

Pflichtenheft (Deutsch)

Pflichtenheft (Deutsch)

Architektur & Struktur:

Module (Core + Services):

- vpn/: Go-basierter WireGuard Wrapper (Client Mgmt, Routing, QR)
- dns/: Konfiguration & Management für Bind9
- monitor/: Uptime-Kuma Deployment inkl. Proxy-Anbindung
- awareness/: Python-Server für Microlearning + Storage (SQLite)
- ui/: SvelteKit Frontend, Admin & User Views
- auth/: JWT + TOTP 2FA, Role-Based Access Control
- config/: TOML/YAML-Konfiguration, Backup-/Restore-Tools

Schnittstellen:

- REST API (OpenAPI) für alle Module
- WebSocket-Layer für Live-Logs & Statusmeldungen
- CLI (Go Cobra)
- Optional: MQTT für Statusnachrichten

Sicherheit:

- Fail2Ban & SSH-Hardening
- TLS über Caddy (Reverse Proxy + Auto-Zertifikate)
- RBAC, Auth-Tokens, Login-Rate-Limit

DevOps & Update:

- Build: GitHub/GitLab CI (Docker Build, ARM64/x86 Images)
- Releasetags mit Changelogs
- Lokale & zentrale Update-Strategie
- GitOps-ready für späteren Cloud-Betrieb

Buildstruktur:

SafeNode/

└ backend/

| └ vpn/

| └ dns/

| └ monitor/

| └ awareness/

| └ main.go

└ frontend/ (SvelteKit)

└ config/

└ docker-compose.yml

└ Dockerfile.*

└ Caddyfile

Entwicklungsumgebung:

- VSCode, Git, Docker, GitLab/GitHub CI
- Dokumentation per Markdown & Swagger
- Testumgebung: localhost + interner Reverse Proxy

Roadmap (v0.1):

- API-Auth via JWT/TOTP fertigstellen
- VPN-Key-Generierung (via Go)
- Awareness aus JSON-Dateien laden
- DNS UI + UptimeKuma Anbindung vollständig
- Rollentrennung Admin/User
- Backup-/Restore-Modul mit UI
- Logging via Loki/Grafana vorbereiten