AllInOne-Suite – Benutzerhandbuch

LocalEye · KnowSora · SearchTool · InfraQuill — Schritt für Schritt einrichten und betreiben

Stand 24.11.2026

Eine Datei, offline nutzbar

1 Überblick

Die **AllInOne-Suite** bündelt vier Anwendungen hinter *einer* Adresse und *einem* Login:

LocalEye

Kameras, Live-Ansicht, Aufnahmen, KI-Erkennung (Personen, Fahrzeuge, Tiere), Diebstahlerkennung mit der Griff-Wache, Benachrichtigungen, Benutzerverwaltung. LocalEye ist die Startseite und verwaltet die Zugänge aller Apps.

KnowSora

KI-Chat mit Neuronenkernen, Dokumenten, Projekten und Code-Werkzeugen — lokal (llama.cpp) oder mit Cloud-Anbietern und den KI-Kommandozeilen Claude Code, Codex und Gemini CLI.

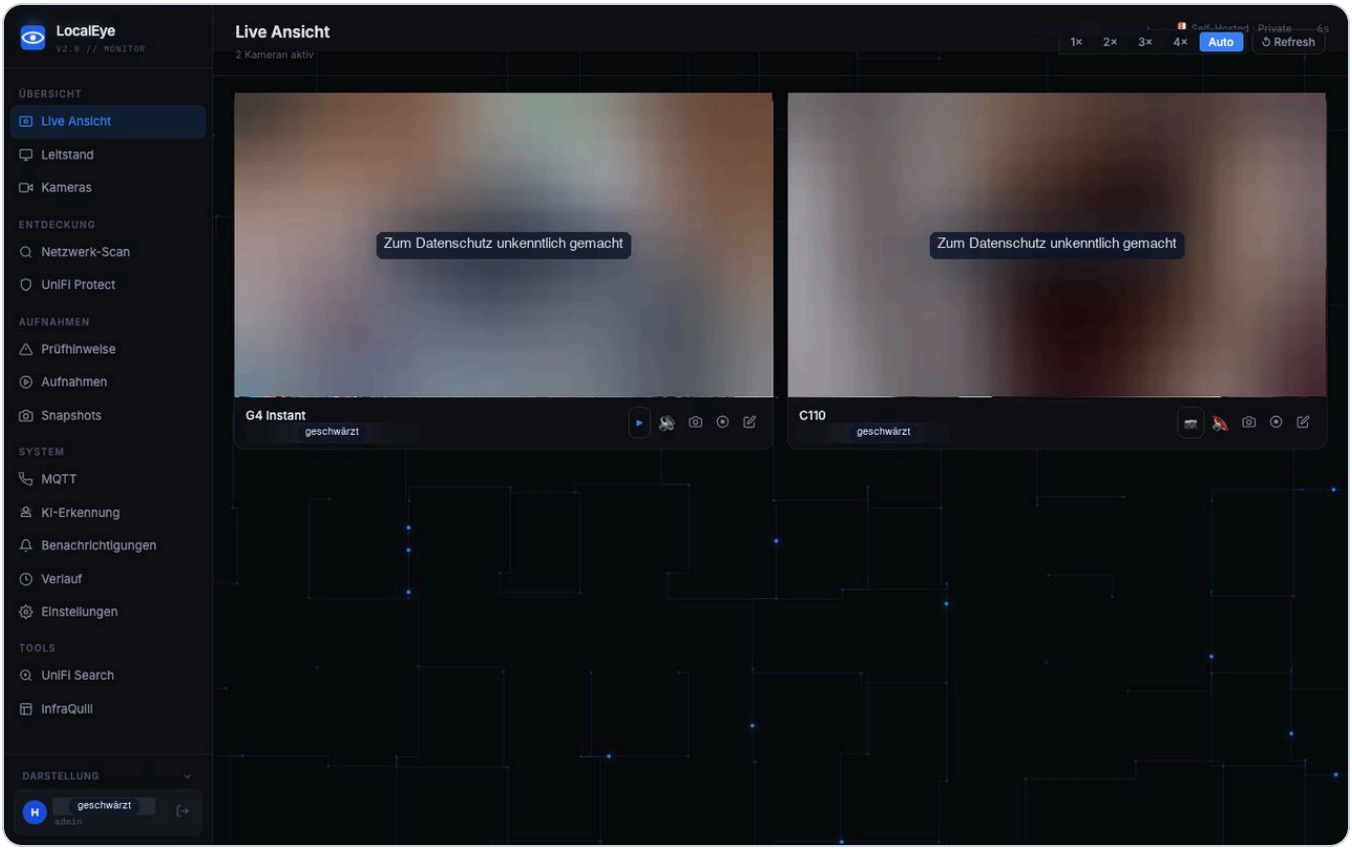
SearchTool

Suche über alle UniFi-Controller: MAC-Adressen, SSIDs, Sites, Gerätenamen — mit Export als CSV/PDF.

InfraQuill

Erzeugt aus Screenshots (UniFi-Geräteliste, CRM-Konto) eine fertige Netzwerk-Dokumentation als Markdown.

Alles läuft in Docker-Containern auf deinem eigenen System (Linux-Server, NAS), **ohne Cloud-Zwang**: Erkennung, Aufnahmen und KI-Auswertung laufen auf deinem Gerät. Bilder gehen nur hinaus, wo du es einrichtest (Meldungen, Prüflink, Fernzugriff). Der Zugriff erfolgt über einen Browser unter `http://<IP>:8987` (bzw. deiner HTTPS-Adresse).



Die Startseite nach dem Login: links die Navigation, rechts die Live-Ansicht (Kamerabilder zum Datenschutz unkenntlich). Unter „Apps“ stehen die Zusatz-Apps.

> **Gut zu wissen:** Kamerabilder, Schnappschüsse, Kennzeichen, IP-/MAC-Adressen, Gerätekennungen und alle persönlichen Inhalte des Betreibers (Projekt-, Chat- und Gedächtnis-Einträge, Namen von Neuronenkernen, Dokumenten, Benutzern und Integrationen) in den Screenshots dieses Handbuchs sind **zum Datenschutz unkenntlich gemacht** (Etikett „geschwärzt“). Alles, was die Apps bei der Installation von sich aus mitbringen, ist sichtbar — in deiner Anlage siehst du an den unkenntlichen Stellen natürlich deine eigenen Inhalte.

Rollen

- **Admin** — darf alles: Kameras, KI, Benutzer, Einstellungen, alle Apps.
- **Viewer** — sieht nur freigegebene Kameras und Apps; KI-Anbieter und Shell-Zugriff in KnowSora vergibt der Admin je Nutzer.
- **Leitstelle**: nur der Prüfzugang der Alarm-Leitstelle unter `/leitstelle/`: keine Kamera, keine Zusatz-App, keine KI (siehe „Alarmverifikation für die Leitstelle“).

Andere Rollen nimmt die Anlage nicht an.

2 Installation

Voraussetzungen

- Linux-System (Ubuntu/Debian, Rocky/Alma, UGREEN- oder Synology-NAS, Proxmox-VM) mit **Docker** und dem Compose-Plugin (`docker compose version`), außerdem `python3` und `curl`.
- x86_64 *oder* ARM64 (z. B. Raspberry Pi 5).
- Mind. 8 GB RAM empfohlen; mehr, wenn lokale KI-Modelle laufen sollen.

Gut zu wissen: Docker installieren (Ubuntu/Debian): erst `sudo apt update && sudo apt install -y ca-certificates curl` (Ubuntu Desktop bringt kein `curl` mit), dann `curl -fsSL https://get.docker.com | sudo sh` und den eigenen Benutzer in die Gruppe `docker` aufnehmen (`sudo usermod -aG docker $USER`, danach neu starten bzw. neu anmelden). Nicht `snap install docker` verwenden – die Snap-Fassung kommt nur an Dateien im Home-Ordner heran.

Tipp: Einfacher mit dem **Install-Skript** (Ubuntu, Debian, Arch Linux): `localeye-install.sh` von der Installationsseite im Shop (localeye.shop/s/allinone-suite) herunterladen und im Terminal `bash ~/Downloads/localeye-install.sh` starten. Es entfernt nach Rückfrage eine Snap-Installation von Docker, installiert Docker samt Compose, legt `/opt/allinone` an, holt Release-ZIP und `manage.sh` aus dem Download-Ordner (fehlt `manage.sh`, nimmt es sie aus der ZIP) und startet auf Wunsch die Einrichtung. Erneut ausführen ist gefahrlos – Erledigtes wird übersprungen.

Erst-Setup in fünf Schritten

1. Installationsordner anlegen, z. B. `/opt/allinone-wl` (NAS: `/volume1/docker/allinone-wl`).
2. Die Release-ZIP (`allinone_whitelabel_release.zip`) und `manage.sh` in diesen Ordner legen.
3. Skript ausführbar machen: `chmod +x manage.sh`.

4. Installation starten: `./manage.sh update` — entpackt die ZIP, legt alle Konfigurationsdateien aus den Vorlagen an, erzeugt **automatisch** alle Geheimnisse (SECRET_KEYs, SSO-Schlüssel), sucht freie Ports und startet die **Anlagen-Oberfläche**. Ohne Lizenzschlüssel hält das Skript hier an und gibt Adresse und **Einrichtungscode** aus.

5. Im Browser (Handy, Mac oder PC) `http://<IP>:8987/anlage/` öffnen, den Code eingeben und den vier Schritten folgen, die 2FA für Administratoren gehört dazu: **Lizenzschlüssel** (Download-Token aus der Bestellbestätigung, nicht die Bestellnummer) → **Admin-Zugang** festlegen → **Einrichtung abschließen** (baut die Images, das Live-Log läuft auf der Seite mit).

```
sudo mkdir -p /opt/allinone-wl && sudo chown $USER /opt/allinone-wl && cd
# ZIP + manage.sh hierher kopieren
chmod +x manage.sh
./manage.sh update
# Ausgabe: [ANLAGE] Einrichtung im Browser: http://<IP>:8987/anlage/   Ein
```

Gut zu wissen: Der Einrichtungscode schützt die Seite im Netz, solange noch kein Admin existiert; am Gerät selbst (Kiosk-Bildschirm) ist er nicht nötig. Code verloren? `./manage.sh config` erzeugt einen neuen und zeigt ihn an.

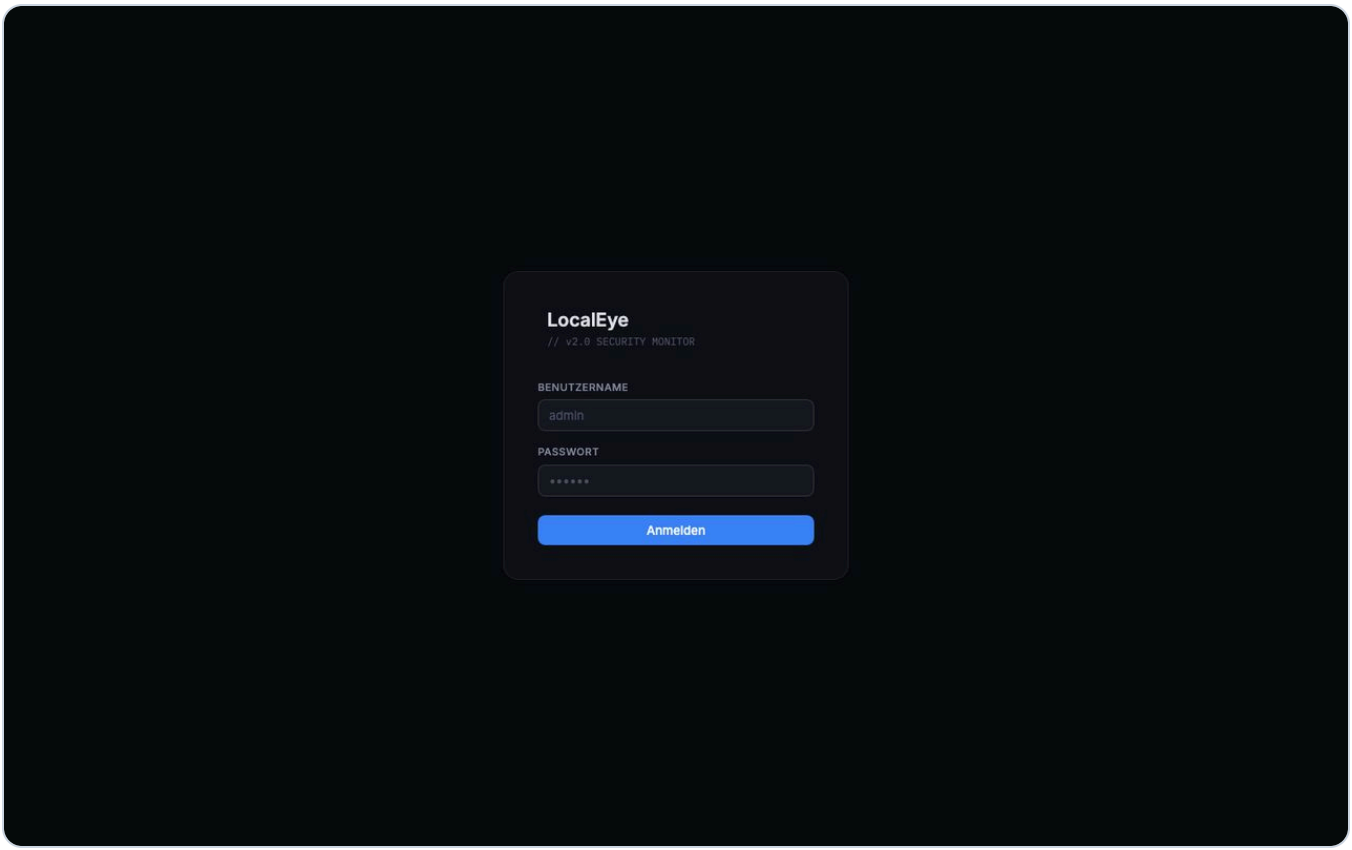
Erst-Login

Es gibt **kein Standardpasswort**: Benutzername und Passwort des ersten Admins legst du selbst bei der Einrichtung unter `/anlage/` fest. Damit meldest du dich im Browser unter `http://<IP>:8987` an — der Login gilt für alle vier Apps.

Rückfall für die Einrichtung per Shell (Lizenz mit `./manage.sh activate`, kein Admin im Browser angelegt): LocalEye erzeugt dann beim ersten Anmeldeversuch ein Zufallspasswort für `admin` in einer Datei, die nur der Installationsbenutzer lesen kann:

```
cat localeye/config/.admin_password_initial
```

LocalEye löscht die Datei selbst, sobald das Erstpasswort an keinem Konto mehr gilt, also nach dem ersten Passwortwechsel (spätestens im nächsten stündlichen Aufräumlauf).



Die Anmeldung gilt für alle vier Apps (Single Sign-on).

Tipp: Die Ports (Standard 8987) vergibt manage.sh beim ersten Start selbst und meidet belegte Ports; die Werte stehen in der `.env`. Status, Protokolle, Update und Sicherung findest du danach im Browser unter `/anlage/` (Kapitel „Anlage verwalten“) — oder per Shell mit `./manage.sh status` und `./manage.sh logs`.

3 LocalEye

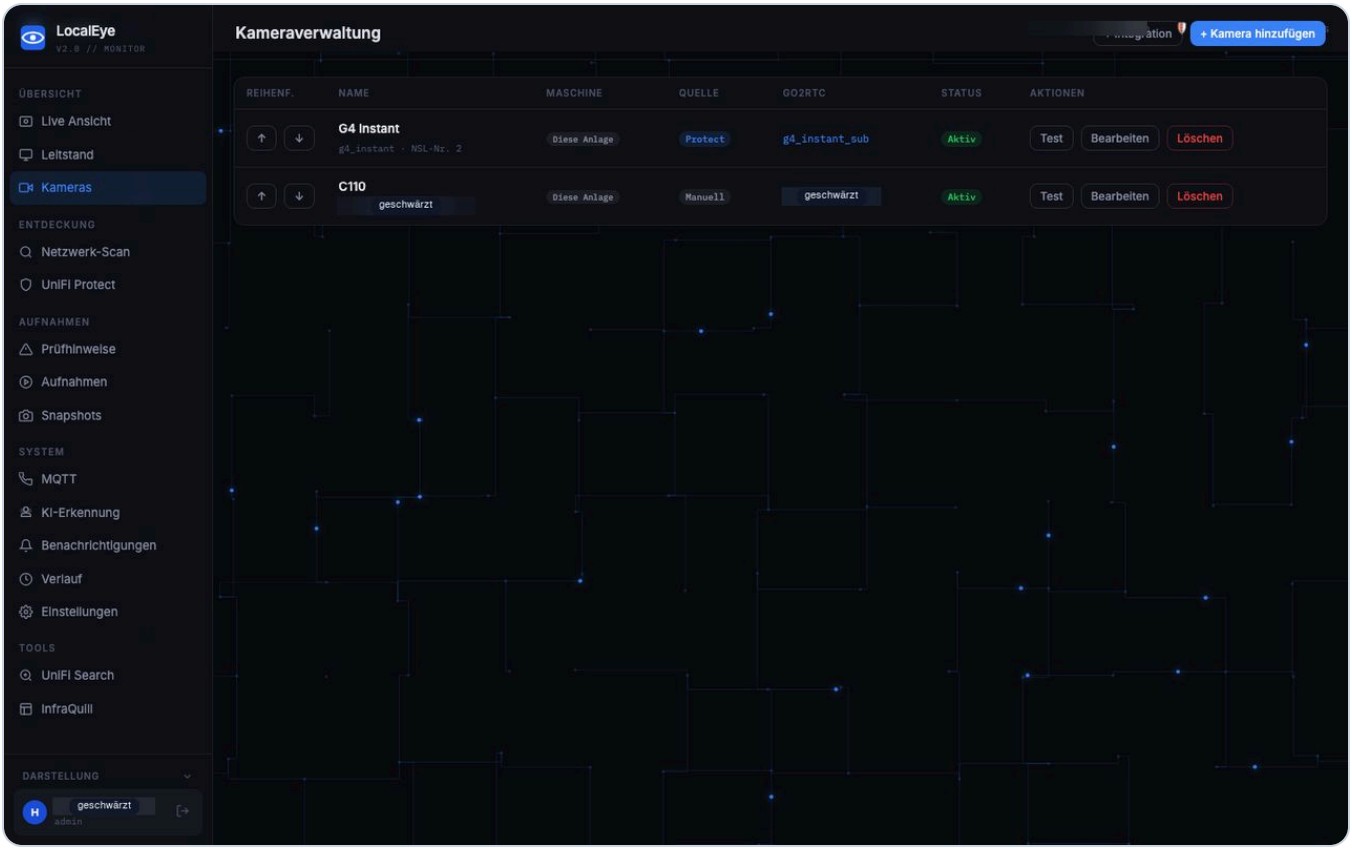
LocalEye ist die Kamera- und Sicherheitszentrale. Die Navigation links zeigt, was du im Alltag brauchst: *Überwachung* (Live-Ansicht, Leitstand, Prüfhinweise, Gelerntes), *Archiv* (Aufnahmen, Snapshots, Meldungsverlauf, Auswertung) und *Apps*. Alles zum Einrichten liegt unter **Einstellungen** unten links: Ein Klick tauscht die Leiste gegen das Einstellungsmenü (Allgemein, Kameras, Erkennung & Meldungen, System), **Zurück** führt wieder heraus. Darunter wählst du die Darstellung (*Auto* folgt dem Gerät, *Hell*, *Dunkel*), den animierten Hintergrund (*Aus*, *Circuit*, *Polygone* – gilt auch in den Apps) und die Sprache; ein Klick auf den eigenen Namen öffnet *Mein Konto* (Passwort, 2FA).

Funktionen ein- und ausschalten

Unter **Einstellungen** → **Funktionen** blendet der Admin ganze Bereiche aus, die er nicht braucht – etwa Leitstand, Scharf/Unscharf, Prüfhinweis „Ware verbergen“, Kassen-Abgleich (Kassenbon), Person verfolgen, Verbund sowie die Zusatz-Apps KnowSora, UniFi Search und InfraQuill. Ausgeschaltete Bereiche verschwinden für alle Nutzer aus dem Menü; ausgeschaltete Apps sind gesperrt. Daten und Einstellungen bleiben erhalten – wieder einschalten genügt. Der Prüfhinweis und der Kassen-Abgleich sind dieselben Schalter wie bisher (kein zweiter Schalter an anderer Stelle).

Bei den **Zusatz-Apps** stoppt *Aus* zusätzlich deren Container (spart Arbeitsspeicher und Rechenzeit), *Ein* startet sie wieder – unter dem Schalter steht, ob der Container läuft. Updates und Neustarts lassen ausgeschaltete Apps aus. Das lokale Bildmodell, mit dem LocalEye Kamerabilder beschreibt, läuft bei ausgeschaltetem KnowSora weiter.

3.1 Kamera hinzufügen



Kameraverwaltung: Reihenfolge, Quelle, go2rtc-Stream, Status und Aktionen je Kamera.

1. Links **Kameras** öffnen und rechts oben **+ Kamera hinzufügen** klicken.

2. Name vergeben und die **RTSP-Adresse** eintragen, z. B. `rtsp://benutzer:passwort@192.168.1.50:554/stream1`. Herstellerbeispiele: Tapo `/stream1` (HD) bzw. `/stream2`, Hikvision `/Streaming/Channels/101`, Reolink `/h264Preview_01_main`, Dahua `/cam/realmonitor?channel=1&subtype=0`.

3. Optional einen zweiten, kleineren Stream (Sub-Stream) für die Kachelansicht angeben — spart Bandbreite.

4. **Test** klicken: LocalEye prüft die Verbindung und zeigt Auflösung und Codec.

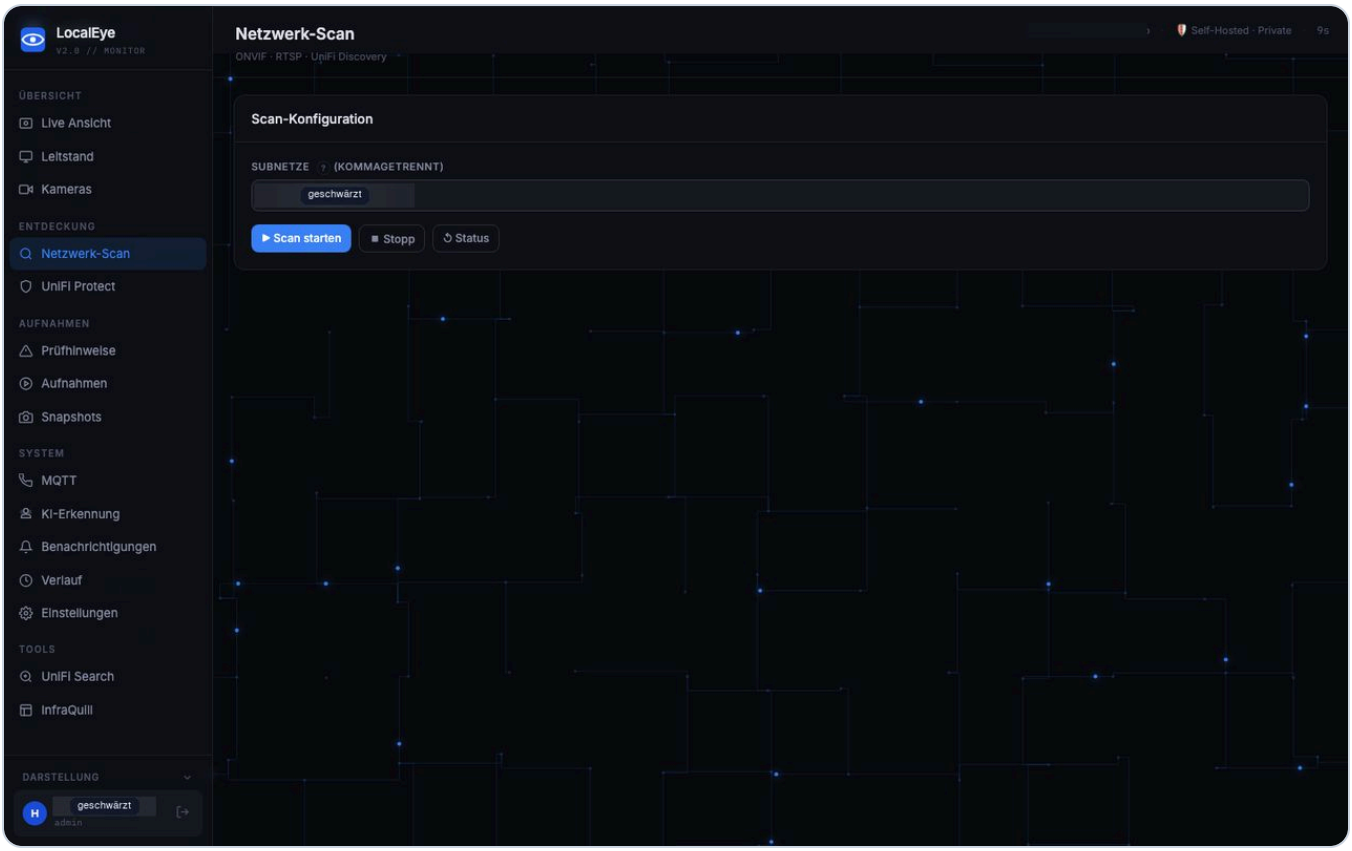
5. Speichern. Die Kamera erscheint sofort in der Live-Ansicht; mit den Pfeilen links ordnest du die Kacheln.

