TrafficCTRL

Requirements

CORE

Reverse Proxy That Stands In Front Of Server

Limits Traffic Before It Hits Backend

Multi-Tenants (IP, Tokens, Keys, Custom Headers)

Multi-Strategies(Sliding/Fixed Window, Token/Leaky Bucket)

Per-Route Configuration (Different Limits, Different Strategies, Different Tenants Per Endpoint)

Redis Integration for distributed state management

MONITORING & OBSERVABILITY

Structured Logging (Requests allowed/rejected, When & Where)

Application Metrics (Request Rates, Success/Error Rates, Total allowed/rejected)

System Metrics (Memory/CPU Usage)

Real-Time Dashboard (Live Traffic, Geographic/tenant/route distribution heatmap)

Notification Alerting System (On Lots Of Violation Spikes)

CONFIGURATION & TESTING

Configuration via YAML

Sane defaults for devs without rate limiting knowledge

Configs Hot Reload

DRY Run Mode

Configuration Validation In Staging

ON ERROR

in case redis failed or something inside trafficCTRL errored

a warning notification sent to the devs and requests are

automatically forwarded to backend without limits until fixed