GeekSTunnel: System Architecture & Core Mechanics

1. High-Level Tech Stack

GeekSTunnel is built on a high-performance, secure, and modern stack designed for low-latency VPN management and robust network-level enforcement.

Layer	Technology	Purpose
Frontend	Vanilla HTML5/JS (Cyber-Glass)	Premium, responsive management dashboard.
Backend	FastAPI (Python 3.10+)	High-speed asynchronous API and system orchestration.
Database	MySQL (MariaDB)	Persistent storage for users, sessions, and audit logs.
Caching/PubSub	Redis	Real-time state management and inter-process communication.
VPN Kernel	WireGuard	Modern, high-performance UDP-based VPN protocol.
DNS Engine	CoreDNS	Network-level DNS filtering and DoH/DoT blocking.
Proxy/SSL	Nginx (OpenResty)	SSL termination, reverse proxy, and rate limiting.

---

2. Docker Infrastructure

The system is containerized for isolation and easy deployment.

- `nginx`: The gateway. Handles SSL (Let's Encrypt), serves the frontend, and proxies API/WebSocket calls to the backend.
- `redis`: Acts as a high-speed message broker for real-time metrics and session tracking.
- `vpn-dns` **(CoreDNS)**: The "Brain" of DNS resolution. It handles all DNS queries from VPN clients, applying the blacklist and blocking bypass attempts (DoH/DoT).
- `mysql`: Stores all persistent data.

---

3. Security Architecture

3.1 Authentication & Authorization

- **Admin Auth**: Uses **Bcrypt** for password hashing. Sessions are managed via **URLSafeTimedSerializer** (signed cookies) with `HttpOnly`, `Secure`, and `SameSite=Lax` flags.
- **2FA (TOTP)**: Implemented using Google Authenticator-compatible codes.
- **CSRF Protection**: Every state-changing request (POST, PUT, etc.) requires a valid `X-CSRF-Token` fetched via `/auth/csrf`.
- **WebSocket Auth**: The `/ws/stats` endpoint validates the session cookie during the handshake.

3.2 Network Security

- **Firewall (iptables)**: Enforces ACL profiles (Full, Internet-Only, Intranet-Only).
- **DNS Hijacking**: All traffic on port 53 is DNAT'ed to the internal CoreDNS server.
- **Anti-Bypass**: Blocks known DoH provider IPs on port 443 and returns `NXDOMAIN` for DoH canary domains to signal browsers to disable private DNS.

---

4. Networking & Kernel Mechanics

4.1 WireGuard Management (`wg.py`)

- **Key Generation**: Uses `wg genkey` and `wg pubkey` for client keypairs.
- **State Sync**: Uses `wg syncconf` for zero-downtime updates.
- **Zombie Purge**: A background "Homeostatic Sync" removes any peer from the kernel that isn't active in the database.

4.2 Firewall Rules (`firewall.py`)

When you click a button on the UI, the backend executes these low-level rules:

- **NAT/Masquerade**: `iptables -t nat -A POSTROUTING -s 10.50.0.0/24 -j MASQUERADE` (Enables internet access).
- **DNS Hijack**: `iptables -t nat -I PREROUTING -i wg0 -p udp --dport 53 -j DNAT --to-destination 10.50.0.1:53`.
- **ACL Profile (Internet-Only)**:
  - `iptables -A VPN_ACL -s [USER_IP] -d 10.0.0.0/8 -j DROP` (Blocks LAN access).

- `iptables -A VPN_ACL -s [USER_IP] -j ACCEPT` (Allows Internet).

---

5. DNS Resolution & Blocking

- **Service**: **CoreDNS** with the `hosts` and `template` plugins.
- **Blocking**:
  - **Standard**: Domains in `blocked.hosts` return `0.0.0.0`.
  - **Wildcard**: Uses the `template` plugin in `wildcards.conf` to block entire subdomains.
  - **Anti-Bypass**: Returns `NXDOMAIN` for `use-application-dns.net` (Firefox) and `mask.icloud.com` (Apple).
- **Resilience**: If CoreDNS is down, DNS resolution fails for all clients (Fail-Safe), ensuring no unblocked traffic leaks.

---

6. Codebase Organization

- `app/main.py`: The entry point. Initializes the app, database, and background workers.
- `app/auth.py`: Handles admin login, sessions, and 2FA.
- `app/users.py`: Manages the VPN user lifecycle (CRUD).
- `app/wg.py`: The interface to the WireGuard kernel module.
- `app/firewall.py`: Orchestrates `iptables` rules.
- `app/alerts.py`: Manages the DNS blacklist and CoreDNS sync.
- `app/worker.py`: Background tasks for stats and cleanup.

---

7. Deployment Workflow

1. **Host Setup**: Install WireGuard, Docker, and Nginx.
2. **Config**: Set environment variables in `.config`.
3. **Database**: `init_db()` creates tables and the default admin.
4. **Firewall**: `init_firewall_chains()` sets up the `VPN_ACL` chain and NAT.
5. **Sync**: `sync_wireguard_state()` ensures the kernel matches the database.
6. **Nginx**: Start the OpenResty container to serve the UI and proxy API.

GeekSTunnel is designed to be a "Set and Forget" system where the software actively maintains the desired network state in the Linux kernel.