Gut zu wissen: Streams laufen über den eingebauten Vermittler **go2rtc**. Er bringt das Kamerabild in den Browser (WebRTC, MSE oder MJPEG) – du brauchst keine Plugins. Das Video selbst bleibt dabei im Codec der Kamera: Haupt- und Unterstream an der Kamera auf **H.264** stellen und „Smart Codec“ (H.264+/H.265+) ausschalten, denn H.265 spielt nicht jeder Browser ab (siehe 3.3). Die Spalte „go2rtc“ zeigt den internen Stream-Namen.

Tipp: Schwenken/Neigen/Zoom (PTZ): Im Feld *ONVIF-URL* steht `http://benutzer:passwort@<IP>/onvif/device_service` – der Import per Scan trägt sie selbst ein. Leitet die Kamera auf HTTPS um oder spricht sie nur alte TLS-Versionen, kommt LocalEye trotzdem durch. Fehlen im Unterstream die Zugangsdaten, übernimmt LocalEye sie beim Speichern aus dem Hauptstream. Sonderzeichen im Passwort bei Eingabe von Hand kodieren (`@` → `%40`, `#` → `%23`).

3.2 Netzwerk-Scan & UniFi Protect

Kennst du die Adresse deiner Kamera nicht, hilft der **Netzwerk-Scan**: Er sucht das lokale Netz nach Geräten mit Kamera-Ports (RTSP 554, ONVIF) ab und schlägt sie zur Übernahme vor.



Netzwerk-Scan: gefundene Geräte lassen sich direkt als Kamera anlegen.

Übernahme per Scan: Bei gefundenen ONVIF-Kameras fragt LocalEye, ob die Streams automatisch per ONVIF aufgelöst werden sollen. Mit *OK* und **einmal Benutzer und Passwort** trägt LocalEye Haupt- und Unterstream, Aufnahme-Stream und ONVIF-URL (für Schwenken/Neigen/Zoom) samt Zugangsdaten selbst ein. Dieselben Zugangsdaten gelten für ONVIF und das Videobild (RTSP) – an der Kamera also ein Konto anlegen, das für beides mit gleichem Namen und Passwort gilt. Klappt ONVIF nicht, nennt die Meldung den Grund je Port (etwa „Anmeldung abgelehnt“, „verweigert“ oder „HTTPS: …“); mit *Manuell per RTSP fortfahren* setzt LocalEye die bekannten Stream-Pfade des Herstellers ein und fragt die Zugangsdaten ab.

Gut zu wissen: Dahua (auch Amcrest, Imou, Lorex): zwei Benutzerlisten unter *Einstellungen* → *System* → *Konto* – Reiter *Benutzer* für das Videobild und Reiter *ONVIF-Benutzer* für PTZ und Ereignisse (je nach Firmware unter *Sicherheit*). Den LocalEye-Benutzer in **beiden** mit gleichem Passwort anlegen; die ONVIF-Anmeldebestätigung bleibt eingeschaltet. Leitet die Kamera auf HTTPS um, folgt LocalEye selbst.
Hikvision (auch HiLook, ANNKE, EZVIZ): Der normale Benutzer gilt nicht für ONVIF – unter *Konfiguration* → *Netzwerk* → *Erweiterte Einstellungen* → *Integrationsprotokoll* ONVIF aktivieren und einen ONVIF-Benutzer mit gleichem Namen und Passwort anlegen. Nach mehreren Fehlanmeldungen sperrt die Kamera die anfragende IP eine Weile, auch für das Videobild – Kamera neu starten oder die Sperrzeit abwarten. *Hik-Connect-/EZVIZ-Modelle* (z. B. DS-2CV…) haben keine Weboberfläche: Benutzer `admin`, Passwort = 6-stelliger Bestätigungscode vom Aufkleber; ONVIF ist ab Werk an. Sie senden ab Werk oft H.265 – der Import meldet das und nennt den Weg zu H.264 (Hik-Connect-App oder iVMS-4200). Ereignisse laufen dort über ONVIF und werden beim Import selbst eingerichtet. Gegensprechen geben diese Modelle über ONVIF nicht aus (auch die Hik-Connect-App bietet es dort nicht) – der Import richtet es deshalb nicht ein. Das Steuerkreuz erscheint nur bei Kameras mit Motor (PTZ über ONVIF).
Bild drehen: Hängt eine Kamera kopfüber oder seitlich, im Kamera-Dialog „Bild drehen“ (180°, 90°) und bei Bedarf „Spiegeln“ wählen. Livebild, Schnappschüsse, KI und Benachrichtigungsbilder erscheinen dann richtig herum; Aufnahmen werden beim Abspielen gedreht.
TP-Link Tapo: Zugang ist das Kamera-Konto aus der Tapo-App (*Kamera* → *Erweitert* → *Kamera-Konto*), nicht das TP-Link-Cloud-Konto.
Alle Hersteller: Haupt- und Unterstream auf **H.264** stellen und „Smart Codec“ (H.264+/H.265+) ausschalten – H.265 spielt der Browser oft nicht ab (etwa unter Linux oder in Firefox), das Live-Bild bliebe schwarz.

Hikvision Hik-Connect-Kamera ohne App einbinden

Kameras der Hik-Connect-/EZVIZ-Reihe (z. B. *DS-2CV…*) haben keine Weboberfläche. LocalEye bindet sie trotzdem direkt ein – die Hik-Connect-App ist dafür **nicht nötig**.

1. Kamera per **Netzwerkkabel** anschließen – sie holt sich ihre Adresse selbst (DHCP).
2. Den **Bestätigungscode** (Verification Code, 6 Großbuchstaben) vom Aufkleber an der Kamera oder auf der Verpackung ablesen.
3. **Einstellungen** → **Netzwerk-Scan**: Die Kamera erscheint samt Modell und Firmware (auch über den Hikvision-Rundruf SADP). **Import** → Benutzer `admin`, Passwort = **Bestätigungscode**.
4. LocalEye trägt Haupt- und Unterstream ein, richtet die Kamera-Ereignisse (ONVIF) ein und meldet das tatsächliche Videoformat (ab Werk oft H.265).

Gut zu wissen: Die App braucht es nur für **WLAN** (die WLAN-Zugangsdaten bekommt die Kamera nur über die Hik-Connect-App) und für Einstellungen wie Videoformat oder Firmware-Update. Ohne Anbindung an die App bleibt die Kamera außerhalb der Hikvision-Cloud. Meldet der Scan „nicht aktiviert“, die Kamera einmal mit der App oder dem Hikvision-Werkzeug SADP aktivieren. **Gebrauchte Kamera:** bei eingeschalteter Kamera die Reset-Taste etwa 15 Sekunden halten, danach gilt wieder der Code vom Aufkleber. Nach mehreren falschen Anmeldungen sperrt die Kamera die anfragende IP eine Weile – ein Neustart der Kamera hebt das auf. Gegensprechen wird bei diesen Modellen nicht eingerichtet.

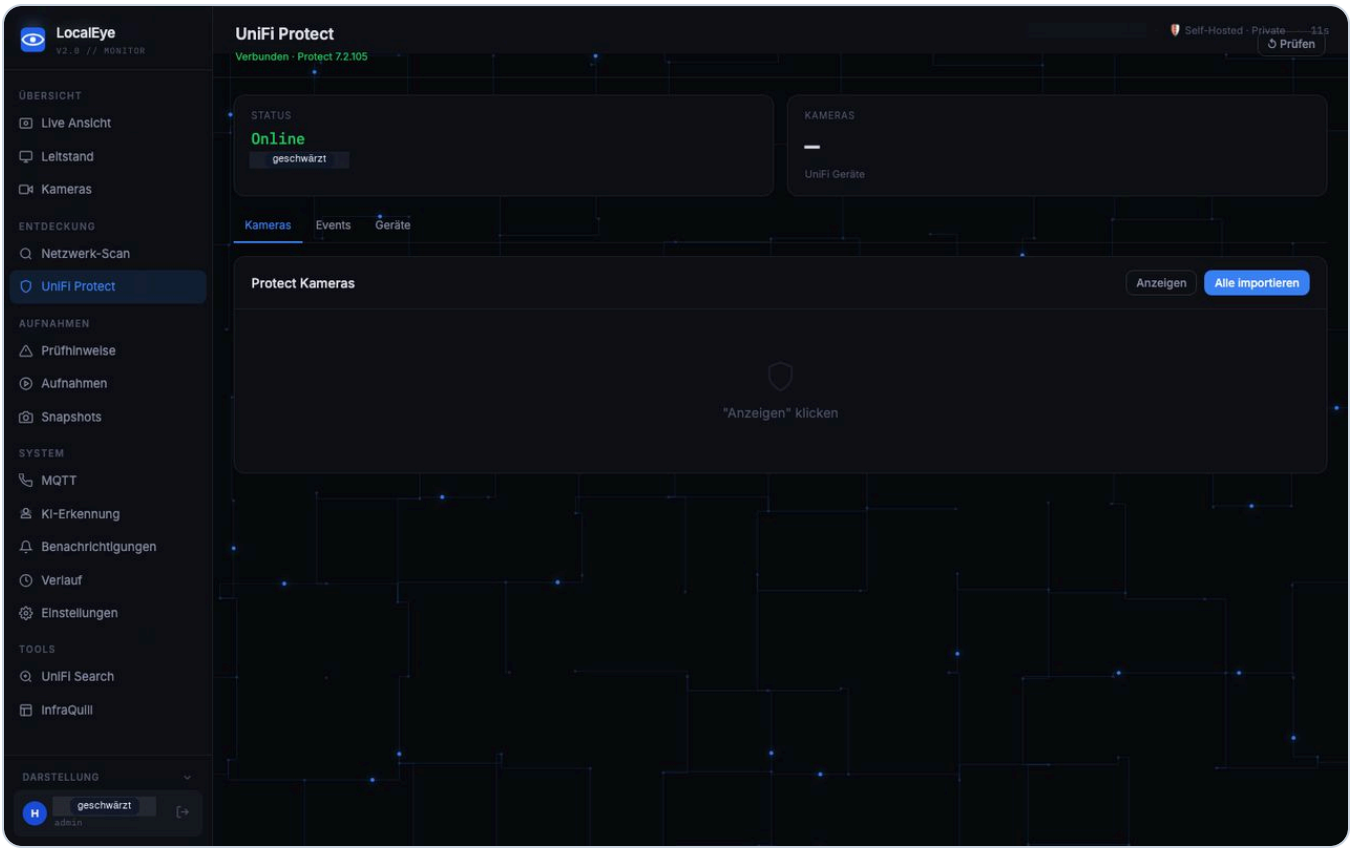
UniFi Protect: LocalEye spricht die offizielle *Protect Integration API* deiner Konsole (UDM/UNVR, Protect ab Version 5) an. Angemeldet wird **nicht** mit Benutzername und Passwort, sondern mit einem **API-Schlüssel** — damit liest LocalEye alle Kameras aus und legt sie mit einem Klick an (Quelle „Protect“ in der Kameraverwaltung).

1. In der **Protect-Weboberfläche** deiner Konsole anmelden und **Einstellungen** → **Integrationen** öffnen (je nach Protect-Version unter *Control Plane* → *Integrations*).
2. Bei **API Keys** auf **Create API Key** klicken, einen Namen wie `LocalEye` vergeben und bestätigen.

3. Den angezeigten Schlüssel **sofort kopieren** — Protect zeigt ihn nur dieses eine Mal. Der Schlüssel hat die Rechte des angemeldeten Kontos; ein Konto mit Zugriff auf die Kameras genügt, Administratorrechte sind nicht nötig.

4. In LocalEye unter **System** → **Einstellungen**, Abschnitt **Verbindungen**: **Protect Host** (IP oder Name der Konsole, z. B. `192.168.1.1`) und **Protect API Key** eintragen und speichern. Das Feld zeigt danach „Key gesetzt“; zum Ändern einfach einen neuen Schlüssel eingeben.

5. Links **UniFi Protect** öffnen und **Prüfen** klicken: Der Status wechselt auf „Verbunden“ samt Protect-Version, darunter erscheinen alle Kameras zur Übernahme.



Gut zu wissen: Der Schlüssel wird getrennt von den übrigen Einstellungen abgelegt und nie wieder im Klartext angezeigt. Wird er in Protect gelöscht oder das zugehörige Konto deaktiviert, meldet **Prüfen** einen Fehler — dann einfach einen neuen Schlüssel anlegen und eintragen. Das selbstsignierte Zertifikat der Konsole ist kein Hindernis.

Gut zu wissen: LocalEye merkt sich beim ersten Kontakt das Zertifikat der Protect-Konsole. Ändert es sich, sendet LocalEye den API-Schlüssel nicht mehr dorthin, und der Anlagenwächter meldet eine Störung. Sie geht an die eingerichteten Kanäle, nicht an die Leitstelle. Nach einem Neustart erscheint sie erst wieder beim nächsten Kontakt mit der Konsole. Nach einem Tausch oder Zurücksetzen der Konsole den API-Schlüssel in den Einstellungen neu eingeben; danach gilt das neue Zertifikat.

Mehrere Konsolen: Hat die Anlage mehr als eine UniFi-Konsole, zum Beispiel in einem zweiten Gebäude, kommt jede weitere Konsole mit eigenem API-Schlüssel dazu. Die Konsole oben (Protect Host) bleibt die erste und ändert sich dabei nicht.

1. Den API-Schlüssel in der Protect-Weboberfläche *der weiteren Konsole* anlegen, wie oben beschrieben.

2. In LocalEye unter **Verbindungen**, Abschnitt **Weitere Protect-Konsolen**, auf **+ Weitere Konsole** klicken, Name (z. B. „Lager“), Adresse und API-Schlüssel eintragen und **Konsole speichern**. LocalEye prüft die Verbindung gleich und zeigt das Ergebnis in der Zeile der Konsole.

3. Links **UniFi Protect** öffnen und oben die Konsole wählen. Dann **Anzeigen** und **Alle importieren**: Die Kameras dieser Konsole kommen dazu, Kameras mit gleichem Namen aus einer anderen Konsole bleiben unberührt.

Gut zu wissen: Jede Kamera fragt danach ihre eigene Konsole nach Schnappschüssen und Gegensprechen. Die Ereignisse aller Konsolen stehen gemeinsam unter *Events*, mit dem Namen der Konsole davor. Zertifikat und Schlüssel gelten je Konsole: Ändert sich das Zertifikat einer Konsole, betrifft die Störung nur sie. Eine neue Adresse verlangt den Schlüssel neu. Wird eine Konsole entfernt, bleiben ihre Kameras und zeigen weiter Bild über ihre gespeicherten Stream-Adressen; Schnappschüsse und Gegensprechen über Protect entfallen dann für sie.

Konsole an einem anderen Standort: Erreicht die Anlage die Konsole nicht im eigenen Netz, gibt es zwei Wege.

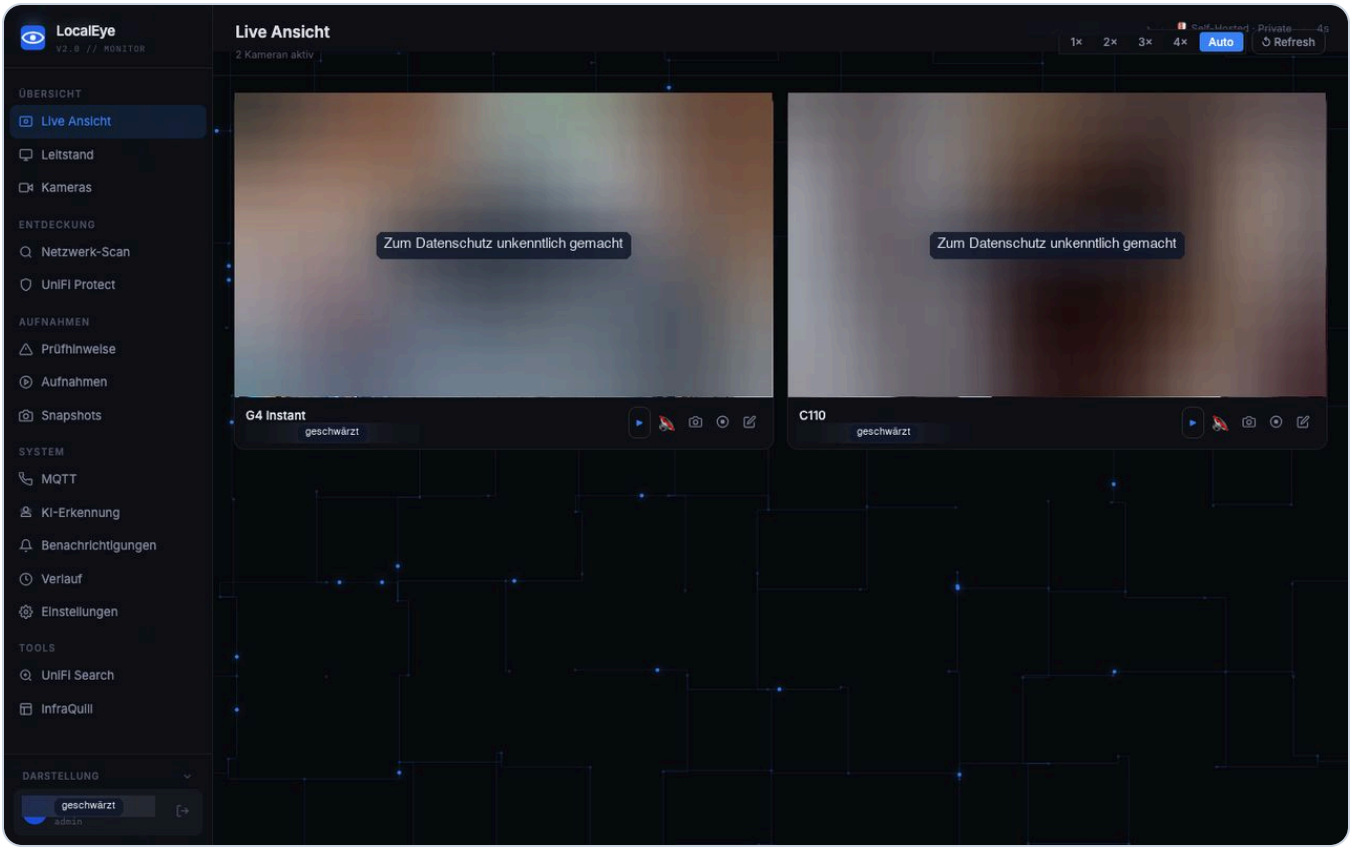
1. **Voller Umfang über VPN (empfohlen):** Beide Standorte per Standort-VPN verbinden, zum Beispiel UniFi *Site Magic* zwischen zwei UniFi-Gateways oder WireGuard bzw. IPsec. Die Anlage muss die lokale Adresse der Konsole erreichen: Port 443 für die Schnittstelle, 7441 für die Videoströme. Dann die Konsole mit Verbindung **Lokale Adresse** und ihrem lokalen API-Schlüssel eintragen. Live-Video, Aufnahme, KI und Gegensprechen laufen wie bei der ersten Konsole.

2. **Nur Bilder über die UniFi-Cloud:** Verbindung **UniFi-Cloud** wählen, den Link der Konsole aus dem Browser einfügen (`https://unifi.ui.com/consoles/…/protect/dashboard`) und einen **Site-Manager-Schlüssel** eintragen: auf unifi.ui.com unter *API → Create API Key*, mit dem Konto, dem die Konsole gehört. Ein lokaler Protect-Schlüssel gilt dort nicht. Die Konsole braucht Firmware 5.0.3 oder neuer.

Gut zu wissen: Über die UniFi-Cloud kommen nur Schnappschüsse, die KI auf diesen Bildern und die Zustände der Kameras. Live-Video, Aufnahme und Gegensprechen gehen nicht, weil die Videoströme auf die Adresse der Konsole in ihrem eigenen Netz zeigen. Ubiquiti erlaubt 100 Anfragen je Minute und Konsole; LocalEye verteilt sie selbst auf die Kameras (bei 7 Kameras etwa alle 6 Sekunden ein Bild je Kamera). Wird später ein VPN eingerichtet, die Konsole auf **Lokale Adresse** umstellen, den lokalen Schlüssel eintragen und die Kameras erneut importieren: Sie behalten ihre Einstellungen und bekommen Video und Aufnahme.

Gut zu wissen: LocalEye folgt bei keiner Konsole einer Weiterleitung. Wer als Adresse etwa `unifi.ui.com` einträgt, bekommt einen Hinweis statt einer Verbindung; der API-Schlüssel geht dabei nirgends hin.

3.3 Live-Ansicht & Leitstand



Live-Ansicht: pro Kamera Play/Pause, Ton, Schnappschuss, Aufnahme starten, Bearbeiten. Oben rechts das Kachelraster (1×–4×, Auto). Kamerabilder zum Datenschutz unkenntlich.

Gut zu wissen: Oben rechts in jeder Kachel steht, wie das Bild übertragen wird: **RTC** (WebRTC, Standard in Chrome, Safari und Edge), **MSE** (Video-Decoder des Systems; Firefox startet hier) oder **MJPEG** (Einzelbilder, die go2rtc selbst erzeugt – läuft in jedem Browser, aber mit weniger Bildern pro Sekunde und ohne Ton). Kommt 12 Sekunden lang kein Bild, wechselt LocalEye selbst zur nächsten Stufe. Steht dauerhaft **MJPEG** da, kann der Browser den Codec der Kamera nicht – meist H.265: an der Kamera auf H.264 umstellen. H.265 können meist nur Safari sowie Chrome und Edge unter Windows und macOS mit passender Grafikhardware.

KI-Rahmen in Echtzeit

Ist für eine Kamera die KI eingeschaltet und läuft die Kachel als **Video** (oder im Vollbild), folgen die farbigen Erkennungsrahmen der Person in Echtzeit: Solange jemand hinsieht, wertet die KI den Unterstream etwa viermal pro Sekunde aus; Rahmen, die älter als gut eine Sekunde sind, verschwinden. Ohne Zuschauer endet das nach wenigen Sekunden von selbst. Gleichzeitig gilt das für höchstens vier Kameras. Alarme, Aufnahmen und Meldungen laufen unverändert im eingestellten KI-Takt und haben bei der Rechenzeit Vorrang. In der Einzelbild-Ansicht bleibt es beim bisherigen Takt.

Vollbild und Zoom

Auch das Standbild lässt sich im Vollbild mit dem Mausrad oder mit zwei Fingern vergrößern. Vergrößert lässt sich der Ausschnitt mit gedrückter Maustaste verschieben, ein Doppelklick stellt zurück.

Gegensprechen (Push-to-Talk / Tippen zum Sprechen)

Kameras mit Lautsprecher lassen sich aus der Live-Kachel ansprechen: Neben dem Ton-Knopf erscheint ein **Mikrofon** 🎙 — am Computer *gedrückt halten* zum Sprechen, loslassen beendet die Verbindung; am Handy/Tablet *einmal tippen* zum Sprechen und *erneut tippen* zum Beenden (nach 90 Sekunden endet die Verbindung automatisch). Das Mikrofon des Browsers wird per WebRTC an den Stream-Vermittler go2rtc gegeben und von dort in den Rückkanal der Kamera; bei UniFi Protect läuft der Ton über LocalEye direkt zur Kamera.

