

## Technical Specification (English)

### Technical Specification (English)

#### Architecture & Structure:

#### Modules (Core + Services):

- vpn/: Go-based WireGuard wrapper (client mgmt, routing, QR)
- dns/: Configuration & management for Bind9
- monitor/: Uptime Kuma deployment incl. proxy setup
- awareness/: Python server for microlearning + SQLite storage
- ui/: SvelteKit frontend, admin & user views
- auth/: JWT + TOTP 2FA, role-based access control
- config/: TOML/YAML configuration, backup/restore tools

#### Interfaces:

- REST API (OpenAPI) for all modules
- WebSocket layer for live logs & status messages
- CLI (Go Cobra)
- Optional: MQTT for status events

#### Security:

- Fail2Ban & SSH hardening
- TLS via Caddy (reverse proxy + auto-certificates)
- RBAC, auth tokens, login rate limit

#### DevOps & Update:

- Build: GitHub/GitLab CI (Docker builds, ARM64/x86 images)
- Release tags with changelogs
- Local & centralized update strategy
- GitOps-ready for future cloud deployment

#### Build Structure:

SafeNode/

└ backend/

| └ vpn/

| └ dns/

| └ monitor/

| └ awareness/

| └ main.go

└ frontend/ (SvelteKit)

└ config/

└ docker-compose.yml

└ Dockerfile.\*

└ Caddyfile

Development Environment:

- VSCode, Git, Docker, GitLab/GitHub CI
- Documentation using Markdown & Swagger
- Test setup: localhost + internal reverse proxy

Roadmap (v0.1):

- Finalize API auth via JWT/TOTP
- VPN key generation via Go
- Load awareness content from JSON files
- Fully implement DNS UI + Uptime Kuma integration
- Role separation Admin/User
- Backup/restore module with UI
- Prepare logging via Loki/Grafana