LLD - Eurotax Integration Platform - Hetzner Cloud Deployment

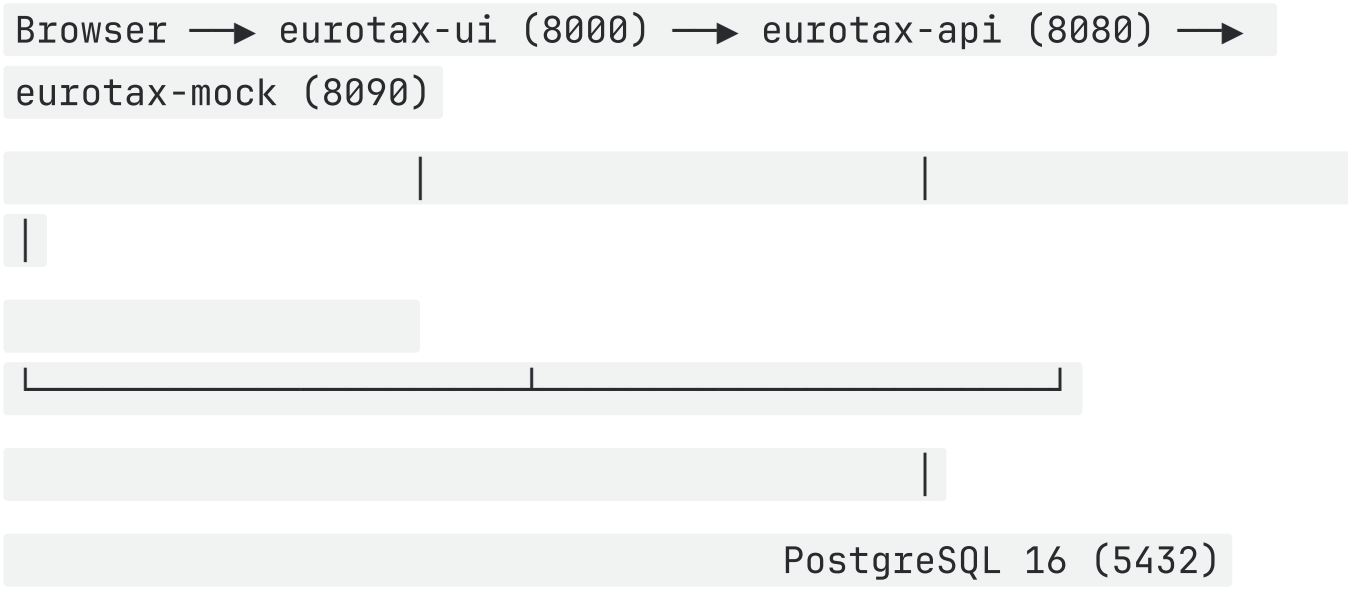
Document Information

Project	Eurotax Integration Platform
Document Type	Low-Level Design (LLD)
Version	1.0
Date	2026-02-24
Environment	Development
Author	DevOps Team

System Overview

The Eurotax Integration Platform consists of three microservices and a shared PostgreSQL database, deployed on a single Hetzner Cloud VPS using Docker Compose.

Architecture Diagram



Services

Service	Technology	Port	Description
eurotax-mock	Java 21 / Spring Boot	8090	Mock Eurotax SOAP API service. Provides simulated SOAP endpoints for development and testing.
eurotax-api	Java 21 / Spring Boot	8080	REST API with Resilience4j. Proxies SOAP to REST, provides HMAC authentication.

eurotax-ui	PHP 8.2 / Laravel 12 + React 19	8000	User-facing web application built with Inertia.js for seamless SPA experience.
PostgreSQL	PostgreSQL 16	5432	Shared database (internal only, not exposed to the internet).