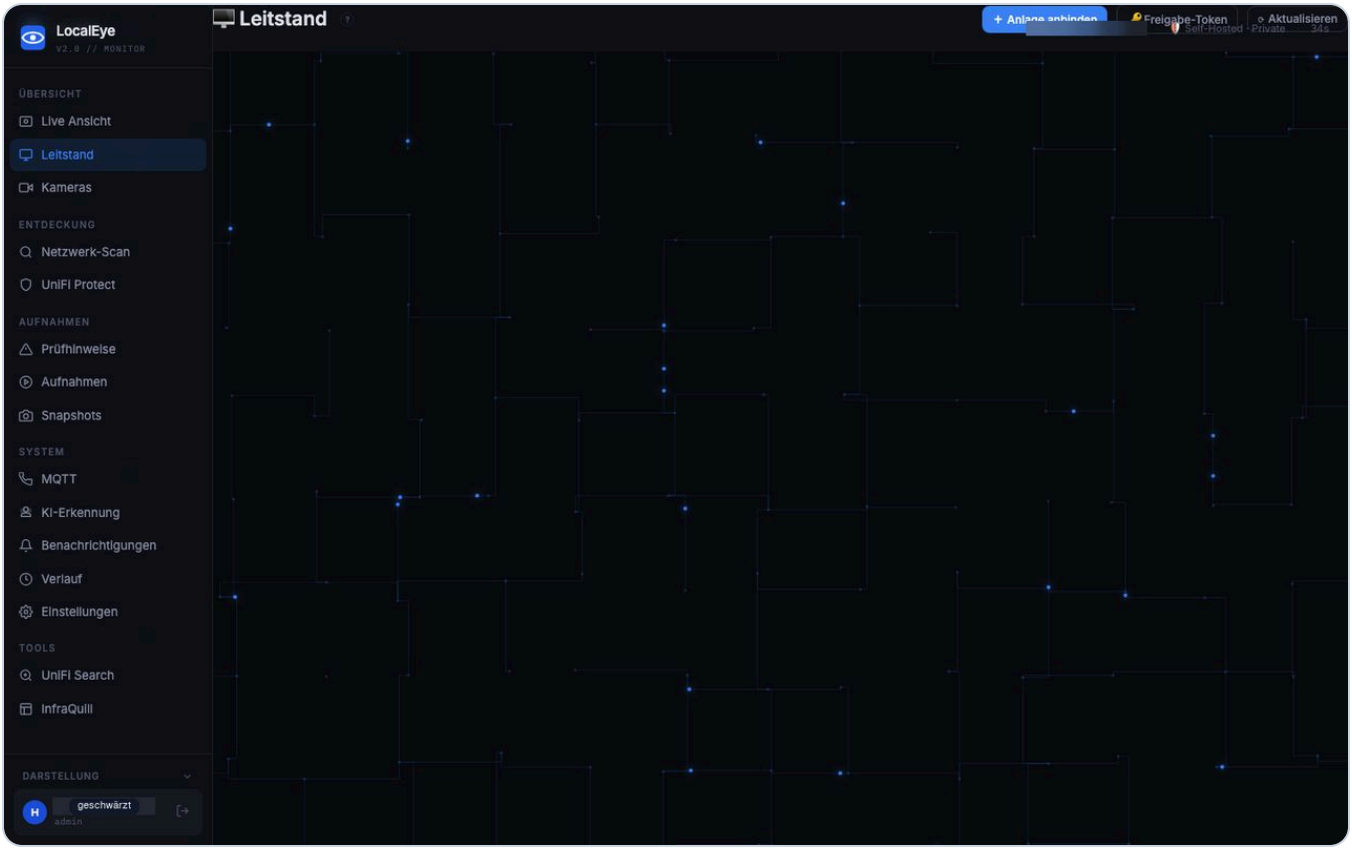
1. **Kameras** → **Bearbeiten** → Abschnitt **Gegensprechen (Rückkanal)**: Typ wählen — *TP-Link Tapo* (Cloud-Passwort deines TP-Link-Kontos, Benutzer leer bzw. bei neuer Firmware „admin“), *Hikvision (ISAPI)* (Kamera-Login), *RTSP-Rückkanal* (ONVIF Profile T, komplette rtsp://-URL) oder *UniFi Protect* (keine Zugangsdaten nötig).

2. Host/IP der Kamera und Zugangsdaten eintragen, **Speichern**. LocalEye legt im Hintergrund einen eigenen Rückkanal-Stream an (der normale Live-Stream bleibt unverändert).

3. In der Live-Ansicht das Mikrofon gedrückt halten (Computer) bzw. antippen (Handy/Tablet); beim ersten Mal fragt der Browser nach der Mikrofon-Freigabe. Rot = verbindet, Grün = verbunden, sprechen. Am Handy beendet ein zweiter Tipp.

Gut zu wissen: Mikrofonzugriff gibt es im Browser nur über **HTTPS**. Der Knopf ist Admins vorbehalten; solange der Rückkanal nicht eingerichtet ist, erscheint er ausgegraut und führt zum Kamera-Dialog. **UniFi-Protect-Kameras** (G4 Instant u. a.) wählen den Typ *UniFi Protect* — ohne Zugangsdaten, LocalEye nutzt die bestehende Protect-Anbindung. Zur Fehlersuche die Seite mit `#dbgtalk` am Ende der Adresse öffnen: unten erscheint ein Protokoll des Gegensprechens (ein Screenshot genügt für die Diagnose).

Der **Leitstand** ist die Ansicht für den Wachraum: große Kacheln, Ereignisliste, Alarm-Quittierung und die SIA-Anbindung an Leitstellen (siehe Benachrichtigungen).

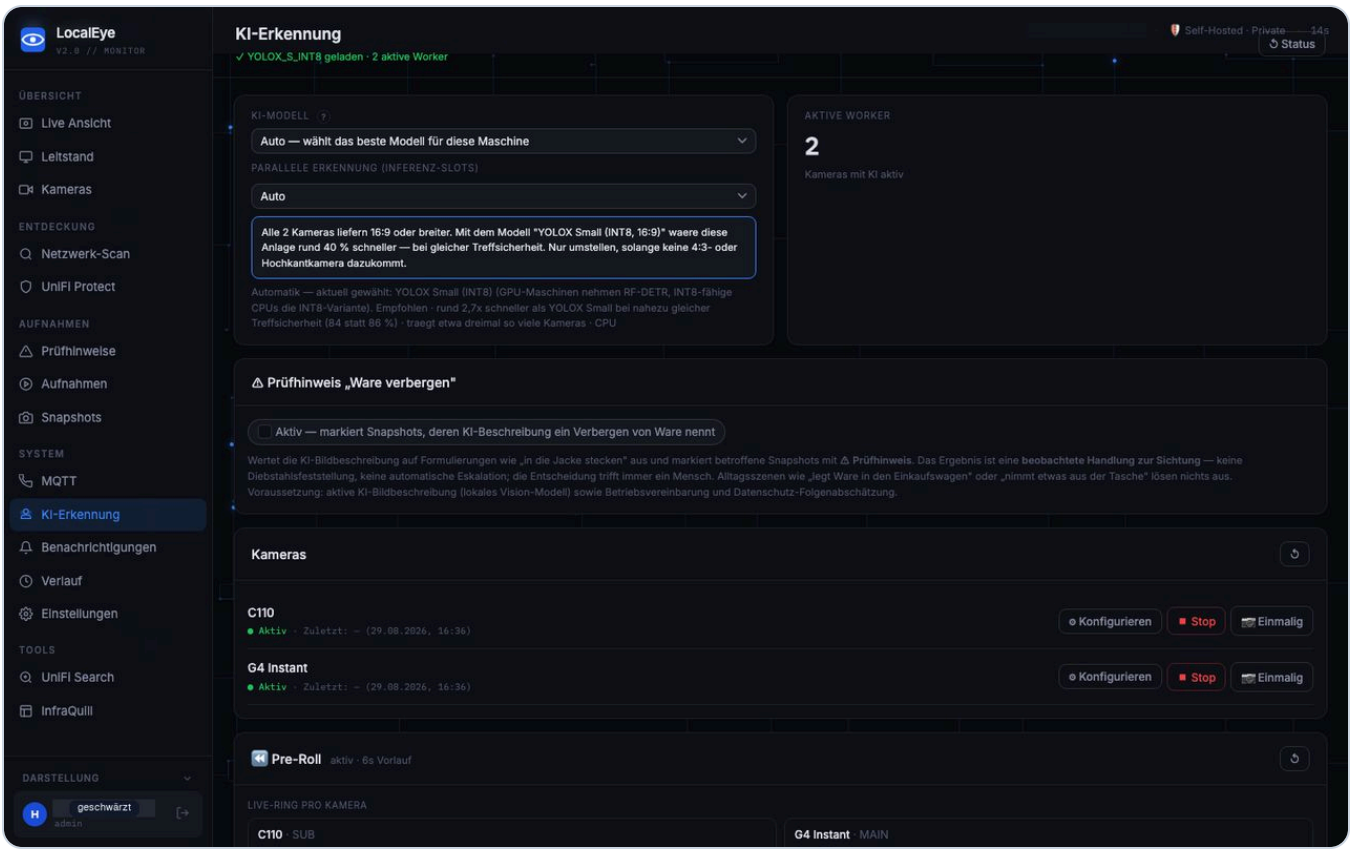


3.4 KI-Erkennung einrichten

Die KI erkennt Personen, Fahrzeuge und Tiere — lokal auf deinem System (YOLOX auf CPU; RF-DETR mit GPU). Erkennungen lösen Aufnahmen, Snapshots und Benachrichtigungen aus. Kennzeichen liest LocalEye nicht: Die Kennzeichenerkennung ist bewusst entfernt (siehe „Schnappschuss-Token“).

Achtung: Die Bildbeschreibung ist ein Anhaltspunkt, kein Protokoll. Zu jeder Erkennung schreibt ein lokales Bildmodell zwei bis drei Sätze: was die Person tut (Haltung, Richtung, was sie trägt) und was drumherum steht. Weil das Modell klein genug sein muss, um auf Ihrer eigenen Hardware zu laufen, erfindet es gelegentlich Einzelheiten — ein Tier, ein Kleidungsstück. Nutzen Sie die Sätze zum schnellen Durchblättern des Verlaufs; was wirklich passiert ist, zeigt die Aufnahme.

Achtung: Hauptstream kostet Rechenzeit. Er holt je Takt ein Bild in voller Auflösung über den Restream — bei einer 4-MP-Kamera ist das eine Videodekodierung alle paar Sekunden. Auf schwacher Hardware oder mit vielen Kameras ist das spürbar. Deshalb gezielt für die Kamera einschalten, auf die es ankommt (Einfahrt, Kasse), nicht überall. Schlägt der Weg fehl, nimmt die Anlage automatisch wieder die Vorgabequelle — die Erkennung fällt nie aus.



KI-Erkennung: je Kamera an/aus, Status des Erkennungs-Workers und Konfiguration.

1. Links **KI-Erkennung** öffnen und bei der gewünschten Kamera auf **Konfigurieren** klicken.

2. **Zonen zeichnen:** Mit gedrückter Maustaste (am Handy mit dem Finger) einen Rahmen ins Kamerabild ziehen – fertig ist die Zone; nur Bewegung in der Zone zählt. Ecken ziehen ändert die Größe, Ziehen in einer Zone des gewählten Zonentyps verschiebt sie (so lässt sich eine Ausschlusszone in eine Erkennungszone legen), mit gedrückter ⇧-Taste lässt sich eine Ecke schräg ziehen. *Entf* löscht die angeklickte Zone, *Rückgängig* nimmt den letzten Schritt zurück. Für unregelmäßige Flächen gibt es *Freie Form* (Eckpunkte anklicken, dann *Zone abschließen*). Ohne Zone gilt das ganze Bild. Mit *Ausschlusszone* blendest du Bereiche aus (Straße, Nachbargrundstück).

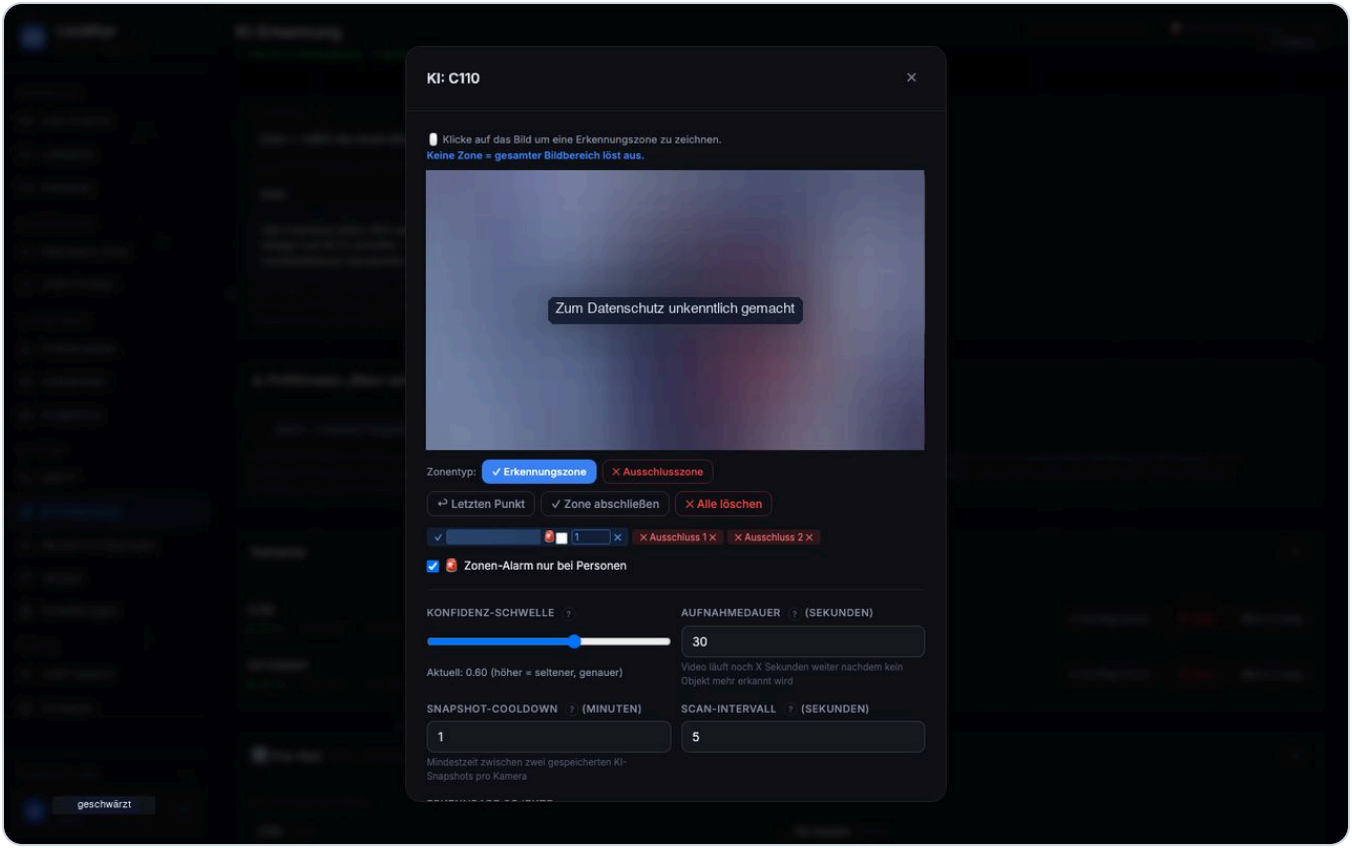
3. **Konfidenz-Schwelle** (0,6 ist ein guter Start): höher = seltener, aber sicherer.

4. **Aufnahmedauer** = wie lange nach der letzten Erkennung weiter aufgezeichnet wird; **Scan-Intervall** = wie oft die KI hinsieht; **Snapshot-Cooldown** = Mindestabstand zwischen gespeicherten Bildern.

5. Klassen und Alarm-Klassen wählen (z. B. nur Personen alarmieren). Zur Wahl stehen Person, Fahrrad, Auto, Motorrad, Bus, LKW und *Tiere* – ein Schalter für alle Tierarten; im Livebild heißt jedes erkannte Tier „Tier“. Im Datenschutzprofil zählt jede Kamera Personen und nur Personen: *Person* steht fest auf an, die anderen Kacheln sind abgeblendet und mit einem Schloss markiert. Deine Auswahl bleibt gespeichert und gilt wieder, sobald das Profil aus ist. Soll eine Kamera gar nichts zählen, schalte ihre KI aus.

6. **Bildquelle für die Erkennung** wählen: *Automatisch* ist der schnellste Weg — bei UniFi Protect ist das der Schnappschuss der Anlage, und der liefert nur 640×360, unabhängig davon was die Kamera kann. Für Personen und Fahrzeuge genügt das.

7. Speichern und die KI für die Kamera einschalten. Der Status zeigt „läuft“.



Zonen-Editor: grüne Erkennungszone, rote Ausschlusszonen, darunter Konfidenz, Aufnahmedauer, Cooldown und Intervall (Kamerabild unkenntlich).

Gut zu wissen: Leere Zahlenfelder werden ignoriert (der alte Wert bleibt) — die Erkennung läuft in jedem Fall weiter.

Unbewegte Objekte melden nur einmal

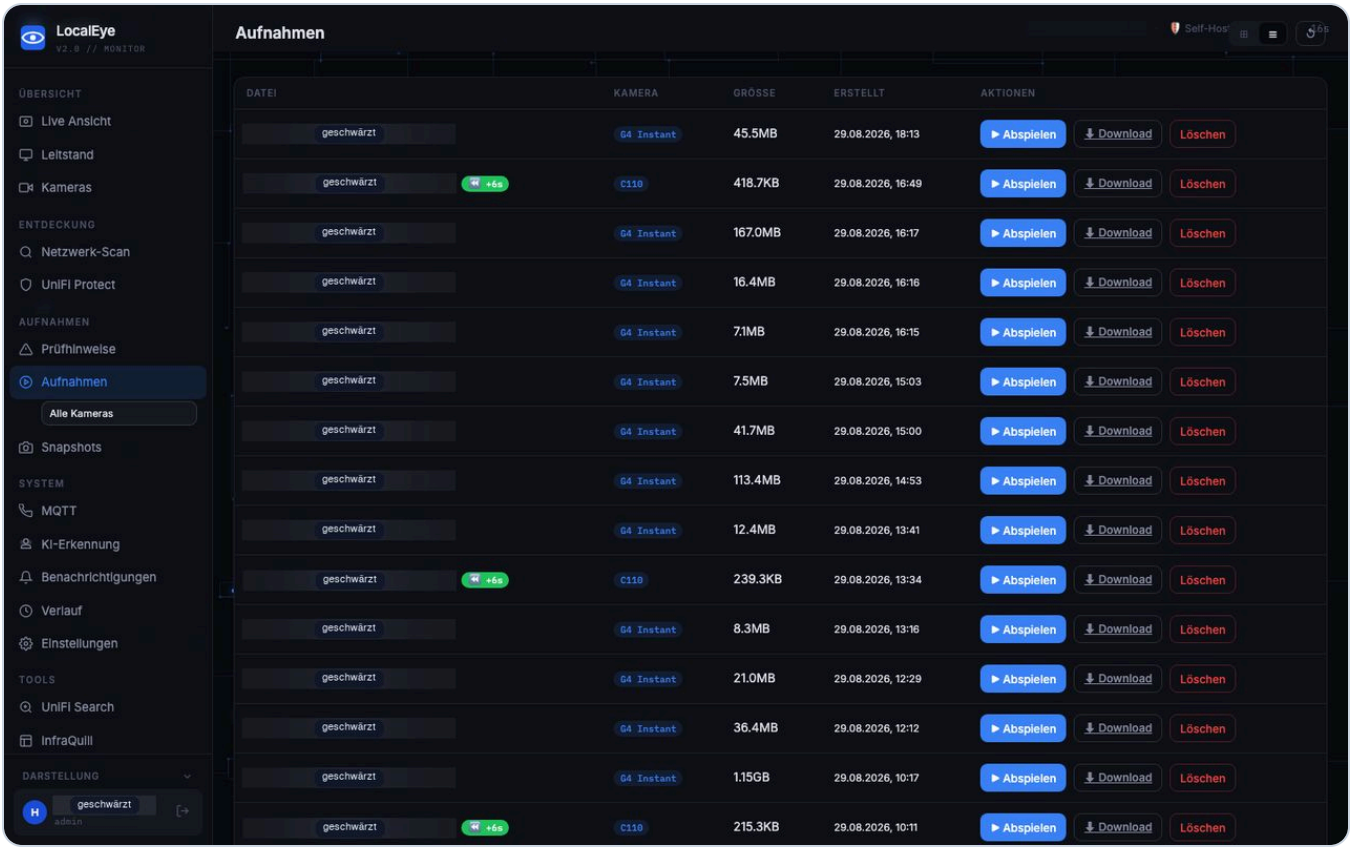
Kein Erkennungsmodell ist fehlerfrei: Ein Gartenstuhl, ein abgestelltes Fahrrad oder ein Haufen Gerümpel kann als Person durchgehen. Früher hätte so ein Fehltreffer in *jedem* Takt — also alle paar Sekunden — eine neue Meldung, einen neuen Schnappschuss und eine Verlängerung der laufenden Aufnahme ausgelöst.

Die Anlage vergleicht deshalb jede Erkennung mit dem, was im Takt davor an derselben Stelle stand. Was sich nicht bewegt hat, gilt als bekannt und meldet **nicht erneut**. Sobald sich das Objekt bewegt, verschwindet oder etwas Neues ins Bild kommt, wird sofort wieder gemeldet.

Gut zu wissen: Die Live-Ansicht zeigt trotzdem alles. Ein Dauer-Fehltreffer bleibt im Kamerabild und in der Statusanzeige sichtbar — sonst könnten Sie ihn gar nicht finden und abstellen. Nur die Meldung bleibt aus. Den Fehltreffer selbst beseitigen Sie mit einer *Ausschlusszone* über die betroffene Ecke oder einer höheren Konfidenz-Schwelle.

Gut zu wissen: Die Ausschlusszone muss den **Mittelpunkt** des roten Rahmens abdecken, nicht bloß den Gegenstand — die Anlage entscheidet anhand des Boxmittelpunkts, ob eine Erkennung in einer Zone liegt.

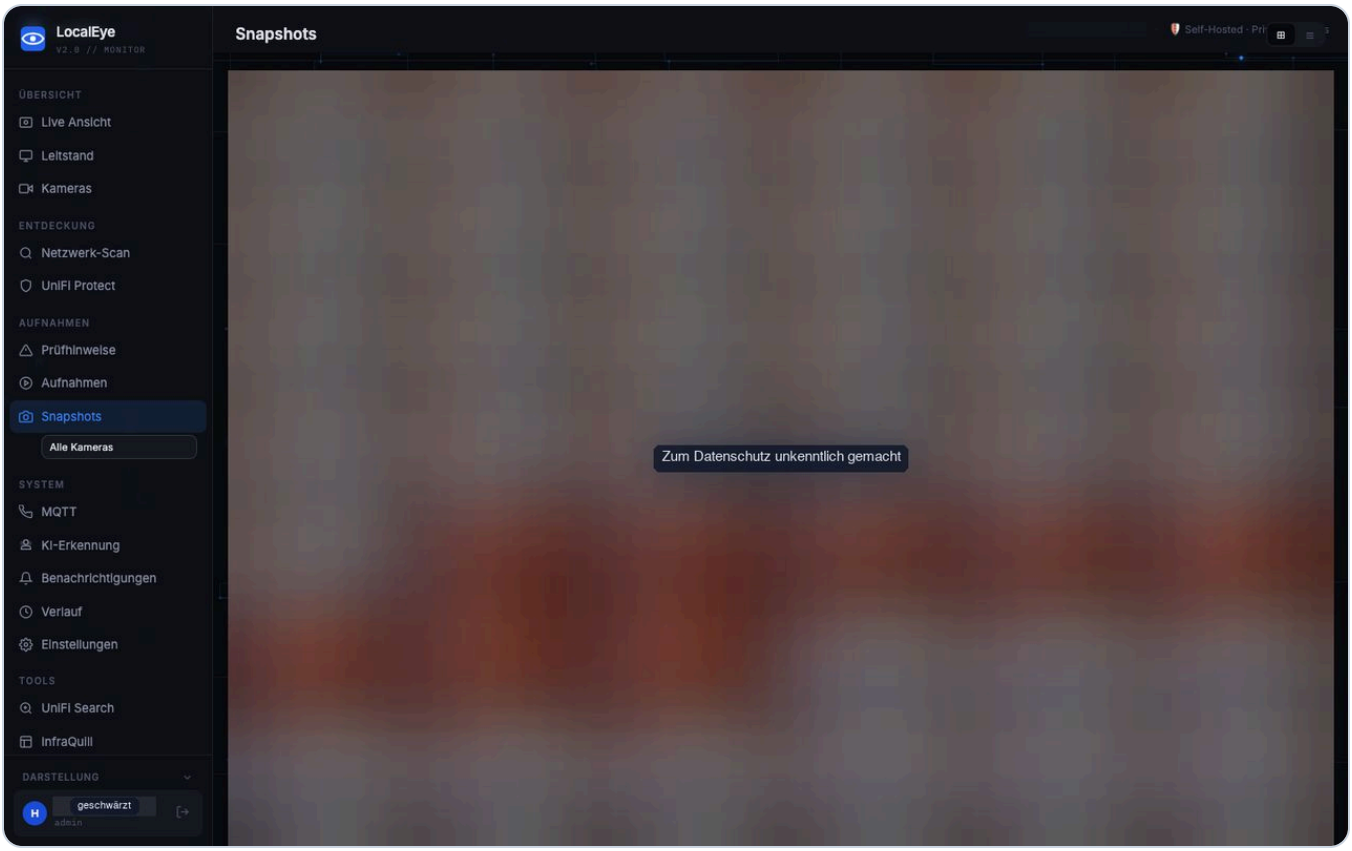
3.5 Aufnahmen, Snapshots & Prüfhinweise



Aufnahmen: Liste oder Galerie, Filter nach Kamera und Art (KI / Bewegung / Manuell), Abspielen direkt im Browser.

