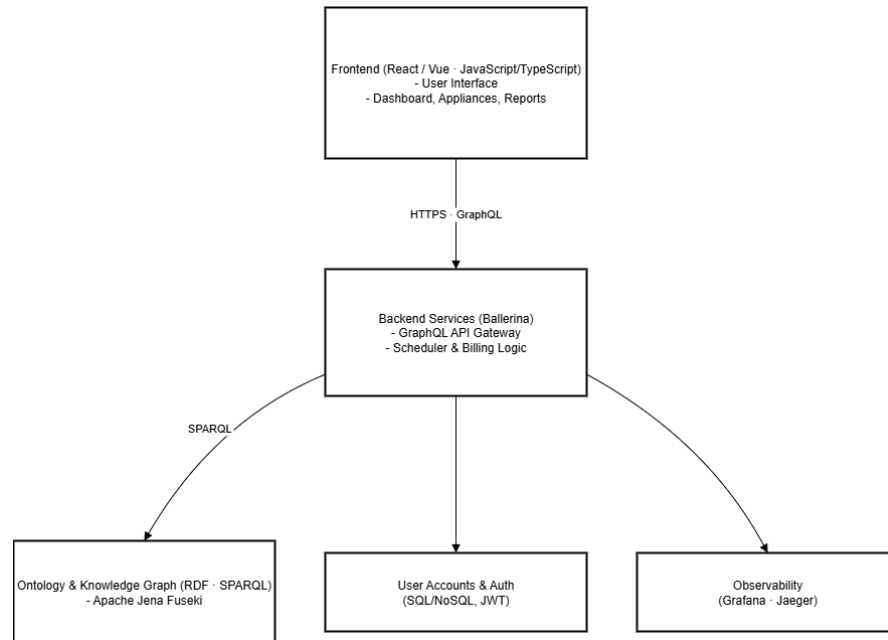
2. Problem Statement

Sri Lankan households face three key issues: 1. **Unpredictable Bills** – Block tariff users unintentionally cross thresholds, causing sudden bill spikes. 2. **Missed Savings** – TOU users don't optimize usage for cheaper off-peak hours. 3. **Invisible Environmental Impact** – CO_2 emissions are abstract, making it difficult for households to connect daily usage with sustainability.

3. Objectives

- Project household electricity bills clearly and accurately.
- Provide optimized **recommendations**: Cheapest, Greenest, and Balanced.
- Convert CO_2 emissions into **tree equivalents** for better understanding.
- Keep the system **simple and usable**, requiring minimal input.
- Leverage **Ballerina** for seamless integration and observability.

4. System Architecture


```
                    ┌─────────────────────────────────────┐
                    │ Frontend (React / Vue · JavaScript/TypeScript) │
                    │  - User Interface                    │
                    │  - Dashboard, Appliances, Reports    │
                    └─────────────────────────────────────┘
                                    │
                              HTTPS · GraphQL
                                    │
                    ┌─────────────────────────────────────┐
                    │ Backend Services (Ballerina)         │
                    │  - GraphQL API Gateway               │
                    │  - Scheduler & Billing Logic         │
                    └─────────────────────────────────────┘
                         │            │            │
                      SPARQL
              ┌──────────────────┐ ┌──────────────┐ ┌──────────────┐
              │ Ontology &       │ │ User Accounts│ │ Observability│
              │ Knowledge Graph  │ │ & Auth       │ │ (Grafana ·   │
              │ (RDF · SPARQL)   │ │ (SQL/NoSQL,  │ │ Jaeger)      │
              │ - Apache Jena    │ │ JWT)         │ │              │
              │ Fuseki           │ │              │ │              │
              └──────────────────┘ └──────────────┘ └──────────────┘
```

High-Level Overview

- **Frontend (React/TypeScript)**: User-friendly web interface for onboarding, dashboards, recommendations, and reports.
- **Backend (Ballerina)**:
  - **GraphQL API Gateway** to connect frontend and backend services.
  - **Scheduler Service (gRPC)** to optimize appliance usage times.
  - **Billing Service (gRPC)** to calculate projected bills for Block/TOU tariffs.
  - **Tariff Catalog Service** to store and version official nationwide tariffs.
- **Ontology (RDF/SPARQL)**: Stores tariff models, appliances, and recommendations for explainability.
- **Database (SQL/NoSQL)**: Manages user accounts and preferences.
- **Observability**: Grafana and Jaeger dashboards using Ballerina's built-in observability.

5. Core Features

5.1 Onboarding Wizard

- Select tariff type (Block/TOU).
- Set billing cycle start date.
- Choose CO_2 model (default or custom).

5.2 Appliance & Task Editor

- Add appliances with name, wattage, and duration.
- Define usage constraints: earliest start, latest end, repeats.

5.3 Dashboard

- Displays monthly bill projection, CO_2 footprint, and tree equivalents.
- Block users see progress toward thresholds.
- TOU users use a slider to balance money vs. CO_2.

5.4 Recommendations

- Provides three suggestions per task: Cheapest, Greenest, Balanced.
- Includes a **Why** panel with cost differences and CO_2 impact.

5.5 Reports

- Monthly history of bills and CO_2.
- Exportable CSV and PDF reports.

6. Role of Ballerina

EcoMeter uses **Ballerina** extensively: - **GraphQL APIs** – clean interface between frontend and backend. - **gRPC Microservices** – separate modules for scheduling and billing. - **SPARQL Integration** – connects with RDF ontology for explainable results. - **Observability** – metrics, logs, and traces via Grafana/Jaeger.

Ballerina's strength lies in integrating diverse systems, making it perfect for linking tariffs, ontology, and user-facing APIs.

7. Example User Scenarios

Scenario A – Block Tariff User

- User has consumed 58 kWh this month.
- Adding a washing machine (3 kWh) would cross into the 61–90 block.
- System warns: *"This task will push you into Block 3. Extra cost: Rs. 240. Suggest deferring."*

Scenario B – TOU Tariff User

- Dishwasher (2 kWh) with flexible time window.
- Options shown:
  - **2 AM (Off-peak): Rs. 30, CO_2: 0.8 kg**
  - **Midday (Day band): Rs. 50, CO_2: 0.6 kg**
  - **Evening (Peak): Rs. 80, CO_2: 1.0 kg**

Scenario C – CO_2 Footprint

- Household uses 200 kWh/month.
- CO_2 = 106 kg/month → 1272 kg/year.
- Equivalent = ~58 trees annually.

8. Impact

- **Financial**: Helps households avoid costly tariff jumps.
- **Environmental**: Connects energy use to tree equivalents for clarity.
- **Educational**: Increases tariff literacy among users.
- **Scalable**: Can expand to other countries with similar tariffs.

9. Future Enhancements

- Dedicated **mobile app** for Android/iOS.
- **Smart meter integration** for real-time data.
- **Community leaderboards** to gamify eco-savings.
- Support for **renewable energy sources** (solar, batteries).

10. Conclusion

EcoMeter is a practical and innovative solution tailored for Sri Lanka. By combining **preloaded tariffs, ontology-driven insights, and Ballerina-powered integration**, it makes household energy use understandable, cost-effective, and eco-friendly. The project demonstrates how technology can simplify complex utility systems while promoting sustainability.