Dependency Chain

The services must be started in the following order due to runtime dependencies:

1. **PostgreSQL** — database must be ready first
2. **eurotax-mock** — SOAP service depends on PostgreSQL
3. **eurotax-api** — REST API depends on PostgreSQL and eurotax-mock
4. **eurotax-ui** — Web UI depends on eurotax-api and PostgreSQL

Infrastructure — Hetzner Cloud

Provider	Hetzner Cloud
Server Name	eurotax-dev-01
Server Type	cx32 (4 vCPU, 8 GB RAM, 80 GB NVMe SSD)
Operating System	Ubuntu 24.04 LTS
Location	fsn1 (Falkenstein, Germany)
Provisioning	Terraform (hetznercloud/hcloud provider ~> 1.45)
Configuration Management	cloud-init (Docker installation on first boot)

Network & Firewall

Firewall Configuration

Firewall Name	eurotax-dev-fw
SSH Key Name	devops-demo-key
SSH Public Key Path	~/.ssh/id_rsa.pub
Docker Network Subnet	172.20.0.0/16

Allowed Ports

Port	Protocol	Source	Description
22	TCP	0.0.0.0/0	SSH access
80	TCP	0.0.0.0/0	HTTP
443	TCP	0.0.0.0/0	HTTPS
8000	TCP	0.0.0.0/0	eurotax-ui (Laravel + React)
8080	TCP	0.0.0.0/0	eurotax-api (Spring Boot REST)
8090	TCP	0.0.0.0/0	eurotax-mock (Spring Boot SOAP)

Database Configuration

Engine	PostgreSQL 16
Port	5432 (internal only — not exposed to internet)
Docker Service Name	postgres

Databases

Database	Username	Password	Used By
eurotax_mock	mock_user	mock_dev_pass_2024	eurotax-mock service
eurotax_api	api_user	api_dev_pass_2024	eurotax-api and eurotax-ui services

Service Configuration

eurotax-mock

Port	8090
Version	latest
Repository	https://github.com/sirbob/eurotax-mock
Health Check	GET /actuator/health

Environment Variables

Variable	Value

SPRING_DATASOURCE_URL	jdbc:postgresql://postgres:5432/eurotax_mock
SPRING_DATASOURCE_USERNAME	mock_user
SPRING_DATASOURCE_PASSWORD	mock_dev_pass_2024
SERVER_PORT	8090

eurotax-api

Port	8080
Version	latest
Repository	https://github.com/sirbob/eurotax-api
Health Check	GET /actuator/health

Environment Variables

Variable	Value
SPRING_DATASOURCE_URL	jdbc:postgresql://postgres:5432/eurotax_api
SPRING_DATASOURCE_USERNAME	api_user
SPRING_DATASOURCE_PASSWORD	api_dev_pass_2024
SERVER_PORT	8080
EUROTAX_MOCK_URL	http://eurotax-mock:8090

eurotax-ui

Port	8000
Version	latest
Repository	https://github.com/sirbob/eurotax-ui
Health Check	GET / (HTTP 200)

Environment Variables

Variable	Value
APP_URL	http://eurotax-dev-01:8000
API_BASE_URL	http://eurotax-api:8080

DB_CONNECTION	pgsql
DB_HOST	postgres
DB_PORT	5432
DB_DATABASE	eurotax_api
DB_USERNAME	api_user
DB_PASSWORD	api_dev_pass_2024

Deployment Notes

- **Orchestration:** Docker Compose is used for container orchestration on a single host.
- **Docker Builds:** All services use multi-stage Docker builds to minimize image size.
- **Startup Order:** PostgreSQL → eurotax-mock → eurotax-api → eurotax-ui (enforced via `depends_on` with health checks in docker-compose.yml).
- **Docker Installation:** Docker Engine and Docker Compose are installed automatically via cloud-init script on first server boot.
- **Smoke Tests:** After deployment, verify: `GET /actuator/health` on ports 8080 and 8090 returns `{"status":"UP"}`, and `GET /` on port 8000 returns HTTP 200.