- **Aufnahmen** entstehen automatisch bei KI-Erkennung, per Bewegungs-Auslöser oder manuell (Aufnahme-Knopf in der Live-Ansicht). Ablageort und Aufbewahrungsdauer stellst du unter *Einstellungen → System* ein.
- Solange weiter erkannt wird, läuft die Aufnahme weiter — allerdings nur bei *neuen* Erkennungen. Ein unbewegter Fehltreffer verlängert sie nicht mehr endlos; sie endet nach der eingestellten Aufnahmedauer.
- **Liste:** Die Seite Aufnahmen zeigt je Abschnitt die neuesten 200 Aufnahmen. **Ältere Aufnahmen laden** unter der Liste holt den nächsten Abschnitt; ein gewählter Kamerafilter gilt dabei weiter. Sind ältere Abschnitte nachgeladen, bleibt die Liste nach Legal Hold, Freigeben oder Löschen stehen. Lehnt die Anlage dann ein Löschen ab (zum Beispiel, weil die Aufnahme noch läuft oder inzwischen unter Legal Hold steht), nennt eine Meldung den Grund.
- **Vorlauf (Pre-Roll):** Das Zeichen *+Xs* an einer Aufnahme zeigt die tatsächlich vorangestellte Vorlauf-Länge in Sekunden, gemessen an den angefügten Segmenten. Direkt nach dem Start einer Kamera kann sie kürzer sein als eingestellt.
- **Snapshots** sind Einzelbilder mit eingezeichneten Erkennungen — ideal für Benachrichtigungen.
- **Prüfhinweise** sammelt Ereignisse, die ein Mensch bestätigen sollte (z. B. unsichere Erkennungen), samt Beweisvideo-Export. Den Zeitraum wählst du per Kalender *von – bis* oder mit den Knöpfen *2 / 7 / 14 / 30 Tage* (Vorgabe: zwei Tage, höchstens 30). Jede

Bewertung ✔ *bestätigt* / ✗ *Fehlalarm* ist zugleich eine Rückmeldung für den KI-Lernlauf — samt der Begründung, die du dabei eingibst. Ein Klick auf das Vorschaubild öffnet die Großansicht mit Zoom: Klick, Mausrad oder Pinch vergrößern um den Zeiger, Ziehen verschiebt, ⟲ setzt zurück, Hintergrund oder Esc schließt.

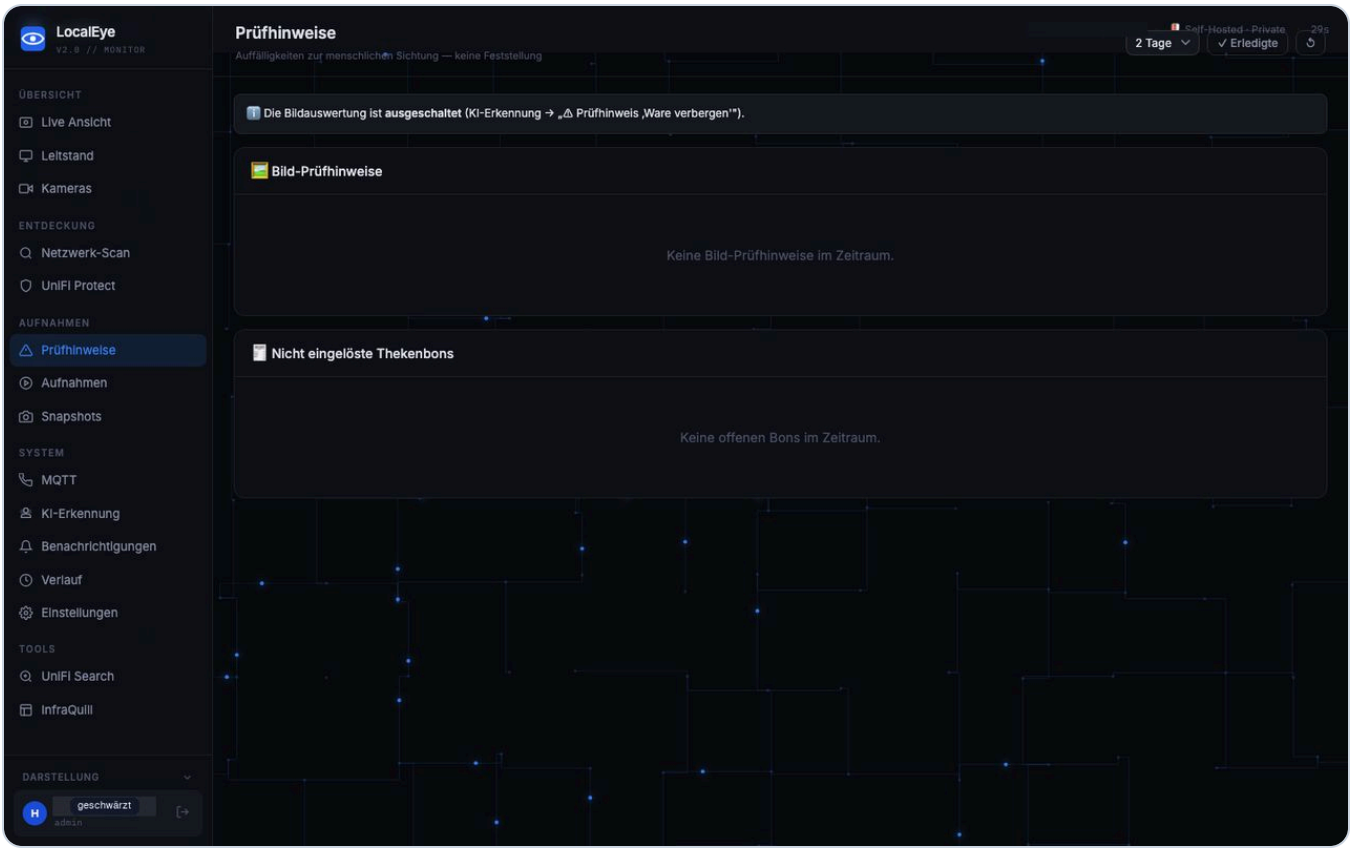


Snapshots-Galerie (Bilder zum Datenschutz unkenntlich).

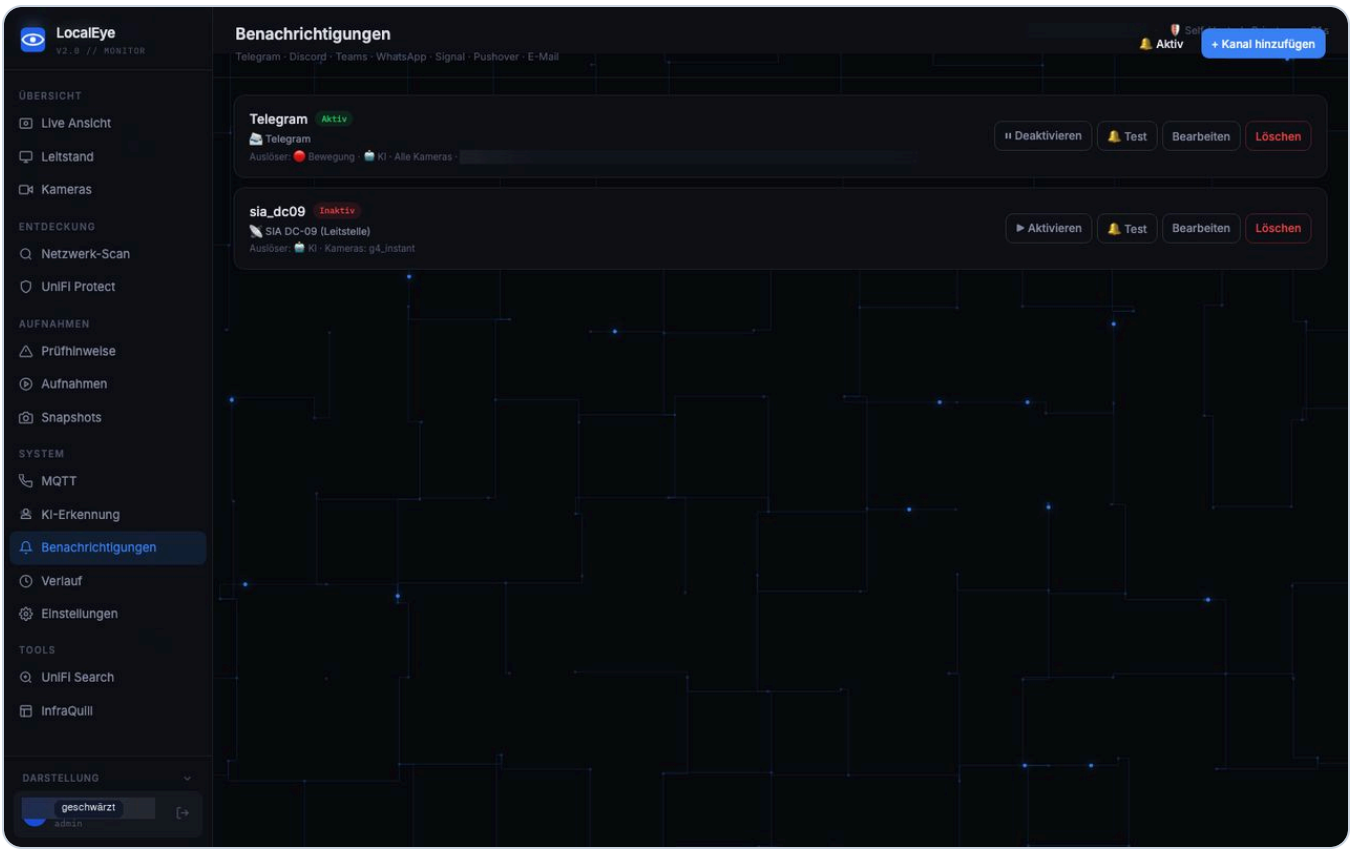
Griff-Wache: Das wichtigste Merkmal von LocalEye zur Diebstahlerkennung. Sie gehört zum Release und ist ab Werk aus, auch für neue Kameras, ein Update schaltet sie nie ein. Eingeschaltet wird sie mit eigenem Schalter unter **Funktionen** → **Griff-Wache**, nur mit Bestätigung und Auswahl der Kameras. Sie braucht die Prüfhinweise, im Datenschutzprofil zusätzlich die Freigaben „Griff-Wache“ und „Prüfhinweise und Verhalten“. Nur mit Betriebsvereinbarung und Datenschutz-Folgenabschätzung betreiben. Die Wache achtet auf mögliches Einstecken, nicht auf das Anfassen von Ware. Ein Hinweis braucht zwei Schritte: Die Hand geht mit Ware an Hosentasche, Bund, Handtasche oder unter die Jacke, gleich wo die Ware genommen wurde (ohne Griff ins Regal kurz davor zählt nur Ware, die die Bild-KI mit Namen nennt, etwa Flasche oder Buch; im Sitzen prüft die Wache nie), und die Bild-KI sieht danach eine sichtbare Hand ohne Ware oder beschreibt das Einstecken. Verlässt die Person gleich danach das Bild, bevor die Hand wieder zu sehen ist, lautet der Hinweis „Ware genommen, Verbleib nicht zu sehen“ (nur allein im Bild, höchstens sechs je Kamera und Stunde). Korb und Einkaufswagen zählen nie. **Getragene Ware:** Bestätigt die Bild-KI bei einem geprüften Griff Ware, die danach noch in der Hand ist, merkt sich die Wache das je

Kamera und Person bis zu 60 Sekunden, nur im Arbeitsspeicher. Geht die Hand dann unter die Kleidung und ist die Ware danach in keiner Hand mehr klar zu sehen, entsteht der Hinweis auch, wenn die Bild-KI die Ware beim Griff selbst nicht sieht. Für Hinweise gilt das Gedächtnis nur, solange die Person allein im Bild ist. Eigene Sachen wie Handy, Geld oder Kassenbon und vage beschriebene Gegenstände werden nie gemerkt. Ein Treffer erscheint hier als „möglicherweise eingesteckt, ungesichert“ mit vier Bildern des Ablaufs: -1 s, Griff, +2 s, +5 s. Ob eingesteckt wurde, entscheidet der Mensch: Der Hinweis ist keine Feststellung und kein Echtzeit-Alarm, und wie gut er anschlägt, hängt vom Kamerawinkel ab. Wie oft die Wache im Markt richtig oder falsch anschlägt, ist noch nicht gemessen. *Fehlalarm* löscht die Bilder des Hinweises sofort. Aufs Handy kommt er nur bei Kanälen, die „Griff-Hinweis“ ausdrücklich wählen, nie an die Leitstelle. **Kameras, die nur den Oberkörper zeigen:** Am zuverlässigsten arbeitet die Wache, wenn die Kamera die Person bis zur Hüfte zeigt; die Person muss mindestens ein Fünftel der Bildhöhe einnehmen, das Gesicht braucht die Wache nie. Verdecken Theke, Regal, Möbel oder Bildrand den Unterkörper, zeigt die Kamera nicht, wohin eine Hand geht, die nach unten oder hinter einen Vordergrund verschwindet, es kann auch der Wagen, ein Korb oder die Ablage unter der Theke sein. An solchen Kameras bleibt die Wache meist still, bis unter **KI-Erkennung → Konfigurieren** der Schalter *Ablage nicht sichtbar* eingeschaltet ist (ab Werk aus). Verschwindet die Hand mit Ware deutlich sichtbar an der Brust, etwa unter die Jacke, prüft sie das an jeder Kamera wie sonst. Mit dem Schalter meldet sie in zwei Fällen, den zweiten nur nach einem Griff ins Regal, den ersten auch ohne (dann nur mit benannter Ware): Die Hand verschwindet mit Ware und ist danach ohne Ware wieder zu sehen („Ware genommen, Hand leer zurück, Ablage nicht sichtbar“), oder sie ist nicht mehr zu sehen, bis die Person das Bild verlässt („Ware genommen, Verbleib nicht zu sehen“; nur solange die Person allein im Bild ist, höchstens sechsmal je Kamera und Stunde, rund eine halbe Minute nach dem Verlassen des Bildes). Den zweiten Hinweis löst auch aus, wer Ware mit gesenkter Hand zur Kasse trägt. Dort kommen mehr Hinweise, die meisten unbegründet: den Schalter nur an einzelnen Kameras einschalten; Kassen, Theken mit Personal und Sitzplätze nimmt dort am besten eine Griff-Ausnahme aus. An Kameras mit dem Schalter rechnet für eine Person allein im Bild zusätzlich ein größeres Skelett-Modell, das kostet dort mehr Rechenzeit. Ohne den Schalter bleiben solche Griffe still; soweit die Wache sie erkennt, zählt sie sie, die Zahl steht in der Kachel Griff-Wache. **Zuletzt:** Die Kachel Griff-Wache und der KI-Dialog jeder Kamera zeigen mit Uhrzeit, was die Wache dort zuletzt entschieden hat und warum, zum Beispiel „kein Hinweis: Die Bild-KI nannte vor dem Verschwinden keine klare Ware in der Hand“. So ist nach einem Test zu sehen, woran es lag. Den Griff an Hosentasche und Bund erkennt die Wache nur, wenn die Hüfte im Bild ist. Je Kamera lässt sich das unter **KI-Erkennung → Konfigurieren** abschalten, Kasse und Personalbereiche nimmt die Zonenart *Griff-Ausnahme* aus. Der Knopf mit dem Strichmännchen in der Kamerakachel blendet das Skelett im Live-Bild ein. Gilt eine Person als „trägt Ware“, sehen Admins um diese Hand

zusätzlich den Rahmen „Gegenstand“, ohne Namen des Gegenstands; andere Rollen sehen ihn nicht. Der Rahmen folgt der Spur der Person: Tritt binnen rund 15 Sekunden eine andere Person an dieselbe Stelle, kann er bis zum Ablauf der 60 Sekunden an deren Hand erscheinen. Ein Hinweis entsteht daraus nicht.



3.6 Benachrichtigungen & Verlauf



Kanäle mit Auslöser, Kameras und Zeitfenstern; jeder Kanal hat einen Test-Knopf.

1. **Benachrichtigungen → + Kanal hinzufügen**.
2. Kanal wählen: Telegram, Discord, Teams, WhatsApp, Signal, Pushover, **ntfy** (Push ohne Cloud, eigener Server möglich), E-Mail, Webhook oder SIA (Leitstelle).
3. Zugangsdaten eintragen (z. B. Telegram-Bot-Token und Chat-ID) und **Test** senden.
4. Auslöser festlegen: Bewegung, KI, Zonen-Alarm, Prüfhinweis, Griff-Hinweis, Kasse, **Störung/Entstörung** (Anlagenwächter), **Kamera-Ereignis** (Sabotage/Videoverlust) — welche Kameras, welche Klassen. **Nur wenn Anlage scharf** koppelt den Kanal an die Scharfschaltung.
5. **Cooldown** = Mindestabstand zwischen zwei Meldungen dieses Kanals in Minuten; gilt immer (0 = aus). Optional **Zeitfenster** setzen (z. B. Mo–Fr 06:45–09:00) — außerhalb bleibt der Kanal stumm; mit Zeitplan zählt der Cooldown je Fenster getrennt.

Unter **Verlauf** siehst du jede gesendete Meldung mit Zeitpunkt, Kanal und Ergebnis.

Stummschalten

Stummschalten (Schalter **Meldungen aktiv** / **Stummgeschaltet**) betrifft alle Meldekanäle außer der Leitstelle (Push, E-Mail, Teams, Webhook, Messenger). Die Leitstelle bleibt aufgeschaltet; wer im Objekt arbeitet, schaltet unscharf. Unscharf hält einen Kanal nur zurück, wenn dort die Option **Nur wenn Anlage scharf** eingeschaltet ist. Fehlt sie an einem aktiven Leitstellen-Kanal, steht am Schalter: *Der Leitstellen-Kanal meldet auch unscharf: Option „Nur wenn Anlage scharf“ am Kanal einschalten.*

War die Anlage beim Update stummgeschaltet, gilt diese Stummschaltung bis zum nächsten Aufheben auch für die Leitstelle. Hat die Anlage einen aktiven Leitstellen-Kanal, steht solange im Störungsbanner der Seitenleiste *Meldungen stummgeschaltet, auch die Leitstelle*. Danach lässt jede neue Stummschaltung die Leitstelle aus.

Leitstelle zuerst: Bei jedem Ereignis kommt die Leitstelle zuerst dran. Quittiert sie nicht innerhalb von 3 Sekunden, gehen die übrigen Kanäle trotzdem raus; ihr Ergebnis steht danach im Meldungsverlauf.

ntfy — der eingebaute Push-Server

Ein eigener **ntfy**-Server (Apache-2.0, kostenlos, keine Cloud) läuft als Container im Stack. Die Handy-App (Android/iOS) ist gratis; Meldungen und Bilder bleiben auf deiner Anlage. Für iPhones geht nur ein Wecksignal über ntfy.sh.

1. **Benachrichtigungen** → **+ Kanal** → **Typ ntfy**: Server-URL, Zugangs-Token, ein zufälliges Topic und Priorität 4 sind bereits vorbelegt — nur noch **Speichern**. Unter jedem Feld steht, was es bewirkt (Priorität 1 = leise … 5 = dringend, Tags = Emoji-Kürzel).

2. Im Dialog erscheint der Kasten **Handy einrichten**: Seit dem 17.09.2026 verlangt der Server für Lesen und Senden eine Anmeldung (`deny-all`) — die Handy-App meldet sich mit dem Benutzer `localeye` und dem Passwort aus dem Kasten an; das Topic ist zusätzlich ein langes Zufallsgeheimnis. **Android:** QR mit der Kamera scannen → die ntfy-App öffnet sich mit Server und Topic → *Abonnieren*. **iPhone:** in der ntfy-App *+* → Topic eintragen → *Anderen Server verwenden* → Adresse eintragen (einmalig als Standard-Server merken); der QR öffnet dort nur die Web-Ansicht, weil die iOS-App keine Links verarbeitet.

3. **Test** senden — die Meldung mit Bild muss auf dem Handy erscheinen.

4. **Push unterwegs:** Der Server muss aus dem Internet erreichbar sein. Im Reverse-Proxy (z. B. Nginx Proxy Manager) eine eigene Subdomain wie `ntfy.deine-domain.de` auf `NAS-IP:8991` legen (WebSocket-Unterstützung einschalten), dann unter **Einstellungen → Verbindungen → ntfy-Push: Adresse für die Handy-App** `https://ntfy.deine-domain.de` eintragen und speichern — der Push-Server übernimmt die Adresse automatisch (ca. 10 s). Ohne Subdomain kommen Meldungen nur im WLAN/VPN an.

Gut zu wissen: Senden geht nur mit Token (legt manage.sh beim ersten Start an, `localeye/config/.ntfy_credentials.json`); Lesen ist seit dem 17.09.2026 nur noch angemeldet möglich (vorher konnte jeder im Netz ein Topic mitlesen, dessen Namen er kannte). Bestehende Handy-Abos ohne Anmeldung empfangen nach dem Update nichts mehr — in der ntfy-App den Benutzer `localeye` nachtragen. **iPhone/iPad:** Apple erlaubt keine eigenen Push-Verbindungen, deshalb schickt der Server nur ein Wecksignal über ntfy.sh (Häkchen *iOS-Wecksignal* in den Verbindungen); den Inhalt holt die App vom eigenen Server. Nur Android? Häkchen aus — kein Kontakt nach außen.

3.7 Anlagenwächter & Scharfschaltung

Der **Anlagenwächter** überwacht die Anlage selbst: Er prüft jede Minute, ob jede Kamera Bilder liefert (bei laufender KI über das Bildalter, sonst über die Netz-Erreichbarkeit der Kamera), ob im Aufnahmeordner genug Platz ist, ob das KI-Modell geladen ist und die Auswertung läuft, ob der Stream-Dienst antwortet, ob die Bild-KI antwortet (bei eingeschalteten Prüfhinweisen; ohne sie dann, wenn die Bildbeschreibung an ist und der Bildserver auf diesem Gerät laufen soll), und ob Meldungen zur Leitstelle wiederholt scheitern. Erst nach drei Treffern in Folge wird eine **Störung** gemeldet, nach zwei guten Durchläufen die **Entstörung** — so entsteht kein Flattern.

1. Links in der Seitenleiste erscheint bei aktiven Störungen ein gelbes Banner mit Klartext (z. B. „Kamera ‚Hof' liefert kein Bild"). Ankündigungen, etwa einer Löschfrist, stehen darunter neutral als *Hinweis*, sie sind keine Störung.

2 Damit Störungen auch verschickt werden: beim gewünschten **Benachrichtigungskanal** den Auslöser **Störung / Entstörung** anhaken. Für die Leitstelle gehen Störcodes nach SIA DC-05 raus (z. B. 380 Kamera, 393 Speicher, 350 Verbindung), die Entstörung als Rückstellung. Zustände der Griff-Wache (pausiert, zu wenige Bilder, Fehlerhäufung, Skelett-Modell fehlt) gehen nicht an die Leitstelle. Sie stehen im Anlagenwächter und gehen an die übrigen Kanäle.

3 Schwellen unter **Einstellungen** → **Verbindungen**: *Bild älter als (s)* und *Speicher-Warnung unter (GB)*; leer = Standard (120 s, 5 GB).

4 Für Monitoring-Systeme (Prometheus, Grafana, Zabbix, Checkmk) liefert `/metrics` Kennzahlen je Kamera und Anlage — Aufruf mit dem API-Token (`?token=…` oder Kopfzeile `X-API-Token`).

Zwei Meldungen aus der Störungsliste:

- *Ereignisquelle der Kamera „Name“ abgelehnt: Zugangsdaten prüfen* – die Kamera hat die Anmeldung der Ereignisquelle abgelehnt. LocalEye versucht es nicht erneut, damit die Kamera nicht sperrt. Nach dem Ändern der Zugangsdaten verbindet es neu und die Störung endet.
- *Bild-KI antwortet nicht — Ereignisse bekommen keine Beschreibung* – der Bildserver soll auf diesem Gerät laufen, antwortet aber nicht. Ereignisse werden ohne Beschreibung gespeichert, die Recherche in KnowSora findet sie dann nicht über den Bildinhalt. Diese Störung geht nicht an die Leitstelle.

Scharf / Unscharf

Der Knopf **Scharf** oben in der Seitenleiste schaltet die Anlage um; das ⚙ daneben öffnet den **Zeitplan**: in den Zeitfenstern (Wochentage, von–bis, Übernacht möglich) ist die Anlage automatisch scharf, außerhalb unscharf. Ein manueller Klick gilt bis zum nächsten geplanten Wechsel; unter dem Knopf steht, wann die Automatik als Nächstes schaltet — bei einem Wechsel am selben Tag als *„heute 22:00“*, am Folgetag als *„morgen 06:45“*, sonst mit Wochentag. Haben Sie von Hand umgeschaltet und weicht der Zustand vom Zeitplan ab, sagt die Zeile das ausdrücklich: *„Von Hand unscharf — Automatik: unscharf ab heute 22:00“*. Meldekanäle, bei denen **Nur wenn Anlage scharf** angehakt ist, schweigen im unscharfen Zustand — Störungen und Verbund-Meldungen gehen immer. Umschaltungen stehen im Verlauf und im Protokoll.

