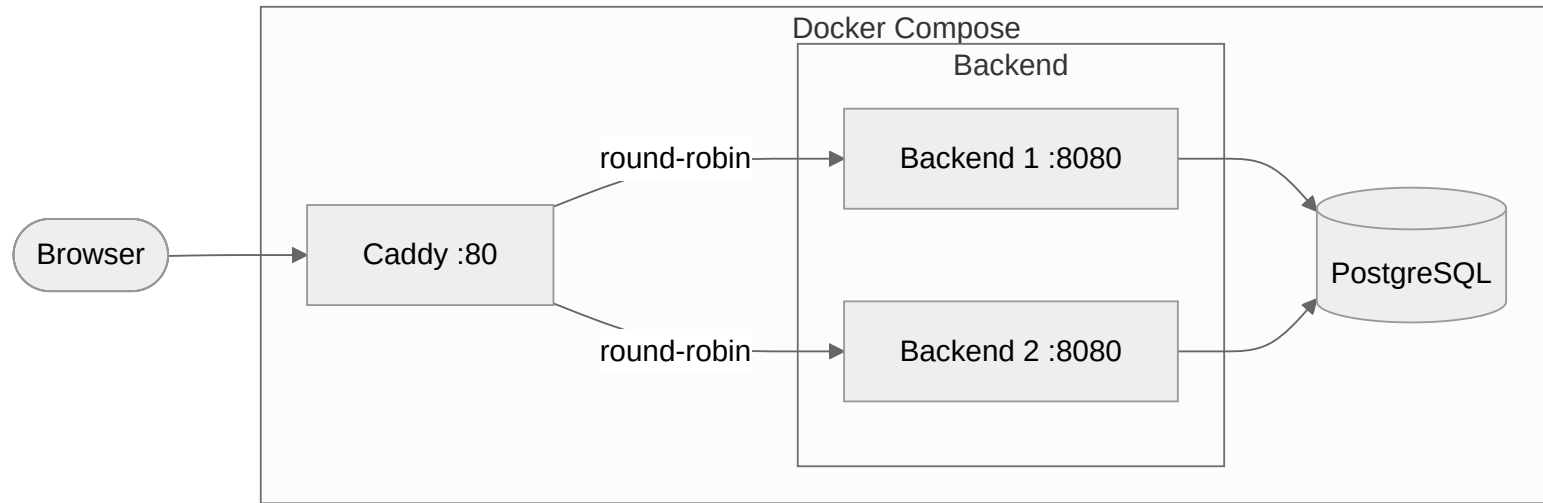


# URL Shortener

## What it does

- Register and log in with email and password
- Paste a long URL → receive a short link (/r/{code})
- Each redirect increments a click counter
- Dashboard lists all your links with click stats

# Architecture



- Two backend instances behind Caddy (round-robin)
- Caddy: Automatic failover if one instance dies

# Tech Stack

## Component

Reverse Proxy / Load Balancer

Backend

Frontend

Database

Auth

Containerization

## Technology

Caddy

Go

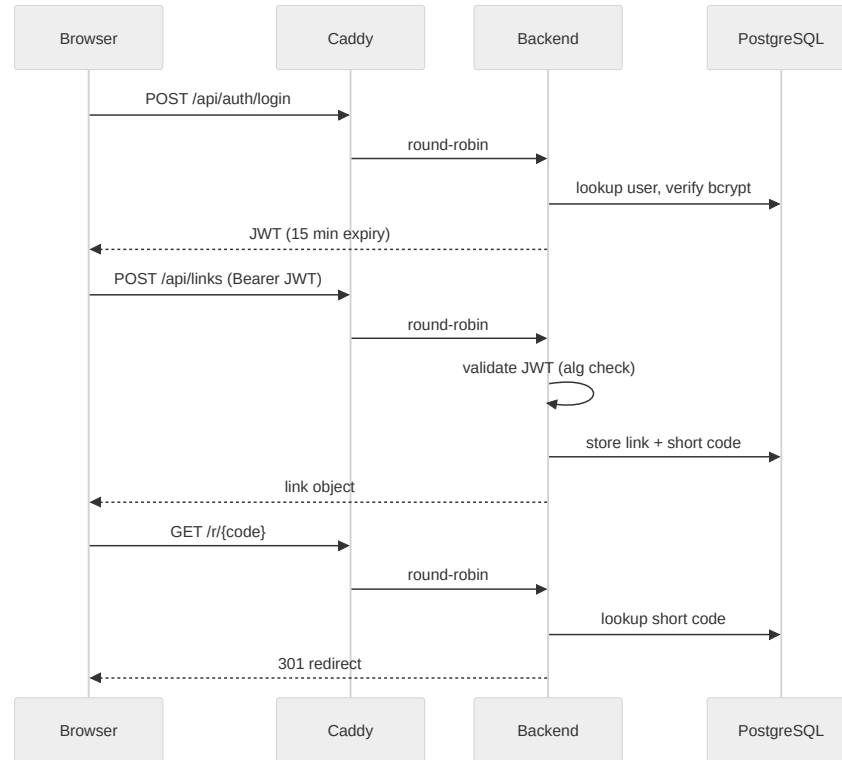
SolidJS + Vite + Bun

PostgreSQL

JWT HS256 + bcrypt

Docker Compose

# Request Flow



# JWT — OWASP Compliance

- none algorithm attack rejected — signing method explicitly verified
- 15-minute expiry with cryptographically random token ID per token
- Long secret length ( $\geq 64$  characters)
- Fingerprint cookie bound to each token — stolen JWT unusable without the cookie
- Logout revokes the token ID in a PostgreSQL denylist — token invalid immediately, not just on expiry

# Performance

| Endpoint        | Requests/sec | Bottleneck        |
|-----------------|--------------|-------------------|
| GET /r/{code}   | 2 200        | PostgreSQL lookup |
| POST /api/links | 1 700        | PostgreSQL insert |

- Measured with hey — 1000 requests, 50 concurrent workers
- Bottleneck is the database round-trip, not Go or the HTTP layer
- Without DB (invalid code → immediate 404): 4 300 req/sec

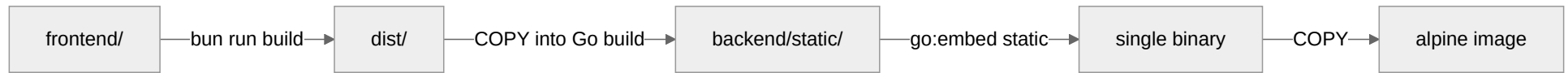
# Code Quality

**Frontend**    `oxlint + oxfmt — strict ruleset`

**Backend**    `golangci-lint — strict ruleset`

- Improves AI produced code and catches bugs

# Build & Deployment



- Bun compiles the SolidJS frontend into static files
- Go embeds the static files directly into the binary via `//go:embed`
- Each backend instance is a single self-contained binary — no separate static file server needed
- Final Docker image is based on Alpine — no Node, Bun, or Go toolchain at runtime