Gut zu wissen: Extern schalten, z. B. von der Einbruchmeldeanlage, einem Shelly oder Loxone: `POST /api/arming` mit `{{"scharf": true}}` und dem API-Token. Den Zustand liefert `GET /api/arming`. Im Datenschutzprofil schaltet der API-Token nur aus dem Adressbereich der Verwaltung (`VERWALTUNG_LAN_CIDRS`) oder direkt am Gerät; ohne gültigen Bereich nur am Gerät.

3.8 Zwei-Faktor-Anmeldung (2FA)

Ist LocalEye aus dem Internet erreichbar (Reverse-Proxy, VPN-Portal), sollte jeder Admin zusätzlich zum Passwort einen Einmalcode aus einer Authenticator-App verlangen (TOTP — funktioniert mit Google/Microsoft Authenticator, Aegis, 2FAS, Bitwarden u. a.).

1. **Mein Konto (Klick auf den eigenen Namen unten links) → Zwei-Faktor-Anmeldung → 2FA einrichten**: QR-Code mit der App scannen (oder das Geheimnis abtippen).
2. Den aktuellen Code aus der App eingeben und **Aktivieren** klicken.
3. Die acht **Wiederherstellungs-Codes** sicher aufbewahren — sie werden nur dieses eine Mal angezeigt und gelten je einmal (für den Fall, dass das Handy verloren geht).
4. Beim nächsten Login fragt LocalEye nach dem Passwort zusätzlich den Code ab.

Gut zu wissen: Handy verloren und keine Codes? Ein anderer Admin setzt die 2FA unter **Einstellungen → Benutzer** mit dem Knopf 🔐 **2FA** zurück. Mit **2FA für Administratoren verpflichtend** (Einstellungen → Verbindungen) können Admins ohne 2FA nichts anderes tun, bis sie sie eingerichtet haben. Nach fünf Fehlversuchen wird die Anmeldung von dieser Adresse für 30 Sekunden (steigend bis 15 Minuten) gesperrt.

Gut zu wissen: Ein Code aus der Authenticator-App gilt nur einmal, auch zwischen LocalEye und „Anlage verwalten“. Wer sich gerade in LocalEye angemeldet hat, nimmt für „Anlage verwalten“ den nächsten Code (höchstens 30 Sekunden warten). Sonst erscheint: „Dieser Code wurde schon verwendet. Bitte den nächsten Code abwarten.“

3.9 Protokoll & Beweissicherung

Das **Protokoll** (Seitenleiste → System) ist das Audit-Journal der Anlage: Anmeldungen (auch fehlgeschlagene), angesehene und gelöschte Aufnahmen, Exporte, Einstellungs- und Benutzeränderungen, Scharf/Unscharf, Legal Hold, Alarmverifikationen, Aufrufe von Prüfseite und Prüfbild (mehr als 30 je Minute und Ereignis als Zahl), Live-Ansichten (im Datenschutzprofil auch das Einzelbild) und jeder Aufbewahrungslauf, der etwas löscht, mit Zählern. Im Datenschutzprofil stehen dort außerdem gebündelt Vorschaubilder von Aufnahmen, abgerufene Snapshots, verpixelte Vorschaubilder an Meldekanäle, abgelehnte Abrufe per Token („Abruf per Token abgelehnt“) und abgewiesene Verwaltung („Verwaltung abgewiesen“). Die Zahl weiterer Aufrufe eines Bündels schreibt ein Abschluss-Eintrag nach. Die Zugriffe der Leitstelle stehen mit und ohne Profil im Protokoll. Die Einträge sind **hash-verkettet**: **Kette prüfen** meldet geänderte oder herausgelöste Einzeleinträge. Abgeschnittene älteste oder jüngste Einträge, ein Zurückspielen per Restore und eine vollständig neu berechnete Kette erkennt erst der Abgleich mit einer externen Kopie. Das beantwortet die Frage „Wer hat wann welche Aufnahme angesehen?“ — wichtig für Betriebsvereinbarung und Datenschutzbeauftragten.

1. Filter nach Aktion, Nutzer und Zeitraum; **Excel-Export** für Prüfer.

2. **Aufbewahrung** in Tagen unter Einstellungen → Verbindungen (Standard 365; gelöscht werden nur ganze Monate).

Legal Hold und Beweispaket

Muss eine Aufnahme trotz kurzer Aufbewahrungsfrist erhalten bleiben (Anzeige, Versicherungsfall), setzt ein Admin in der Aufnahmen-Liste 🔒 **Hold** mit Begründung. Gehaltene Aufnahmen kann weder der Löschen-Knopf noch der automatische Aufbewahrungslauf entfernen; 🔓 **Freigeben** hebt das wieder auf. Der automatische Lauf lässt außerdem Bilder, Berichte und Indexzeilen derselben Kamera aus dem Zeitraum der Aufnahme stehen (mit zwei Minuten Rand), auch von Hand ausgelöste Schnappschüsse.

📦 **Beweispaket** lädt eine ZIP-Datei mit dem Video, den zugehörigen Dateien, einem Manifest mit SHA-256-Prüfsummen, einem Prüfwert (HMAC) dieser Anlage und dem Skript `PRUEFEN.py`, das die Prüfsummen ohne LocalEye kontrolliert (`python3 PRUEFEN.py paket.zip`). Ob ein Paket wirklich von dieser Anlage stammt und unverändert ist, prüft **Paket prüfen** oben auf der Aufnahmen-Seite.

Gut zu wissen: Die Signatur bestätigt Herkunft und Unversehrtheit gegenüber dieser Anlage — sie ersetzt keine gerichtliche Beweiswürdigung, macht die Kette aber nachvollziehbar.

3.10 Alarmverifikation für die Leitstelle

Eine Leitstelle (NSL) oder ein Bereitschaftsdienst kann einen Alarm mit Bild prüfen, bevor jemand losfährt. Dazu trägst du unter **Einstellungen** → **Verbindungen** die **Öffentliche Basis-URL** ein, unter der LocalEye von außen erreichbar ist (z. B. `https://localeye.example.de`).

1. Ab dann enthält jede Meldung einen signierten Link `/verify/…` (24 Stunden gültig, ohne Anmeldung): bei Zonen-Alarmen und Prüfhinweisen im Meldetext, im Webhook als `verify_url`, bei ntfy als Klick-Ziel.

2. Die Seite zeigt das eingefrorene Ereignisbild, das Livebild der Kamera (alle 3 s) und die Meldung. Mit **Alarm bestätigt**, **Fehlalarm** oder **Unklar** (optional Name/Notiz) wird die Bewertung gespeichert.

3. Die Bewertung erscheint im **Verlauf** (Typ „Verifikation“) und im Protokoll.

Datenschutzprofil: Im Datenschutzprofil gibt es statt des Verifikationslinks den Prüflink und den Prüfzugang der Leitstelle (unten). Meldungen tragen keinen Verifikationslink, und die Seite /verify/ antwortet nur noch mit einem kurzen Hinweis, auch für schon verschickte Links. Eine Bewertung durch die Leitstelle ist dort nicht möglich.

Gut zu wissen: Der Webhook-Kanal liefert Snapshot-Links jetzt über die Basis-URL mit Tickets, die 24 Stunden gelten, statt `localhost` und Dauer-Token; ohne Basis-URL bleibt der Link lokal.

Prüfzugang der Leitstelle

Die Alarm-Leitstelle kann einen gemeldeten Alarm über ihr VPN direkt auf der Anlage prüfen. Dafür bekommt sie ein eigenes Konto mit der Rolle **Leitstelle**. Es sieht keine Kamera, keine Zusatz-App und keine KI.

1. In der `.env` unter `LEITSTELLE_CIDRS` das Netz eintragen, aus dem die Leitstelle zugreift, z. B. `192.168.50.0/24` (mehrere mit Komma). Leer, ungültig oder ein Netz über alle Adressen sperrt das Konto ganz; über den Fernzugriff ist es immer gesperrt, mit und ohne Datenschutzprofil.

2. **Einstellungen** → **Benutzer**: Konto mit der Rolle *Leitstelle* anlegen und an einen oder mehrere SIA-Kanäle mit Kennung binden. Das geht nur im lokalen Netz oder per VPN.

3. Die Leitstelle öffnet `https://<IP>:8943/leitstelle/`, richtet beim ersten Anmelden den zweiten Faktor ein und bewahrt die Wiederherstellungscodes auf. Eine Sitzung gilt höchstens 12 Stunden.

Sie sieht die Zonen-Alarme (E130), die die Anlage per SIA an ihre Kanäle gemeldet hat und die dort quittiert wurden, mit Kamera, Leitstellen-Nummer und Zone. Dazu das Prüfbild: verpixelt wie der Prüflink, darüber ein Wasserzeichen mit Konto, Uhrzeit und Objektname. Kein Livebild, keine Aufnahme, kein Clip, kein Download. Abrufbar ist ein Alarm ab der Quittung so lange, wie `LEITSTELLE_FENSTER_MIN` sagt (Vorgabe 60 Minuten, höchstens 240). Die Ansicht zeigt nur Alarme von Kameras dieser Anlage, nicht die einer Kamera-Erweiterung. Ändert ein Admin Typ, Ziel, Ports oder Kennungen eines gebundenen SIA-Kanals oder löscht ihn, löst die Anlage die Bindung und meldet „Bindung der Leitstelle gelöst, bitte neu zuordnen“. Abweisungen und Zugriffe stehen gebündelt im Protokoll.

3.11 Kamera-Ereignisse (ISAPI / ONVIF)

Viele Kameras erkennen selbst Bewegung, Linienüberschreitung, Bereichseintritt, Personen, Sabotage/Abdeckung oder Videoverlust. LocalEye kann diese **kameraseitigen Ereignisse** abhören und als Auslöser nutzen — bei Hikvision, HiLook und ANNKE über den ISAPI-Ereignisstrom, bei Dahua, Reolink, Axis, Uniview u. a. über ONVIF-Events.

1. **Kameras** → **Bearbeiten** → **Kamera-Ereignisse**: Typ wählen (*Hikvision ISAPI* oder *ONVIF Events*), Host, Benutzer und Passwort eintragen – bei ONVIF der **ONVIF-Benutzer** der Kamera (bei Dahua und Hikvision ein eigenes Konto, siehe 3.2); bei einem NVR den Typ Hikvision ISAPI wählen und den Kanal immer angeben.

2. Bei Hikvision am jeweiligen Ereignis „Surveillance Center benachrichtigen“ aktivieren; bei ONVIF-Kameras die Ereignisse in der Kamera einschalten.

3. **Sabotage, Abdeckung, Videoverlust** und Kamera-Speicherfehler meldet LocalEye sofort — Kanal-Häkchen **Kamera-Ereignis**; zur Leitstelle gehen Sabotage (137) und Videoverlust (380) mit der Meldelinie der Kamera.

4. **KI nur bei Kamera-Ereignis**: Die Objekterkennung wartet auf ein Ereignis der Kamera und wertet dann sofort aus (statt im Dauertakt) — das spart Rechenleistung bei vielen Kameras; ein Kontrolldurchlauf alle 60 Sekunden bleibt. Ohne dieses Häkchen wird das Kamera-Ereignis selbst als Meldung „Kamera-Ereignis“ verschickt.

5. Der Status (verbunden, Fehler, Zahl der Ereignisse) steht im Kamera-Dialog.

Gut zu wissen: Gesichts-Ereignistypen der Kamera werden grundsätzlich verworfen (keine biometrische Verarbeitung). Nur steigende Flanken zählen, mit Cooldown je Kamera und Ereignisart — Kameras, die Dauerzustände melden, lösen nicht ständig aus.

Gut zu wissen: Datenschutzprofil: Dort verwirft LocalEye Kamera-Ereignisse zu Fahrzeug, Tier, Kennzeichen und Parken, ebenso Smart-Erkennungen ohne Person. Sie werden weder gemeldet noch ins Ereignisprotokoll geschrieben, nur gezählt (Selbsttest). Bewegung, Linie und Bereich gehen auch dann durch, wenn ein Fahrzeug sie auslöst. Sabotage und Videoverlust gehen immer durch.

3.12 Inbetriebnahme: Gehtest & Aufschaltprotokoll

Zwei Dokumente, die bei jeder Inbetriebnahme entstehen müssen und bisher von Hand geschrieben wurden. Beide erzeugt LocalEye jetzt als PDF — die Anlage kennt die Daten ohnehin.

Gehtest — die Abnahme der Erfassungsbereiche. Sie starten ein Zeitfenster, laufen durch jeden Bereich, und die Anlage schreibt mit, welche Zone tatsächlich ausgelöst hat.

1. **Benachrichtigungen** → 🚶 **Gehtest**: Prüfername und Dauer eintragen (Standard 10 Minuten), dann *Gehtest starten*.

2. Durch jeden Erfassungsbereich laufen. Die Ansicht aktualisiert sich alle fünf Sekunden und zeigt je Zone, ob sie schon ausgelöst hat.

3. Am Ende *Beenden* und *Protokoll (PDF)* — mit Uhrzeiten je Zone, erkannten Objekten, Ergebnis und Unterschriftszeilen.

Gut zu wissen: Zonen, die nie ausgelöst haben, stehen im Protokoll ausdrücklich als „nicht ausgelöst“ — das ist der eigentliche Zweck. Erkennt eine Kamera etwas außerhalb aller Zonen, wird auch das ausgewiesen.

Aufschaltprotokoll — das Objektblatt für die Notruf- und Serviceleitstelle. Ohne dieses Dokument schaltet keine Leitstelle ein Objekt auf.

1. **Benachrichtigungen** → 📄 **Aufschaltprotokoll** (derselbe Knopf steht auch unter **Einstellungen** → **Verbindungen** → **„Leitstelle (NSL) — Pflichtdaten“**): Objektdaten, Leitstelle, Interventionsanweisung, Schlüsseldepot und die Kontaktkette eintragen (die Leitstelle ruft von oben nach unten an).

2. Die Meldelinien-Tabelle darunter füllt LocalEye selbst — sie zeigt je Kamera und Zone genau die Adresse, die im Alarmfall gesendet wird.

3. **Diebstahlverdacht an die Leitstelle** ist Verhandlungssache: Standardmäßig bleiben Prüfhinweise der Bild-KI intern (Push, E-Mail, Telegram — mit Bild und Verifikationslink). Erst wenn der SIA-Meldeweg auf *Diebstahlverdacht als Voralarm (E138)* steht, geht er als Contact-ID 138 „Near Alarm“ auf der Meldelinie der Kamera an die Leitstelle — und erscheint dann als eigene Zeile im Aufschaltprotokoll. E138 ist eine Verdachtsmeldung, keine Einbruchmeldung (E130); die Leitstelle muss das in ihrer Interventionsanweisung kennen. Im Datenschutzprofil ist der Voralarm aus und nicht freigebbar, Griff-Hinweise gehen nie an die Leitstelle.

4. **Meldecodes:** Zonen-Alarme gehen als Einbruch E130 hinaus, ein neuer Leitstellen-Kanal ist darauf vorbelegt. E602 ist allein die Testmeldung (zyklisch oder per Kanaltest); Bewegung und KI-Erkennung gehen nie an die Leitstelle. Im Datenschutzprofil ist die zyklische Testmeldung Pflicht, ohne eingestellten Takt alle 6 Stunden.

5 **Prüflink:** Zu jedem Zonen-Alarm entsteht ein Link auf eine Seite ohne Anmeldung mit dem Ereignisbild, ohne Livebild – ab Werk 60 Minuten gültig, höchstens 24 Stunden. Verpixelt sind Köpfe und Gesichter erkannter Personen und der untere Teil erkannter Fahrzeuge, wo in der Regel das Kennzeichen sitzt, im Datenschutzprofil Motorräder ganz. Verpixelt werden alle Personen und Fahrzeuge, die die KI im Bild erkennt, auch außerhalb der Zonen und unabhängig von der Objektauswahl der Kamera. Personen zählen dafür schon ab geringer Sicherheit, auch unter der Konfidenz-Schwelle der Kamera. Im Datenschutzprofil verpixelt LocalEye bei gebückten, vorgebeugt knienden und liegenden Personen einen größeren Bereich um den Kopf. Den Kopf schätzt LocalEye aus der Lage der erkannten Person: Bei ungewöhnlicher Haltung (Kopf unten, frontal gebückt, Kind im Einkaufswagen) kann er sichtbar bleiben, Gesichter sucht LocalEye zusätzlich im ganzen Bild. Wen Erkennung und Gesichtssuche in einem Bild nicht finden, der bleibt dort unverpixelt. Liegen zum Ereignis keine Erkennungsrahmen vor, zeigt die Seite statt eines Bildes einen kurzen Hinweis. Aufrufe von Seite und Bild stehen im Protokoll, mehr als 30 je Minute und Ereignis als Zahl. Der Webhook trägt ihn als Feld `pruef_link`, im Datenschutzprofil nur mit *Prüflink mitsenden*; E-Mail, Push und Teams als Zeile, wenn am Kanal *Prüflink mitsenden* an ist (ab Werk aus). Die Gültigkeit stellst du unter **Einstellungen** → **Verbindungen** ein.

6 **Clip im Prüflink (ab Werk aus):** Mit der Einstellung unter **Einstellungen** → **Verbindungen** steht unter dem Prüfbild ein kurzer Clip der 8 Sekunden nach dem Alarm. LocalEye baut ihn nur, wenn ein Kanal den Prüflink wirklich versendet hat, und holt die Bilder nur über den eigenen Restream der Kamera. Verpixelt wird wie im Prüfbild. Es gibt keinen Clip, wenn das bei einem einzigen Bild scheitert, wenn die Erkennung die Person oder das Fahrzeug des Alarms in keinem Bild wiederfindet oder wenn der Rechner ausgelastet ist. Eine Person, die Erkennung und Gesichtssuche in einem Bild nicht finden, bleibt dort unverpixelt. Aufrufe stehen im Protokoll. Der Clip wird mit seinem Ereignisbild gelöscht und beim Ausschalten sofort, er geht in keine Sicherung. Im Datenschutzprofil gibt es ihn nur mit der befristeten Freigabe „Clip im Prüflink“ unter **Einstellungen** → **Funktionen** (Begründung, Grundlage, Enddatum). Ohne gültige Freigabe ist er aus, und die fertigen Clips werden gelöscht; ein gehaltenes Ereignis behält seinen.

7 *PDF erzeugen*. Fehlen Pflichtangaben, ist das Dokument sichtbar als *Entwurf* gekennzeichnet.

Achtung: Die Meldeadressen im Blatt stammen aus derselben Funktion, die den Alarm verschickt — das Protokoll kann nicht von der Wirklichkeit abweichen. Fehlt im Meldeweg die SIA-Kennung oder ist er abgeschaltet, weist das Blatt darauf hin: Ohne Kennung kann die Leitstelle eine eingehende Meldung keinem Objekt zuordnen.

3.13 Texterkennung in Bildern und Aufnahmen

LocalEye kann Text in Kamerabildern lesen — **standardmäßig ausgeschaltet**, in zwei Varianten.

1. **Einstellungen → Verbindungen → Text auf Ereignisbildern erkennen**: Ab dann wird zu jeder Erkennung auch der Bildtext gelesen und ist über die Ereignis-Recherche auffindbar. Kostet je Ereignis rund zwei Sekunden Rechenzeit.

2. **Aufnahmen →** 🔤 **Text**: durchsucht eine einzelne Aufnahme auf Anforderung; das Ergebnis nennt die Fundstellen mit Zeitmarke.

Im **Datenschutzprofil** braucht auch die Suche in einer Aufnahme die Freigabe *Texterkennung*. Ohne sie trägt der Knopf den Zusatz *gesperrt*, und die Anlage lehnt die Suche ab. Die Freigabe schaltet auch die Texterkennung auf Ereignisbildern ein; diese lässt sich danach unter **Einstellungen → Verbindungen** wieder ausschalten.

Gut zu wissen: Ehrlich zur Erwartung: Zuverlässig gelesen wird **große, nahe Schrift** — Schilder, Aufdrucke, eingeblendete Zeitangaben. Ein Regaletikett aus vier Metern ist auf einem Übersichtsbild zu klein für die Texterkennung. Jedes Ergebnis nennt deshalb selbst, wie belastbar es ist („3 von 12 Bildern trugen lesbaren Text“), und wenn die Texterkennung im Backend fehlt, sagt die Anlage das, statt stumm nichts zu finden.

3.14 Kassendaten einspielen (DSFinV-K)

Für den Bon-Abgleich gibt es zwei Wege. Der eine ist die **Direktanbindung**: Die Kasse schickt jeden Vorgang sofort — nur damit läuft der Abgleich in Echtzeit. Der andere ist der **DSFinV-K-Export**, den jede deutsche Kasse liefern können muss.

1. **Einstellungen → Kassen-Abgleich → DSFinV-K-Export einspielen**: Exportdatei (ZIP) wählen, optional eine Filialkennung eintragen.

2. *Prüfen* — ein Trockenlauf. Er zeigt, wieviele Verkaufsvorgänge enthalten sind, welcher Zeitraum abgedeckt ist und was übersprungen wird. Gespeichert wird dabei nichts.

3. *Übernehmen* — die Vorgänge werden in den Abgleich eingespielt, jeweils am Belegtag. Ein zweiter Import derselben Datei erzeugt keine Dubletten.

Gut zu wissen: Thekenbons und Kassenvorgänge werden getrennt gezählt: Als doppelt gilt nur ein Beleg derselben Art mit derselben Nummer.

Gut zu wissen: Der Export entsteht beim Kassenabschluss und liegt damit frühestens am Tagesende vor — er **ersetzt die Direktanbindung nicht**, sondern ergänzt sie: als herstellerunabhängiger Weg, für den niemand etwas an der Kasse einbauen muss, und als einziger Weg, zurückliegende Tage nachzutragen.

Gut zu wissen: Beschäftigten- und Kundendaten aus dem Export (Bedienername, Personalnummer, Kundenanschrift, Freitextnotizen) werden beim Einlesen **verworfen** und erreichen die Anlage nie. Übernommen werden nur Warenkorbdaten. Stornierte Belege und Übungsbuchungen zählen nicht als Verkauf.

3.15 Schnappschuss-Token

Schnappschuss-Token für eigene Werkzeuge (`/api/snap/<kamera>/live.jpg?token=…` oder Kopfzeile `X-Snapshot-Token`): LocalEye erzeugt ihn seit dem 17.09.2026 beim ersten Start zufällig und legt ihn unter `localeye/config/.snapshot_token` ab (nur für den Betreiber lesbar) — dort auslesen. Der frühere feste Vorgabewert gilt nicht mehr; wer ihn in einer bestehenden Verdrahtung weiter braucht, trägt `SNAPSHOT_TOKEN` bewusst in die `.env` ein.

Im **Datenschutzprofil** öffnet kein Token ein Kamerabild oder Video, auch nicht der API-Token oder der Leitstand-Token: Kamerabilder gibt es dort nur nach Anmeldung. Ein Abruf per Token bekommt die Antwort 410 (der Weg zum Videodienst 403) und steht gebündelt als „Abruf per

Token abgelehnt“ im Protokoll. LocalEye legt im Profil keine Schnappschuss-Token-Datei an und löscht eine vorhandene (Eintrag „Token entfernt“). Auch ein `SNAPSHOT_TOKEN` in der `.env` öffnet im Profil kein Bild, der Selbsttest weist darauf hin.

> **Gut zu wissen: Kennzeichenerkennung entfernt:** LocalEye liest seit Version 2026.11.16 keine Kennzeichen mehr, auf jeder Anlage, mit und ohne Datenschutzprofil. Die Seite „Kennzeichen“, ihr Schalter unter *Funktionen* und die Option im KI-Dialog der Kamera sind bewusst entfallen. Der erste Start nach dem Update löscht die gespeicherte Liste, den Verlauf, die Bildausschnitte und die Kennzeichen-Meldungen im Meldungsverlauf; im Datenschutzprofil steht das mit Zahlen im Protokoll. Fahrzeuge macht LocalEye im Prüfbild im unteren Teil unkenntlich.

3.16 Leistung: Vorrang der Kamera-KI

Der Menüpunkt **Leistung** (unter Einstellungen → System, nur für Admins) regelt, wie sich die Suite-Dienste die Rechenleistung der Maschine teilen. Im empfohlenen **Auto-Modus** gilt: **Die Kamera-KI — Objekterkennung, Bildbeschreibung und Streams — hat immer Vorrang.** KnowSora-Hintergrundarbeit (Einbettung, Kontextstufe, Chat-KI) nutzt nur freie Reserven und wird bei Knappheit automatisch zurückgedrängt.

GPU (NVIDIA) — automatisch erkannt

Steckt eine NVIDIA-Karte im Server, richtet die Suite sie beim Update selbst ein: `./manage.sh update` erkennt Host-Treiber und Container-Toolkit, baut LocalEye aus dem GPU-Image (CUDA 12.4) und stellt die lokale KI (Chat, Bildbeschreibung) auf die GPU-Ausgabe von llama.cpp um. Die Karte **GPU — Kamera-KI** im Menü Leistung zeigt Name, Treiber, VRAM, Auslastung und Temperatur. **Auto** (Vorgabe) bevorzugt immer die GPU: RF-DETR, sonst YOLOX Medium in voller Genauigkeit auf CUDA. **Nur CPU** erzwingt die CPU — sofort für die Kamera-KI; die lokale KI folgt mit `./manage.sh gpu cpu` bzw. `gpu auto`.

Vollautomatisch, ohne Nacharbeit: `./manage.sh update` erkennt die Karte schon am PCI-Bus, auch ohne Treiber. Fehlt der Treiber, installiert das Update ihn selbst (Debian/Ubuntu), richtet den Nachlauf-Dienst ein, baut und startet zuerst alle Dienste mit der neuen Version und fragt erst am Ende des Updates nach dem Neustart, beim Online-Update nach dem Selbsttest (ohne Antwort startet es nach 30 Sekunden neu; `GPU_AUTO_REBOOT=1` in der `.env` startet ohne Rückfrage). Nach dem Neustart baut der Dienst `allinone-gpu-nachlauf` das GPU-Profil von selbst. Fehlt nur das Container-Toolkit, wird es im selben Lauf installiert. Zum Schluss prüft ein Selbsttest im Container, ob die CUDA-Kette wirklich rechnet

— scheitert er, bleibt die Anlage im CPU-Profil, der Grund steht in der `.env` unter `GPU_SELBSTTEST_FEHLER`, und die Überwachung läuft weiter.

1. Server mit NVIDIA-Karte aufstellen, Suite wie gewohnt installieren: `./manage.sh update`. Alles Weitere macht das Update.

2. `./manage.sh gpu status` zeigt jederzeit Karte, Treiber, Toolkit, Betriebsart und den letzten Selbsttest; `./manage.sh gpu auto|cpu` schaltet das Profil um (mit Selbsttest).

3. `GPU_AUTOMATIK=0` in der `.env` schaltet die automatische Treiber-/Toolkit-Installation ab, falls ein Kunde das nicht möchte.

Gut zu wissen: Für das RF-DETR-Modell im GPU-Image muss beim Bau `RFDETR_URL` in der `.env` auf einen Spiegel der ONNX-Datei zeigen; ohne die Datei nimmt Auto YOLOX Medium auf der GPU. Treiber ab R525 genügen (CUDA 12.4 mit Minor-Version-Kompatibilität). Der Selbsttest ersetzt die Abnahme vor Ort: Er öffnet eine echte CUDA-Sitzung auf einem Modell des Images und rechnet einen Durchlauf.

KI-Lernlauf (Auto-Training) und KnowSora-Ablage

Die Karte **KI-Lernlauf** im Menü Leistung lernt nach Zeitplan (Wochentage, Uhrzeit) oder auf Knopfdruck aus euren Rückmeldungen — aus ✔ *bestätigt* / ✗ *Fehlalarm* in der Prüfliste und aus dem Verifikationslink der Leitstelle. Prüfhinweise, die als Fehlalarm bewertet wurden, werden zu Normal-Mustern der Kamera (die Begründung des Bewerters bleibt am Muster gespeichert); die Konfidenz-Schwelle je Kamera wandert in kleinen Schritten zwischen 0,3 und 0,8 (hoch bei vielen Fehlalarmen, runter, wenn echte Treffer knapp über der Schwelle lagen). **Schwellen zurücksetzen** stellt den Stand vor dem letzten Lauf wieder her. Kein KI-Modell wird dabei verändert. Im Datenschutzprofil verliert der gespeicherte Bericht des letzten Laufs die Tabelle der einzelnen Rückmeldungen, sobald der Lauf älter als die Frist der Entscheidungen ist (höchstens 72 Stunden); die übrigen Abschnitte bleiben. Eine Kopie in KnowSora (Freigabe KnowSora-Ablage) fällt nicht unter diese Frist. Die Karte sehen nur Admins. Im Datenschutzprofil läuft der Lernlauf nur mit der befristeten Freigabe „Lernlauf“ unter **Einstellungen** → **Funktionen** (Begründung, Grundlage, Enddatum): Einschalten, Start von Hand und jeder Lauf nach Zeitplan brauchen sie, Ausschalten und *Schwellen zurücksetzen* gehen immer. Läuft die Freigabe ab, schaltet die Anlage den Lernlauf aus. Der Datensatz fürs Modell-Feintuning bleibt im Profil aus.

Gelerntes

Unter **Überwachung → Gelerntes** (nur Admins) steht auf einen Blick, was die KI gelernt hat: Kennzahlen, eine Zeitleiste der *Neuerungen* – neue Normal-Muster aus Kalibrierung oder Lernlauf, angepasste Schwellen, übernommene Pool-Muster, Versand an den Hersteller – mit Datum und Grund, dazu je Kamera Schwelle und Normal-Muster und die Muster aus dem Trainingspool samt Messung.

KI-Training mit dem Hersteller

Freiwillig und standardmäßig aus, einzuschalten unter **Anlage verwalten → KI-Training**. *Prüfhinweise senden* überträgt einmal pro Woche die bewerteten Prüfhinweise an den Hersteller, damit die KI Diebstahl und Vandalismus auch bei anderen Kunden sicherer erkennt. Vorher verpixelt die Anlage automatisch und ohne Gewähr Köpfe und Gesichter erkannter Personen, den unteren Teil erkannter Fahrzeuge, wo in der Regel das Kennzeichen sitzt, und die Einblendungen der Kamera. Den Kopf schätzt sie aus der Lage der Person, bei ungewöhnlicher Haltung kann er sichtbar bleiben. Wen Erkennung und Gesichtssuche nicht finden, der bleibt unverpixelt. Die Fälle bleiben deiner Lizenz zugeordnet; Kameranamen, Zonen, Notizen und Uhrzeiten gehen nicht mit. Vor dem Versand ersetzt LocalEye die Namen von Kameras, Zonen und Standort (Objektname, Ort und Straße aus dem Aufschaltprotokoll, Verbund-Standorte, eigener Firmenname) in Beschreibung und Satz durch „Kamera“, „Zone“ und „Standort“: den ganzen Namen, auch mit Bindestrich geschrieben, und einzelne Wörter mehrteiliger Namen ab vier Zeichen. Das gilt auch für frühere Namen umbenannter oder gelöschter Kameras und Zonen. Gewöhnliche Wörter wie Kasse, Eingang, Flasche oder links bleiben einzeln stehen, damit die Beschreibung verständlich bleibt. Einen umgestellten Namen erkennt der Export deshalb nur an Wörtern, die nicht zu den gewöhnlichen zählen. Abkürzungen, Zusammensetzungen und gebeugte Formen eines Namens erkennt er nicht. *Gelerntes übernehmen* holt einmal pro Woche nur Texte aus dem Pool – nie Bilder anderer Kunden. Ein Fehlalarm-Satz wird erst zum Pool-Muster, wenn er in mindestens zwei Anlagen als Fehlalarm bewertet wurde und keinen einzigen echten Fall – aus dem Pool oder eigene bestätigte Prüfhinweise – unterdrücken würde; schneidet ein neuer Stand schlechter ab, bleibt der alte. Unter *Gelerntes* lässt sich beides auch sofort anstoßen.

1. Mit **Ereignisse & Berichte in KnowSora ablegen** landen die Ereignisse je Kamera und Tag als kurze Tabellen (Zeit, Objekte, Zone, Beschreibung, Verdacht, Rückmeldung samt Begründung) in einem KnowSora-Kern — durchsuchbar und per Wiki-Verdichten zu Wochenberichten zusammenfassbar. Ändert sich eine Bewertung oder Begründung nachträglich, ersetzt der nächste Lauf die Tagesdatei im Kern; ältere Tage kommen nach, sobald jemand darin bewertet hat. Der Laufbericht (mit allen neuen Rückmeldungen und Begründungen) kommt dazu. Es werden nur Texte übertragen, keine Bilder.

2. Der Kern gehört dem Admin, der die Einstellungen zuerst speichert (andere Admins lassen ihn sich in KnowSora freigeben); die Anmeldung läuft über dasselbe SSO wie die App-Kachel.

3. Ein Trainingsdatensatz für echtes Modell-Feintuning entsteht nur, wenn das Image einen Trainingsstack enthält, und wird nach dem Lauf gelöscht — der Arbeitsordner wird in jedem Fall geräumt, es sammelt sich nichts an.

Auto oder Manuell

1. **Auto** (empfohlen): Prioritäten und CPU-Deckel werden aus der Kernzahl der Maschine berechnet — nichts weiter zu tun.

2. **Manuell**: je Dienst die Priorität (Minimal bis Hoch) und einen CPU-Deckel in 0,5-Kern-Schritten wählen, dann **Anwenden**.

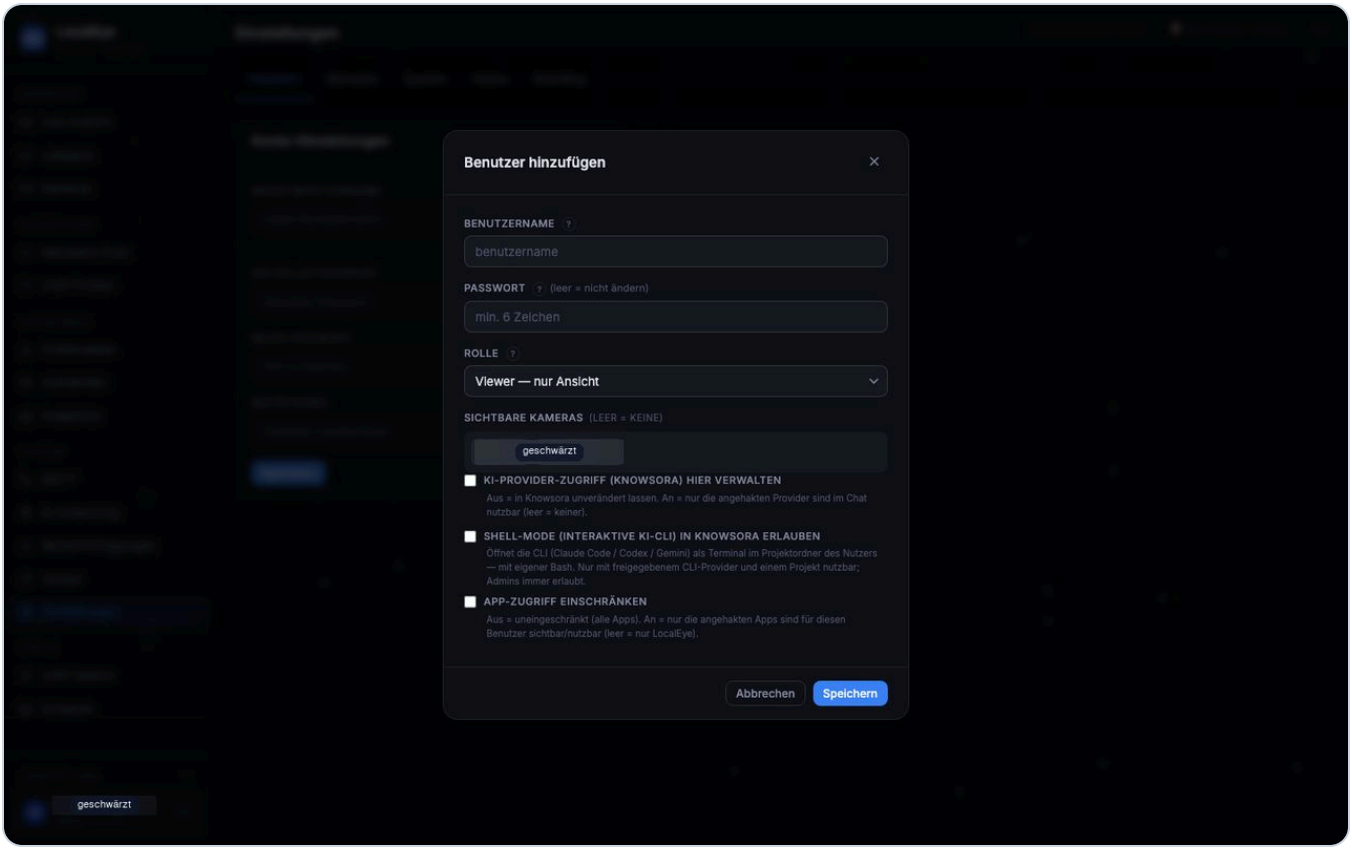
3. Änderungen wirken sofort, ohne Neustart — und werden alle 10 Minuten automatisch erneut durchgesetzt, damit sie auch Updates und Container-Neustarts überleben.

4. **Ausnahme lokale KI:** Die beiden KI-Server (Chat/Kontextstufe und Bildbeschreibung) legen ihre Rechenthreads beim Start passend zum CPU-Deckel fest. Ändern Sie deren Deckel, startet die Anlage den betroffenen Server einmal kurz neu (10–20 Sekunden), damit die Threads folgen — mehr Threads als Deckel würden nur gedrosselt.

Gut zu wissen: Die Priorität wirkt nur, wenn die CPU wirklich knapp ist — im Leerlauf wird nichts verschenkt. RAM-Werte sind bewusst nur Anzeige: ein zu kleines Speicherlimit würde die geladenen KI-Modelle abstürzen lassen.

3.17 Benutzer & Rechte

Die Benutzerverwaltung unter **Einstellungen** → **Benutzer** ist die zentrale Stelle für *alle* Apps: Was du hier vergibst, gilt beim nächsten Login auch in KnowSora, SearchTool und InfraQuill.



Benutzer anlegen: Rolle, sichtbare Kameras, KI-Provider-Zugriff, Shell-Mode und App-Zugriff.

1. **Einstellungen → Benutzer → Benutzer hinzufügen**.
2. Benutzername, Passwort (mind. 6 Zeichen) und **Rolle** wählen.

3. Bei *Viewer*: die **sichtbaren Kameras** anhaken. Die Rolle *Leitstelle* ist nur für den Prüfzugang der Alarm-Leitstelle (Kapitel „Alarmverifikation für die Leitstelle“): Das Konto bekommt keine Kamera, keine App und keine KI, sondern eine Bindung an SIA-Kanäle. Anlegen und Binden gehen nur im lokalen Netz oder per VPN. Wechselt ein Konto in diese Rolle oder aus ihr heraus, enden seine Sitzungen; das eigene Konto kann die Rolle weder bekommen noch abgeben.

4. **KI-Provider-Zugriff (KnowSora) hier verwalten** anhaken und die erlaubten KI-Anbieter wählen — nur diese sieht der Nutzer im Chat.

5. **Shell-Mode erlauben**: gibt dem Nutzer in KnowSora die interaktive KI-Kommandozeile im eigenen Projektordner (mit Bash). Nur an vertrauenswürdige Nutzer vergeben. Wichtig: Die CLI arbeitet mit der hinterlegten CLI-Anmeldung des Betreibers (Abo-Login) — Shell-Freigabe bedeutet Zugriff auf diese Anmeldung. Seit dem 17.09.2026 ist sie außerdem Voraussetzung für die Werkzeuge `python_exec` und `project_run_shell` in Skills, Agenten und Workflows.

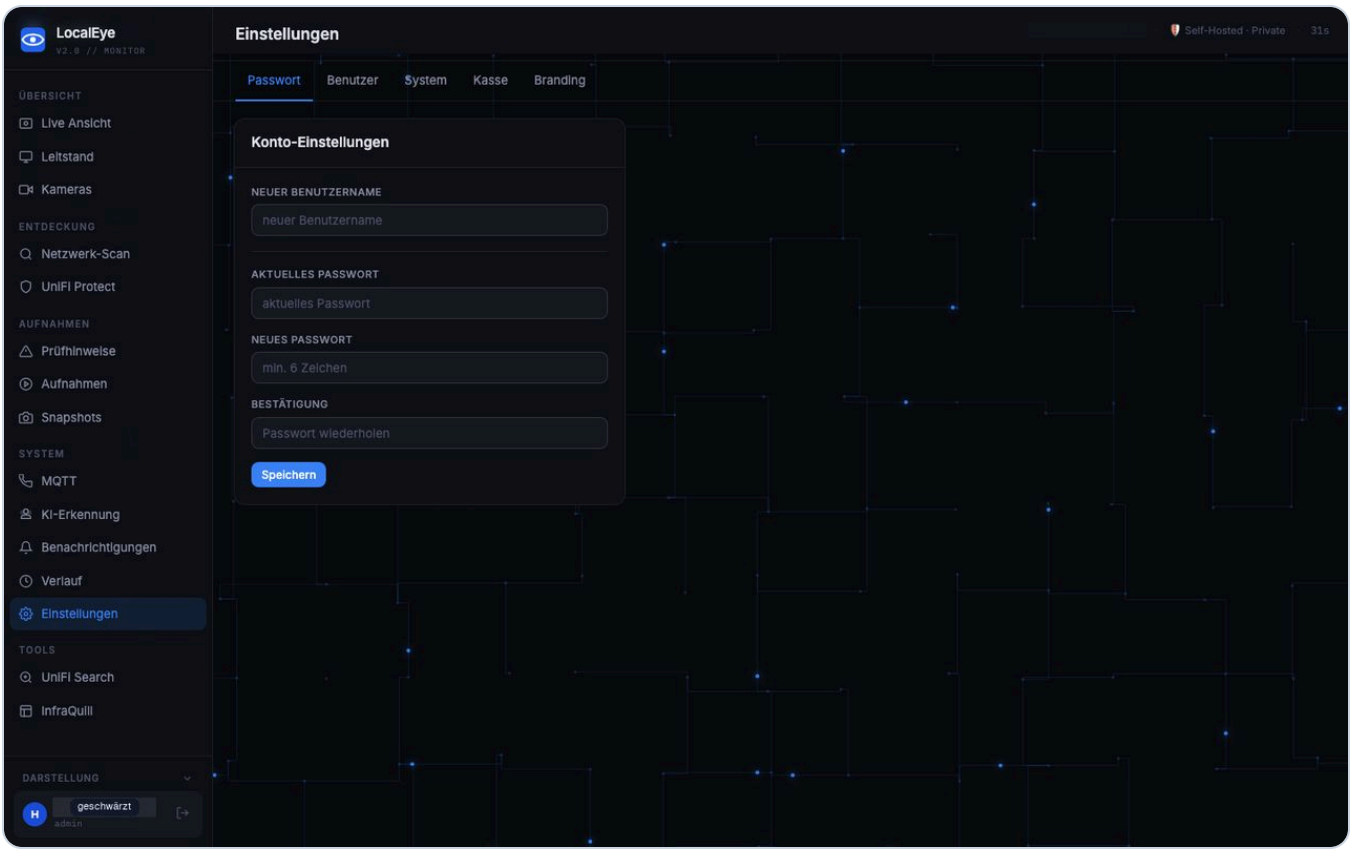
6. **App-Zugriff einschränken**: welche der vier Apps der Nutzer überhaupt sieht.

7. Speichern. Änderungen greifen beim nächsten Login des Nutzers; ein gelöschter Nutzer verliert den Zugriff sofort.

Gut zu wissen: Jedes Konto hat eine feste interne Kennung. Sie bleibt beim Umbenennen erhalten. Wird ein Konto gelöscht und später unter demselben Namen neu angelegt, ist es für die Zusatz-Apps eine neue Person: In KnowSora beginnt es mit einem leeren Konto.

Gut zu wissen: Anmelde-Link der Zusatz-Apps: Ein Anmelde-Link einer Zusatz-App gilt nur in dem Browser, in dem LocalEye ihn erzeugt hat. Wer ihn weiterschickt, meldet damit niemanden an. Wer den Anmelde-Token per Skript holt (`/api/auth/sso-token`), gibt beim Einlösen das Cookie `le_sso_bindung` aus der Antwort mit oder trägt `SSO_BINDUNG=aus` in die `.env` ein (auch `off` , `0` , `false` oder `nein` , ohne Kommentar in derselben Zeile).

3.18 Einstellungen & Branding



Einstellungen: Passwort, Benutzer, System (Speicherorte, Aufbewahrung, Ports), Kasse, Branding.

Der Datei-Browser zur Ordnerauswahl (Speicherorte, Importe) steht seit dem 17.09.2026 nur noch Administratoren offen; Viewer sehen keine Ordnerstruktur des Servers.

- **Passwort** — eigenes Passwort und Benutzername ändern.
- **System** — Aufnahme- und Snapshot-Ordner (auch außerhalb des Suite-Ordners; ein neuer Ordner wird nach dem Speichern unter „Anlage verwalten“ per Aktualisieren angelegt und eingehängt, nach einem Plattentausch „Datenträger getauscht“ bestätigen), Aufbewahrungsdauer (leer = Standard, 0 = unbegrenzt nach Bestätigung), HTTPS mit Fingerabdruck und eigenem Zertifikat, Sprache, Zeitzone.
- **Verbindungen** — außerdem: Öffentliche Basis-URL (Alarmverifikation), Wächter-Schwellen, Protokoll-Aufbewahrung, 2FA-Pflicht für Admins.
- **Branding** (Whitelabel-Edition) — Name, Logo, Farben und Texte der Oberfläche; gilt für alle Apps. Entfernst du hier das Branding-Theme, schaltet KnowSora ohne Neuladen auf das dunkle Standard-Theme zurück.

3.19 Verhaltenserkennung: Herumlungern / Verweildauer

Zusätzlich zum reinen Zonen-Zutritt erkennt LocalEye **Verhalten**: wenn eine Person ungewöhnlich lange an einer Stelle **verweilt** (Herumlungern / „Loitering“) – der klassische Fall

vor Regalen, an Geldautomaten oder im Kassenbereich. Die Prüfung läuft **vollständig lokal und ohne GPU**, zusätzlich zur Bildbeschreibung; es verlässt kein Bild das Gerät.

1. Einmalig einschalten: **Einstellungen** → **Funktionen** → **„Verhaltens-Erkennung“** aktivieren. Ohne diesen Schalter erscheinen die Felder nicht und es wird kein Verhalten gemeldet.

2. Links **KI-Erkennung** öffnen und bei der gewünschten Kamera **Konfigurieren** – der Abschnitt *Verhalten* ist jetzt sichtbar.

3. **Herumlungern/Verweilen** für diese Kamera anhaken.

4. **Verweildauer** (Sekunden) setzen: so lange muss eine Person an derselben Stelle bleiben, bis gemeldet wird. Standard **90 s**; zum Ausprobieren ruhig kurz (z. B. 20).

5. Optional **Nur in Alarmzonen**: dann zählt Verweilen nur in den gezeichneten Erkennungszonen, nicht im ganzen Bild (z. B. nur direkt vor dem Regal).

6. Speichern. Verweilt künftig jemand länger als eingestellt, erscheint das Ereignis *Verweildauer* im Meldungsverlauf und löst – wie jede Erkennung – Aufnahme, Snapshot und die eingestellten Meldungen aus.

Damit das Verhalten auch **verschickt** wird: beim gewünschten Benachrichtigungskanal den Auslöser **Verhaltens-Erkennung (Loitering)** anhaken (Kapitel „Benachrichtigungen & Verlauf“). An eine **Leitstelle** geht Verweilen als **Voralarm** (SIA Contact-ID **E138**, „Near Alarm“) auf der Meldelinie der Kamera – dieselbe Verdachts-Meldung wie der Diebstahlverdacht, ausdrücklich keine Einbruchmeldung (E130). Die Leitstelle muss das in ihrer Interventionsanweisung kennen.

Gut zu wissen: Die Verweildauer ist **je Kamera** getrennt – an der Kasse kurz, im Lager lang. Höherer Wert = seltener, aber sicherer. „Nur in Alarmzonen“ hält Durchgangsbereiche (Gang, Eingang) aus der Meldung.

Gut zu wissen: Einstecken: Verweilen läuft ohne GPU. Mögliches Einstecken erkennt die **Griff-Wache** mit einem Skelett aus 17 Körperpunkten (Kapitel „Aufnahmen, Snapshots & Prüfhinweise“). „Verbergen“ meldet LocalEye zusätzlich über die Bildbeschreibung (Prüfhinweis „Ware verbergen“).

3.20 Fernzugriff: sicher von überall im Browser

Mit dem **Fernzugriff** erreichst du deine Anlage(n) von unterwegs – im normalen Browser, **ohne VPN-App und ohne Portfreigabe** am Router. Die Anlage baut nur eine *ausgehende* Verbindung zu einem Vermittler (Relay) auf; von außen ist nichts direkt erreichbar. Jeder Kunde sieht ausschließlich **seine eigenen** Standorte.

1. Im Browser **https://remote.localeye.shop/remote** öffnen – oder in der Anlage unter **/anlage/** in der Karte *Fernzugriff* auf **„Fernzugriff öffnen ↗“** klicken.

2. Im Portal mit deinem **Lizenzschlüssel** anmelden (derselbe Schlüssel wie fürs Update). Hast du den zweiten Faktor eingerichtet, gehört der Code aus der Authenticator-App dazu.

3. Das Portal listet **alle Anlagen dieses Schlüssels** mit Status (online/offline). Ein Klick auf **Öffnen** startet Live-Bild und Oberfläche genau dieser Anlage: Live-Video, Aufnahmen, KI und Einstellungen von LocalEye. *Anlage verwalten* und die Zusatz-Apps (KnowSora, SearchTool, InfraQuill) sind nur im lokalen Netz erreichbar.

Gut zu wissen: Der Fernzugriff richtet sich **bei Installation und Update selbst ein**, sobald die Einrichtung abgeschlossen ist (die Anlage meldet sich eigenständig am Relay an); danach genügt für dich der Lizenzschlüssel. Die Firewall am Standort muss dafür ausgehend TCP-Port 7000 erlauben (Ziel remote.localeye.shop, Freigabe möglichst per Hostname). Abschalten lässt er sich mit dem Schieberegler in *Anlage verwalten → Fernzugriff*; wieder einschalten geht dann nur noch im lokalen Netz. Mehrere Märkte unter einem Schlüssel erscheinen automatisch alle in der Liste; Standorte mit eigenem Schlüssel nimmst du im Portal unter **„Standort hinzufügen“** dazu.

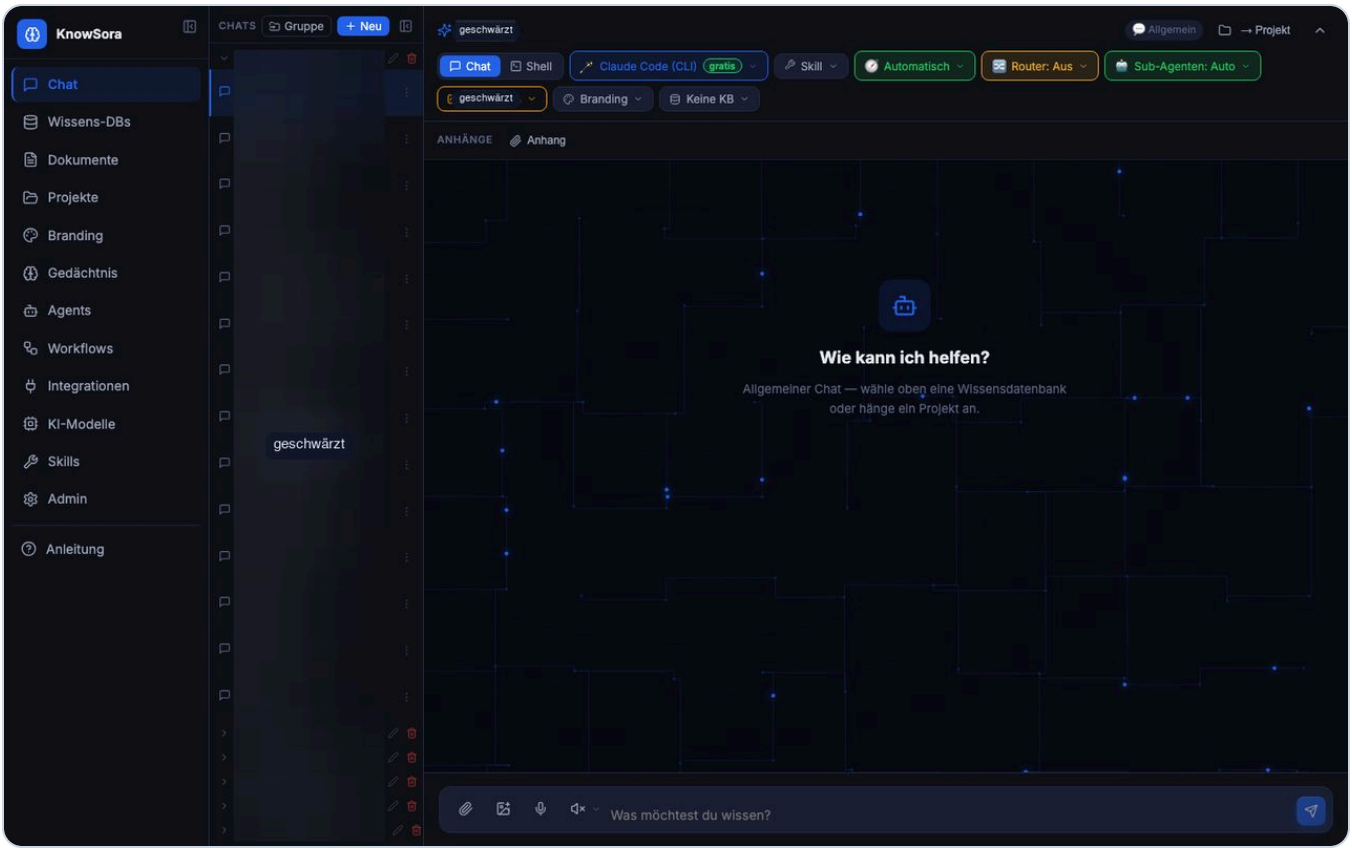
Achtung: Sicherheit: Der Zugang läuft ausschließlich über Relay und Portal, nie direkt auf das Markt-Netz oder die Kameras (Application-Layer, **kein L3-VPN** ins Kundennetz), pro Kunde getrennt: Jeder Kunde sieht im Portal nur seine eigenen Standorte. Zwischen Browser und Relay läuft alles über HTTPS/WSS, zwischen Relay und Anlage über einen TLS-Tunnel. Auf dem LocalEye-Relay wird der Inhalt im Durchlauf entschlüsselt und nicht gespeichert. Die Lizenz lässt sich zentral deaktivieren. Neue Anmeldungen und Tunnelverbindungen werden dann abgewiesen, laufende Verbindungen trennt das Relay beim nächsten Lebenszeichen der Anlage, also binnen rund 30 Sekunden.

Gut zu wissen: Zweiter Faktor: Im Portal unter *Zweiter Faktor* einrichten. Danach kommt niemand mehr nur mit dem Lizenzschlüssel hinein, auch nicht über einen verknüpften Standort. Die acht Wiederherstellungs-Codes sicher aufbewahren. Einen geschützten Standort verknüpfst du nur mit seinem Code.

4 KnowSora

KnowSora ist der KI-Arbeitsplatz der Suite: Chat mit Neuronenkernen, Dokumenten und Projekten, wahlweise mit lokalen Modellen oder externen Anbietern. Du erreichst es über die Sidebar von LocalEye (*Apps* → *KnowSora*) oder direkt unter `/knowsora/`.

4.1 Oberfläche & Chat



Chat: links Chats in Gruppen, oben die Kopfzeile mit Modus, KI-Anbieter, Skill, Denk-Modus, Router, Sub-Agenten, Projekt und Neuronenkern.

1. **+ Neu** legt einen Chat an; per **Gruppe** ordnest du Chats in Ordner.
2. In der Kopfzeile den **KI-Anbieter** wählen (lokal, Cloud oder eine CLI wie Claude Code).
3. Optional einen **Neuronenkern** (KB) oder ein **Projekt** anhängen — dann antwortet die KI mit deinen Dokumenten bzw. arbeitet in deinem Code.

4. **Denk-Modus**: Automatisch, Denken, Gründlich, Ultrathink — und bei Claude Code zusätzlich *Ultracode* (höchste Gründlichkeit, sehr hoher Verbrauch).

5. Frage eintippen; Anhänge (Dateien, Bilder, Sprache) über die Symbole links im Eingabefeld.

6. **PDF** exportiert den Chat; → **Projekt** kopiert erzeugte Dateien in den Projektordner.

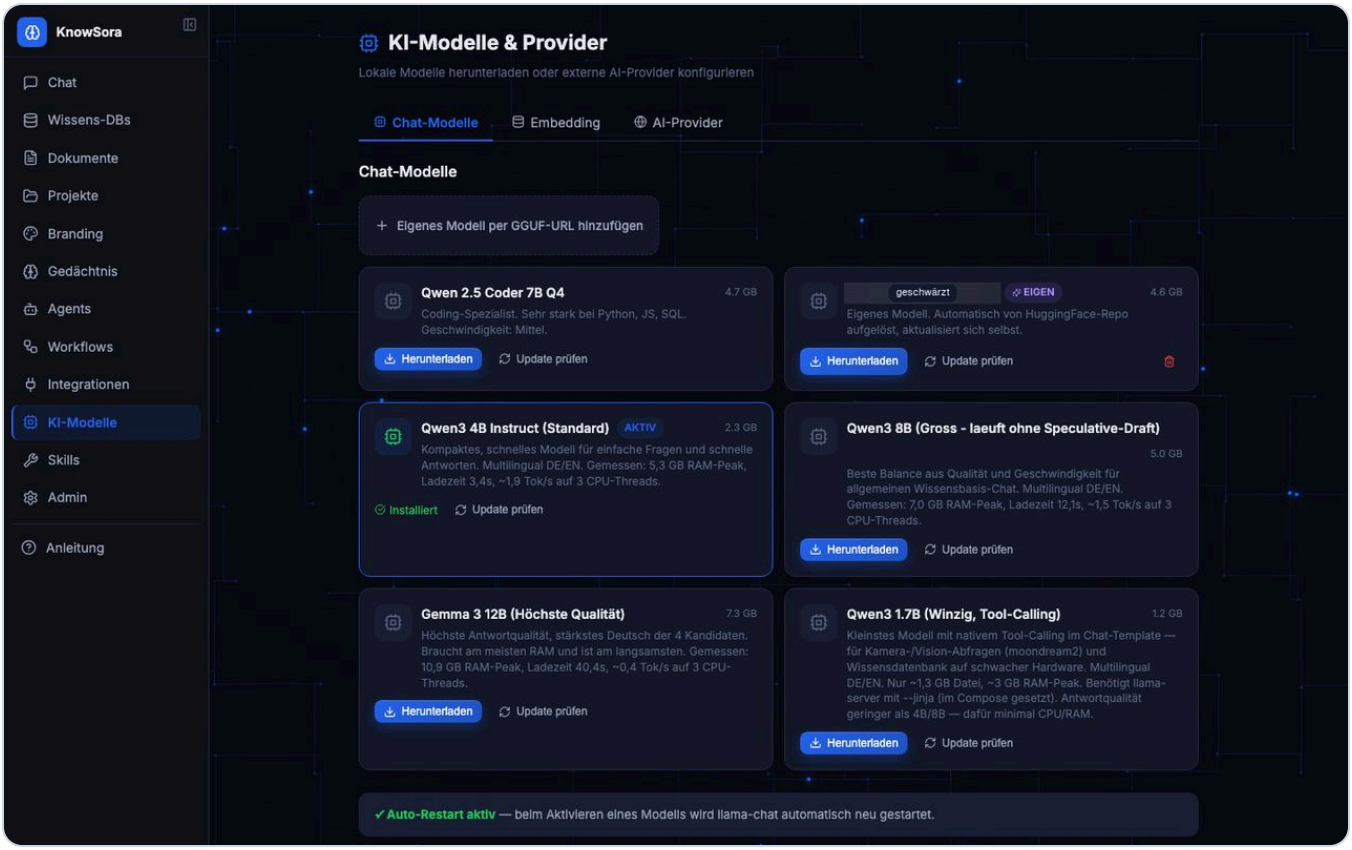
Tipp: Auf dem Smartphone ist die zweite Kopfzeile eingeklappt — der Pfeil rechts oben blendet sie ein.

Gut zu wissen: Externe Bilder: Externe Bilder in KI-Antworten lädt KnowSora erst auf Klick. Statt des Bildes steht ein Platzhalter mit der Zieladresse und dem Knopf **Bild laden**. Bilder der eigenen Anlage erscheinen wie bisher sofort.

Achtung: Anmeldung über LocalEye: Öffnet jemand einen KnowSora-Anmeldelink, der nicht in diesem Browser gestartet wurde (zum Beispiel einen weitergeleiteten Link), fragt KnowSora mit dem Kontonamen nach (*Anmeldung bestätigen*). Bestätige nur, wenn du KnowSora gerade selbst über LocalEye geöffnet hast. Dieselbe Rückfrage kommt, wenn in diesem Browser noch ein anderes KnowSora-Konto angemeldet ist, zum Beispiel an einem gemeinsam genutzten Rechner, wenn die LocalEye-Sitzung des Vorgängers abgelaufen ist. Mit **Abbrechen** bleibt das bisherige Konto angemeldet. Wird ein Konto in LocalEye umbenannt, übernimmt KnowSora den neuen Namen bei der nächsten Anmeldung über die Kachel. Chats und Dokumente bleiben beim Konto.

Gut zu wissen: Browser: Auf Safari und iOS unter Version 16.4 zeigt KnowSora Antworten ohne Tabellen- und Durchstreich-Formatierung an. Ab Version 16.4 erscheint alles vollständig.

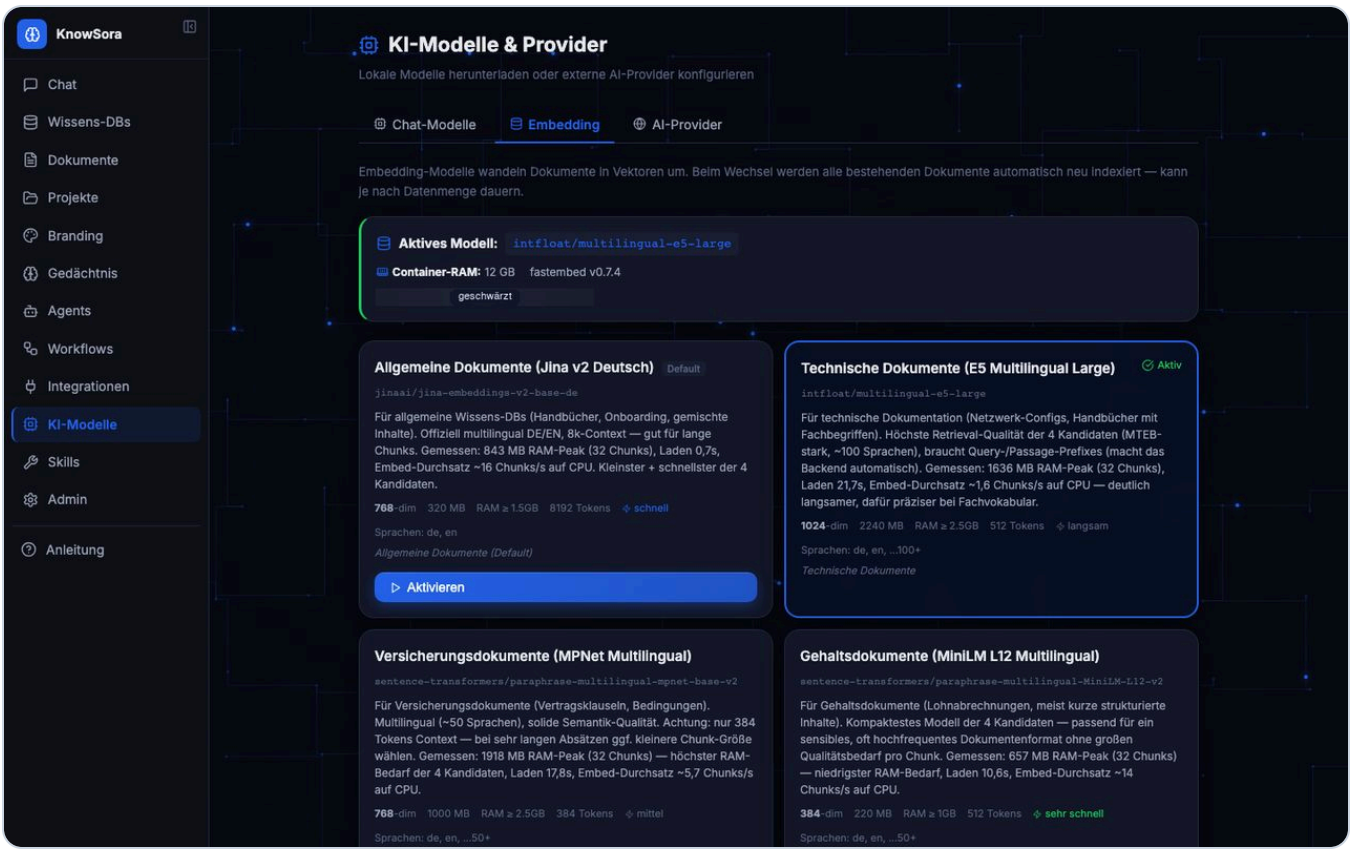
4.2 KI-Modelle & Anbieter



Chat-Modelle: lokale GGUF-Modelle herunterladen, aktivieren, aktualisieren; eigene per URL ergänzen.

Lokale Modelle

1. **KI-Modelle** → **Chat-Modelle**: ein Modell **Herunterladen** (Größe und Speicherbedarf stehen dabei).
2. **Aktivieren** — der lokale llama-Server startet automatisch neu.
3. Unter **Embedding** das Modell für die Dokumentensuche wählen (Standard: deutsch-optimiert).

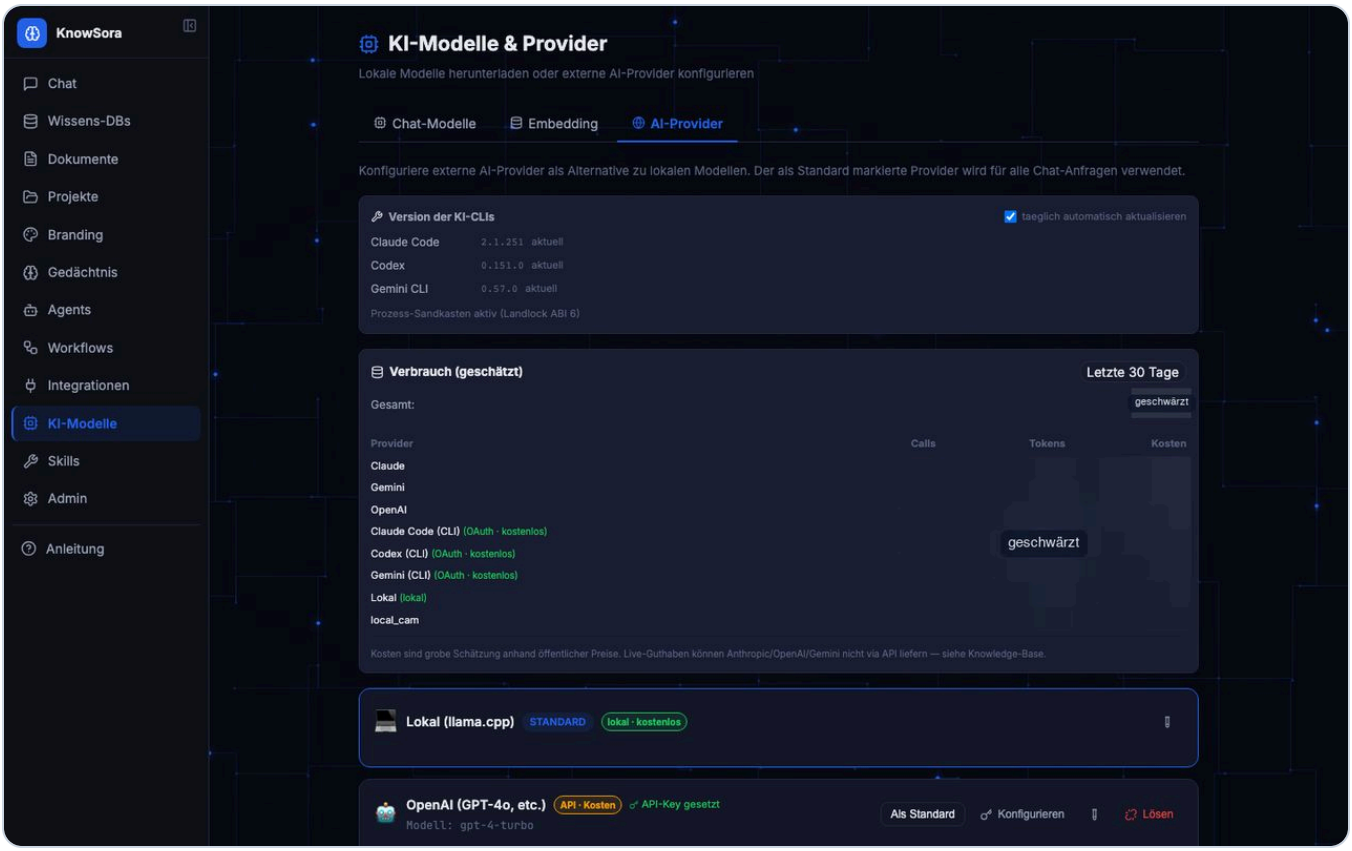


Achtung: Warum die erste Frage nach einem Neustart lange dauert. Bevor ein lokales Modell antwortet, muss es den ganzen Prompt einmal durchrechnen — bei einem Skill mit Werkzeugen sind das schnell ein paar tausend Wörter, und auf einem NAS dauert das **Minuten**. Danach merkt sich der Modell-Server diesen Anfang: jede weitere Frage im selben Skill ist in **Sekunden** da. Gemessen an einer echten Anlage: 277 Sekunden beim ersten Mal, 4 Sekunden danach.

Tipp: Die Anlage wärmt sich dafür selbst vor: kurz nach dem Start rechnet sie diesen Anfang im Hintergrund durch, damit Ihre erste echte Frage bereits schnell ist — und wiederholt das, solange die Anlage benutzt wird. Wird tagelang nicht gearbeitet, pausiert sie. Das kostet **keine Token und kein Geld**: es läuft ausschließlich auf dem lokalen Modell im Haus. Abschalten mit `PROMPT_WAERMER=0` in der `.env`.

Achtung: Zu viele Werkzeuge in einem Skill sprengen das lokale Modell. Jede Werkzeugbeschreibung steht mit im Prompt. Ein Skill mit rund 25 Werkzeugen brauchte in der Praxis 12.114 Wörter-Einheiten, das lokale Modell fasst 8.192 — es kann dann *gar nicht* antworten und meldet einen Fehler. Die Fehlermeldung nennt seit September 2026 den Grund und die Abhilfe: im Skill weniger Werkzeuge auswählen, oder für diesen Skill einen Cloud-Anbieter mit größerem Fenster wählen.

Externe Anbieter und KI-Kommandozeilen



AI-Provider: API-Schlüssel je Anbieter, OAuth-Login der Kommandozeilen, Versionen mit Auto-Update, Verbrauch.

1. **KI-Modelle → AI-Provider**. Für OpenAI, Gemini, Claude, Grok, OpenRouter den **API-Schlüssel** eintragen, **Testen**, **Standard** setzen.

2. **Claude Code / Codex / Gemini CLI**: Statt Schlüssel kannst du dich per **Login** mit deinem Abo anmelden (OAuth im eingebauten Terminal). Der Chat nutzt dann das Abo, nicht die API. Der Verbindungstest startet die Kommandozeile im selben Sandkasten wie einen Chat-Lauf.

3. **Weiterer Login** legt einen zweiten Eintrag derselben CLI mit eigener Anmeldung an — z. B. das Abo eines Freundes; im Chat als eigener Anbieter wählbar. Ist der Login-Ordner eines weiteren Claude-Code-Logins nicht beschreibbar, nutzt KnowSora nicht mehr still den Haupt-Login, sondern behandelt den Eintrag als nicht angemeldet. Rechte mit `./manage.sh update` reparieren und den Eintrag neu anmelden.

4 **Version der KI-CLIs**: zeigt installierte und neueste Version; *Aktualisieren* oder das Häkchen *täglich automatisch aktualisieren*. Darunter steht, ob der **Prozess-Sandkasten** aktiv ist, eine aufklappbare Zeile mit den Stufen (*Temp je Lauf*, *CLI-Schutz*, *Claude verwaltet*, *Mitschrift der Chat-Läufe*) und der Stand der Netzregeln. Kann KnowSora den Hilfetext einer KI-Kommandozeile nicht lesen, zum Beispiel während einer Aktualisierung, steht das gelb im Kasten. Die Schutzschalter dieser Kommandozeile sind dann aus, KnowSora versucht es nach fünf Minuten erneut.

Gut zu wissen: Der Sandkasten (Landlock) sorgt dafür, dass KI-Kommandozeilen und Werkzeuge nur ihren Projektordner sehen — nicht die Datenbank, Schlüssel oder Projekte anderer Nutzer. Fehlt er (sehr alter Kernel), zeigt der Kasten eine Warnung. Jeder Lauf bekommt außerdem ein eigenes Temp-Verzeichnis ($TMPDIR), das danach gelöscht wird. /tmp selbst ist für KI-Kommandozeilen und Werkzeuge nicht beschreibbar. Aus der gemeinsamen CLI-Ablage starten in Läufen nicht mehr: bei Claude Code Haken und MCP-Server, bei Codex-Chatläufen die config.toml (MCP-Server, notify), bei Gemini MCP-Server und Erweiterungen. Das gilt je Kommandozeile nur, wenn ihre installierte Version den Schutzschalter kennt; den Stand zeigt die aufklappbare Zeile im Kasten (Schalter „ja“ oder „nein“). Was Einstellungsdateien im Projekt- und Login-Ordner darüber hinaus bewirken, begrenzt die Zielbindung (Hinweis unten): Bei Claude Code gelten Adresse, Proxy, Zertifikate, Anbieter-Modus, Telemetrie und Schlüssel-Hilfsprogramme (apiKeyHelper und verwandte) aus Einstellungsdateien nicht; die Datei-Werkzeuge von Claude Code (Lesen, Bearbeiten) fassen die Anmeldedateien im Login-Ordner nicht mehr an. In Chat-Läufen ohne Shell-Freigabe erreichen Lese-, Such- und Schreibwerkzeuge den Login-Ordner gar nicht: Dateien neben einer Fähigkeit des Login-Ordners lassen sich dort nicht nachlesen, die Fähigkeit selbst, die Anweisungsdatei und eigene Befehle lädt Claude Code weiter. Bei Codex laden Chat-Läufe und Verbindungstest weder die config.toml noch die .env des Login-Ordners. In der Shell-Sitzung gelten beide weiter; dort sind notify und Telemetrie an eine eigene Adresse abgeschaltet, die Adressen des Abo-Zugangs stehen fest. Bei Gemini ändern .env-Dateien weder Ziel noch Zugang, Haken (hooks) und Werkzeug-Suchbefehle aus Einstellungsdateien laufen nicht, und die .gemini/settings.json im Projektordner wird nicht geladen; GEMINI.md, eigene Befehle und Fähigkeiten des Projekts gelten weiter. Das HOME eines CLI-Eintrags teilen sich alle seine Läufe: Startdateien dort (.bashrc, .profile, .gitconfig, .npmrc, Python-Pakete unter .local) wirken in jedem Lauf dieses Eintrags, auch /dev/shm und die Docker-Sandbox sind gemeinsam. Gib die Shell-Freigabe deshalb nur Personen, denen du vertraust.

Gut zu wissen: Haken für Claude Code: Claude Code führt nur noch die Haken der verwalteten Vorlage aus. Beim ersten Start nach dem Update übernimmt KnowSora die vorhandenen Haken einmalig. Die Vorlage steht unter *KI-Modelle* → *AI-Provider* im Kasten *Version der KI-CLIs* (aufklappbare Zeile, nur zum Lesen) und liegt in `knowsora/data/.knowsora_cli_verwaltung/claude-code.json`. Neue Haken trägt ein Admin in diese Datei ein; sie gelten ab dem nächsten Lauf. Jeder übernommene Haken ist an den Login-Ordner seines Claude-Eintrags gebunden und läuft nur in dessen Läufen. Haken ohne Shell-Befehl (prompt, http, agent) übernimmt KnowSora nicht; sie stehen an derselben Stelle. Ruft ein Haken ein Skript im Login-Ordner der CLI oder im Projektordner auf, warnen die Admin-Ansicht und der Selbsttest, solange das so ist. Lege das Skript dann an einen Ort, den Läufe nicht beschreiben können, und ändere den Pfad in der Vorlage, oder entferne den Haken. Steht in der Zeile *Claude verwaltet: nicht schreibbar*, ließ sich die verwaltete Datei nicht bereitstellen; dann laufen die Haken der gemeinsamen Ablage weiter.

Gut zu wissen: Schlüssel-Vermittler: Ist bei Claude Code, Codex oder Gemini CLI ein API-Key hinterlegt, bekommt die Kommandozeile ihn nicht mehr selbst, sondern eine Adresse am Gerät und einen Wegwerf-Schlüssel, der nur für diesen einen Lauf gilt. KnowSora setzt den echten Schlüssel erst auf dem Weg zum Hersteller ein. Im Lauf ist nur noch der Wegwerf-Schlüssel zu sehen; nach dem Lauf und außerhalb des Geräts ist er wertlos. Für die Anmeldung per Abo ändert sich nichts. Wird der API-Key eines Eintrags geändert, entfernt oder durch eine Adressänderung verworfen, der Eintrag gelöscht oder sein Auth-Modus umgestellt (zum Beispiel auf *Nur OAuth*), verlieren laufende Sitzungen dieses Eintrags sofort den Zugang über den alten Schlüssel. Codex nutzt den API-Key erst über den Vermittler wirklich (die installierte Fassung 0.165 wertete ihn nicht mehr aus): Im Auth-Modus *Nur API-Key* und beim Ausweichen auf den API-Key im Modus *Automatisch* (Abo-Limit erreicht) entstehen bei Codex jetzt tatsächlich Token-Kosten. Den Zustand zeigt der Kasten *Version der KI-CLIs*: *Schlüssel-Vermittler aktiv: API-Keys der KI-Kommandozeilen bleiben im Backend*, sonst gelb *Schlüssel-Vermittler läuft nicht* – dann starten Läufe mit API-Key nicht, bis das KnowSora-Backend neu gestartet ist, Läufe mit Abo-Anmeldung gehen weiter; der Selbsttest meldet *Schluessel-Vermittler der KI-Kommandozeilen laeuft*. Rückschalter: `KS_CLI_VERMITTLER=aus` in `knowsora/data/sandkasten-schalter.env`, danach das KnowSora-Backend neu starten; dann bekommt die Kommandozeile wieder den echten Schlüssel. Die Abo-Anmeldung hält die Kommandozeile selbst in ihrem Login-Ordner; wer die Shell-Freigabe hat, kann sie dort lesen.

Gut zu wissen: Zielbindung: Claude Code, Codex und Gemini CLI lesen Einstellungen aus dem Projektordner, in den jeder Nutzer Dateien hochladen kann, und aus ihrem Login-Ordner, den jede Shell-Sitzung beschreibt. Solche Dateien bestimmen nicht mehr, wohin eine Kommandozeile ihre Anfragen samt Zugangsdaten schickt: KnowSora legt vor jedem Start das Ziel fest (die Adresse des Herstellers, mit API-Key der Schlüssel-Vermittler am Gerät), Proxy und Zertifikate kommen nur aus der Umgebung des KnowSora-Backends, also vom Betreiber, und Einstellungen, deren Wert ein Befehl ist, laufen nicht. Was das je Kommandozeile heißt, steht im Hinweis zum Sandkasten oben. Eine eigene Adresse, ein Proxy oder ein Zertifikat in einer Einstellungsdatei einer KI-Kommandozeile wirkt deshalb nicht mehr; trage es in die Umgebung des KnowSora-Backends ein (Dienst `knowsora-backend` in der `docker-compose.yml` bzw. `docker-compose.override.yml`: `HTTPS_PROXY`, `NO_PROXY`, `NODE_EXTRA_CA_CERTS`, für ein eigenes Gateway `ANTHROPIC_BASE_URL` bei Claude Code bzw. `GOOGLE_GEMINI_BASE_URL` bei Gemini CLI, eine eigene Kopfzeile für das Gateway über `ANTHROPIC_CUSTOM_HEADERS`); ein Zertifikat muss im Container unter `/etc` liegen. Kann KnowSora einen Lauf nicht binden, startet er nicht, statt ungebunden zu laufen: Gemini CLI, wenn seine Einstellungen im Abbild des KnowSora-Backends fehlen (Meldung *Gemini CLI startet nicht: …*), Codex, wenn sich der eigene Ordner des Laufs nicht anlegen lässt, Claude Code ohne Shell-Freigabe, wenn der Pfad seines Login-Ordners Leer- oder Sonderzeichen enthält (die Ordner, die KnowSora selbst anlegt, tun das nicht). Haken für Gemini gibt es nicht mehr; Haken für Claude Code laufen wie bisher aus der verwalteten Vorlage. Es bleiben Abo-Anmeldung, Anweisungsdateien (`CLAUDE.md`, `AGENTS.md`, `GEMINI.md`) aus Projekt- und Login-Ordner, eigene Befehle und Fähigkeiten, alle bisherigen Schutzschalter und das Cloud-Projekt eines Google-Firmenkontos (`GOOGLE_CLOUD_PROJECT` aus der `.env` des Login-Ordners). Den Zustand zeigt der Kasten *Version der KI-CLIs* (*Zielbindung aktiv: …*, sonst gelb *Zielbindung unvollständig: …* mit dem, was fehlt), der Selbsttest meldet *Zielbindung der KI-Kommandozeilen aktiv*. Rückschalter: `KS_CLI_ZIELBINDUNG=aus` in `knowsora/data/sandkasten-schalter.env`, danach das KnowSora-Backend neu starten. Die Shell-Freigabe bleibt Vertrauenssache: Wer sie hat, kann im Login-Ordner Dateien ablegen, die in anderen Läufen dieses Eintrags wirken (Startdateien, bei Codex in der Shell-Sitzung config.toml und .env).

Gut zu wissen: Netzregeln: Die Netzregeln sperren für KI-Kommandozeilen, Werkzeuge und Shell-Sitzungen den direkten Weg zu den Ports der Plattform, auch über 127.0.0.1: Kamerabilder und go2rtc, LocalEye, Anlage verwalten, Datenbank und die lokalen KI-Modelle. Gesperrt werden TCP-Verbindungen; UDP regeln sie nicht. Internet (HTTPS, HTTP, SSH, DNS) und die Docker-Sandbox bleiben erreichbar. Die Ports der Plattform sind aus KI-Läufen zu keinem Ziel erreichbar; ein eigenes Projekt sollte deshalb nicht auf einem dieser Ports lauschen (z. B. 5000, 5050, 8554). Voraussetzung ist ein Linux-Kernel ab 6.7; sonst steht im Kasten der CLI-Versionen *Netzregeln nicht verfügbar* und alles läuft wie bisher. Neuere Linux-Kerne lassen Umwege um diese Sperre zu (MPTCP, TCP Fast Open). KnowSora misst das bei jedem Start. Ist ein Umweg messbar, lehnt ein Filter in KI-Läufen MPTCP, TCP Fast Open und SMC-Sockets ab und sperrt dort io_uring und die 32-Bit-Sammelaufrufe für Sockets. Der Filter wird nur benutzt, wenn ein Probelauf beim Start ihn bestätigt. Was trotzdem offen bleibt, steht gelb im Kasten der CLI-Versionen und im Selbsttest (*Netzregeln greifen für TCP, lassen sich aber umgehen*). Scheitert ein KI-Lauf an den Netzregeln, die Zeile `KS_SANDBOX_NETZ=off` in `knowsora/data/sandkasten-schalter.env` eintragen und das KnowSora-Backend neu starten.

Gut zu wissen: Startet eine KI-CLI einen Hintergrundprozess (zum Beispiel einen Entwicklungsserver), läuft er nach dem Ende des Laufs bewusst weiter. KnowSora vermerkt ihn einmal im Backend-Protokoll („Hintergrundprozess eines beendeten CLI-Laufs läuft weiter“) und räumt beendete Reste selbst ab. Das gilt, wenn die Kommandozeile von selbst zu Ende ist (in der Shell mit `/exit`). Stoppst du einen Chat-Lauf oder eine Shell-Sitzung, während die KI-Kommandozeile noch läuft, beendet KnowSora auch die Programme, die der Lauf gestartet hat. Dasselbe gilt, wenn ein Lauf sein Zeitlimit erreicht, die Shell-Freigabe entzogen wird oder eine Shell-Sitzung abläuft (60 Minuten ohne verbundenen Browser, spätestens nach 8 Stunden), und für den Stopp eines Hintergrund-Auftrags der Projekt-Shell (`project_shell_stop`). Container in der Docker-Sandbox laufen in jedem Fall weiter.

4.3 Neuronenkerne & Dokumente



Neuronenkerne: jeder Neuronenkern ist eine Datenbank, jeder Punkt ein Dokument; oben Übersicht/Status/Neuronenkosmos.

Der Neuronenkosmos — und das Eintauchen in 3D

Die Ansicht **Neuronenkosmos** zeigt Ihr Wissen als Sternenhimmel: jeder Neuronenkern ein leuchtender Stern, jedes Dokument ein Punkt in seinem Umlauf. Mit dem Knopf **3D** (oben, nur im Neuronenkosmos) tauchen Sie hinein:

1. **Klick auf einen Kern** fliegt die Kamera zu ihm — die Dokumente umschwirren Sie. Ein zweiter Klick aus der Nähe öffnet den Kern wie gewohnt.

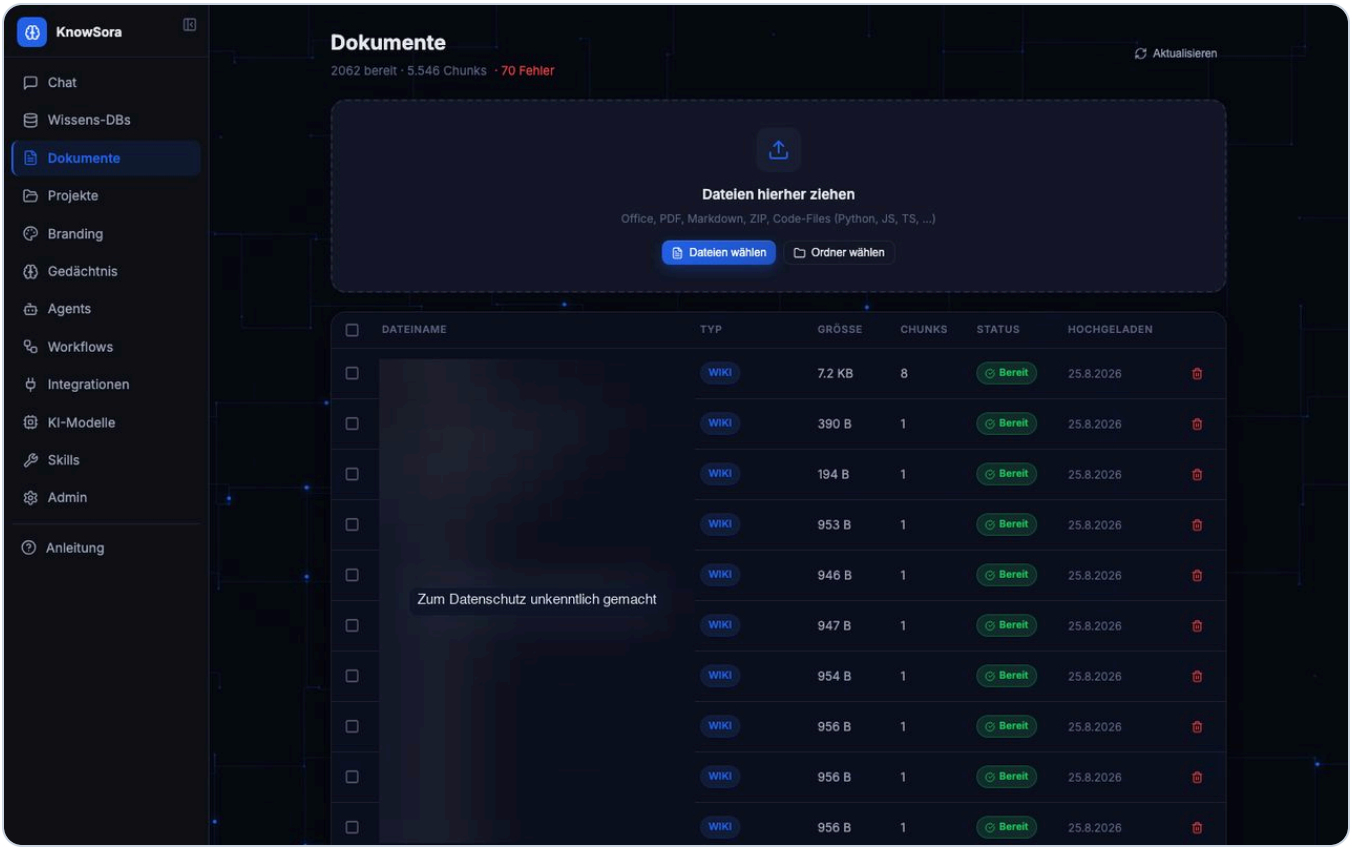
2. **Ziehen** dreht den Raum, **Scrollen** fährt hinein — und zwar dorthin, wo der Mauszeiger steht. Nach dem Eintauchen heißt Ziehen *Umsehen* und Scrollen *Fliegen*; der Mauszeiger ist dabei der Steuerknüppel: Zeiger am Rand fliegt eine Kurve, Zeiger in der Mitte geradeaus.

3. Die Knopfleiste rechts fährt mit **±** vor und zurück; ⛶ bringt Sie jederzeit zur Gesamtübersicht zurück, der Knopf **2D** zur flachen Ansicht — die Wahl merkt sich der Browser.

Gut zu wissen: Die 3D-Ansicht lädt ihre Grafikbibliothek erst beim ersten Umschalten (~700 KB) und rechnet komplett im Browser des Betrachters — das NAS kostet sie nichts. Die 2D-Ansicht behält ihre eigenen Effekte (künstliches Gehirn, Blitz-Kaskaden); beide zeigen dieselben Daten.

Tipp: Die Verbindungen ergeben Sinn. Dokumentpunkte, die inhaltlich zusammengehören, sind durch echte Kanten verbunden — berechnet aus der Nähe ihrer Einbettungsvektoren, ohne einen einzigen KI-Aufruf, auch über Kerngrenzen hinweg (kernübergreifende Bahnen leuchten wärmer). In 2D laufen die Blitz-Kaskaden über diese Bahnen, in 3D fließen Partikel darauf. Je Dokument werden nur die zwei nächsten Nachbarn verbunden, insgesamt höchstens 400 Kanten — der Kosmos bleibt lesbar statt zum Spinnennetz zu werden.

1. **Neuronenkerne** → **+ Neuer Neuronenkern**: Name, Symbol, Beschreibung. Optional Freigabe für andere Nutzer.
2. **Dokumente**: Dateien (PDF, Word, Text, Markdown, Bilder …) in den Neuronenkern hochladen — sie werden zerlegt, eingebettet und durchsuchbar.
3. Status beobachten (*Übersicht* → *Status*): wartend, in Arbeit, angehalten, bereit.
4. Im Chat den Neuronenkern als KB wählen — Antworten nennen die Quellen mit Trefferquote.



Dokumente: Upload, Status je Datei, Spalte „Kontext“, Neu-Indexierung.

Kontextstufe

Ist beim Neuronenkern die **Kontextstufe** eingeschaltet, schreibt ein KI-Modell zu jedem Ausschnitt ein bis zwei Sätze Einordnung aus dem Gesamtdokument. Das verbessert die Trefferquote deutlich, kostet aber je Dokument Modellaufrufe — reißt das Kontingent Ihres Anbieters mittendrin ab, werden die restlichen Dokumente *ohne* Kontext eingebettet.

1. Die Spalte **Kontext** in der Dokumentliste zeigt je Datei *voll*, *teilw.*, *ohne* oder *?* (vor September 2026 eingelesen, Zustand nicht überliefert).

2. Über der Tabelle steht die Bilanz, z. B. „🧠 *Kontext: 192 von 289 vollständig · 73 ohne*“.

3. Der Knopf **Kontext nachziehen** holt genau die Dokumente ohne Kontext nach — nicht den ganzen Neuronenkern.

Tipp: Einmal erzeugte Kontexte werden gespeichert. Wechseln Sie später das Einbettungs-Modell, läuft die Neu-Einbettung für diese Dokumente ohne einen einzigen Modellaufruf. Das gilt für alles, was seit September 2026 eingelesen wurde; für älteren Bestand wurde der Kontexttext damals noch verworfen — er wird beim ersten Wechsel einmalig neu erzeugt und dabei gespeichert. Ändert sich der Text eines Dokuments, wird sein Kontext ebenfalls neu geschrieben. Auch das erneute *Einlesen* entfällt beim Modellwechsel: Der gespeicherte Volltext wird wiederverwendet — gescannte PDFs laufen also nicht noch einmal durch die Texterkennung.

Gut zu wissen: Große Dokumente auf dem lokalen Weg: Je Dokument und Durchlauf gilt ein Zeitbudget für die Kontextstufe (Vorgabe 30 Minuten; Umgebungsvariable `CONTEXT_BUDGET_LOCAL` in Sekunden, 0 = kein eigenes Budget). Ist es aufgebraucht, wird der Rest des Dokuments ohne Kontext eingebettet und die Datei steht als *teilw.* in der Liste — durchsuchbar, nicht als Fehler. Die bereits erzeugten Kontexte sind gespeichert, und **Kontext nachziehen** setzt genau dort fort, ohne einen Aufruf zu wiederholen. So blockiert ein 130-Seiten-Dokument nicht mehr stundenlang die Warteschlange. Cloud- und CLI-Anbieter haben kein eigenes Budget, sie antworten in Sekunden.

Gut zu wissen: Überschreitet ein Dokument das harte Zeitlimit der Verarbeitung (`DOC_PROCESS_TIMEOUT_MAX`, Standard 2 Stunden), steht es sofort als Fehler in der Liste. Seine schon gespeicherten Abschnitte entfernt KnowSora danach aus der Suche. **Erneut versuchen** setzt bei den gespeicherten Kontexten fort.

Status-Ansicht: Was für alle Kerne zusammen gilt

Die Kopfkarte **Indexierung aktiv** im Neuronenkosmos zeigt, was systemweit gilt: die Warteschlange über alle Kerne, die Zahl der Worker und — falls pausiert — den Grund („Nachziehlauf angehalten“ oder „Alles angehalten (Modellwechsel)“). Sie ist für alle Nutzer sichtbar und zählt jeweils die Kerne, auf die der Nutzer Zugriff hat; nur der Knopf **Pausieren**/**Fortsetzen** ist Admins vorbehalten. Die Karten der einzelnen Neuronenkerne darunter zeigen nur noch, was diesen Kern betrifft: Fortschrittsbalken, Zähler und die gerade laufenden Dokumente.

Indexierungs-KI für alle Kerne umstellen

Welche KI die Kontextstufe rechnet — die lokale KI, ein Cloud-Anbieter oder gar keine — stellen Sie nicht mehr je Neuronenkern im Bearbeiten-Dialog ein, sondern zentral in der

Status-Ansicht des Neuronenkosmos. Die Änderung gilt ab dem nächsten Dokument; das gerade laufende wird noch mit der alten Einstellung fertig.

1. *Neuronenkosmos* → *Status*: Unter der Karte „Indexierung aktiv“ steht die Karte **Indexierungs-KI für alle Kerne** (nur für Admins).

2. Im Dropdown die KI wählen — *Aus (nur Einbettung)*, *Lokal*, *Claude Code*, *Codex*, *Gemini CLI*, *OpenRouter*, *OpenAI*, *Gemini*, *Groq* oder *DeepSeek* — und **Auf alle Kerne anwenden** drücken. Die Rückfrage nennt die Zahl der betroffenen Kerne; daneben steht der Ist-Zustand („24 eigene Kerne: alle „Lokal““ oder „gemischt“).

3. Einen einzelnen Kern stellen Sie direkt in seiner Status-Karte um: Dropdown **KI: …** neben „Verdichten“ und „Verwalten“, gespeichert wird sofort.

4. **Aus** schaltet die Kontextstufe ab und merkt sich den bisherigen Anbieter — beim Wiedereinschalten ist er wieder da. Dokumente werden dann nur eingebettet (Sekunden statt Minuten); fehlende Kontexte holt später *Kontext nachziehen* je Kern nach.

Gut zu wissen: Betrifft nur Ihre eigenen Kerne, Wiki-Kerne ausgenommen. Wiki-Seiten sind reine KI-Ausgabe und entstehen bei jedem Verdichten neu — eine Kontextstufe dort würde bei jedem Lauf die volle Modellrechnung kosten. Fremde, mit Ihnen geteilte Kerne bleiben unangetastet.

Tipp: Wenn die Maschine tagelang rechnet: Nach einem Wechsel des Einbettungs-Modells stehen alle Dokumente noch einmal an — mit lokaler Kontextstufe je Dokument ein bis zwei Minuten, bei tausend Dokumenten also Tage. Wer die Dauerlast nicht will, stellt für den Rückstau auf *Aus* oder einen Cloud-Anbieter und danach wieder auf *Lokal*. Der Normalbetrieb mit einzelnen neuen Dokumenten ist davon nicht betroffen.

Tipp: Die Liste in „Über mich“ pflegt sich selbst. Legst du einen Neuronenkern an, benennst ihn um oder löschst ihn, zieht der Abschnitt mit deinen Kernen automatisch nach. Deine eigenen Anmerkungen hinter dem Namen bleiben dabei erhalten, und der übrige Text der Seite wird nicht angefasst — er geht bei jeder Frage als Kontext mit. KnowSora ändert dabei nur die betroffene Verweiszeile: Einleitung, Unterüberschriften und die eigene Reihenfolge bleiben stehen; neue Kerne kommen hinter den letzten Verweis. Von der Seite gibt es genau eine je Nutzer; ein zweiter Anlegeversuch führt auf dieselbe.

Gut zu wissen: Neuronenkerne sind strikt je Nutzer getrennt — auch Admins sehen fremde Neuronenkerne nur bei ausdrücklicher Freigabe.

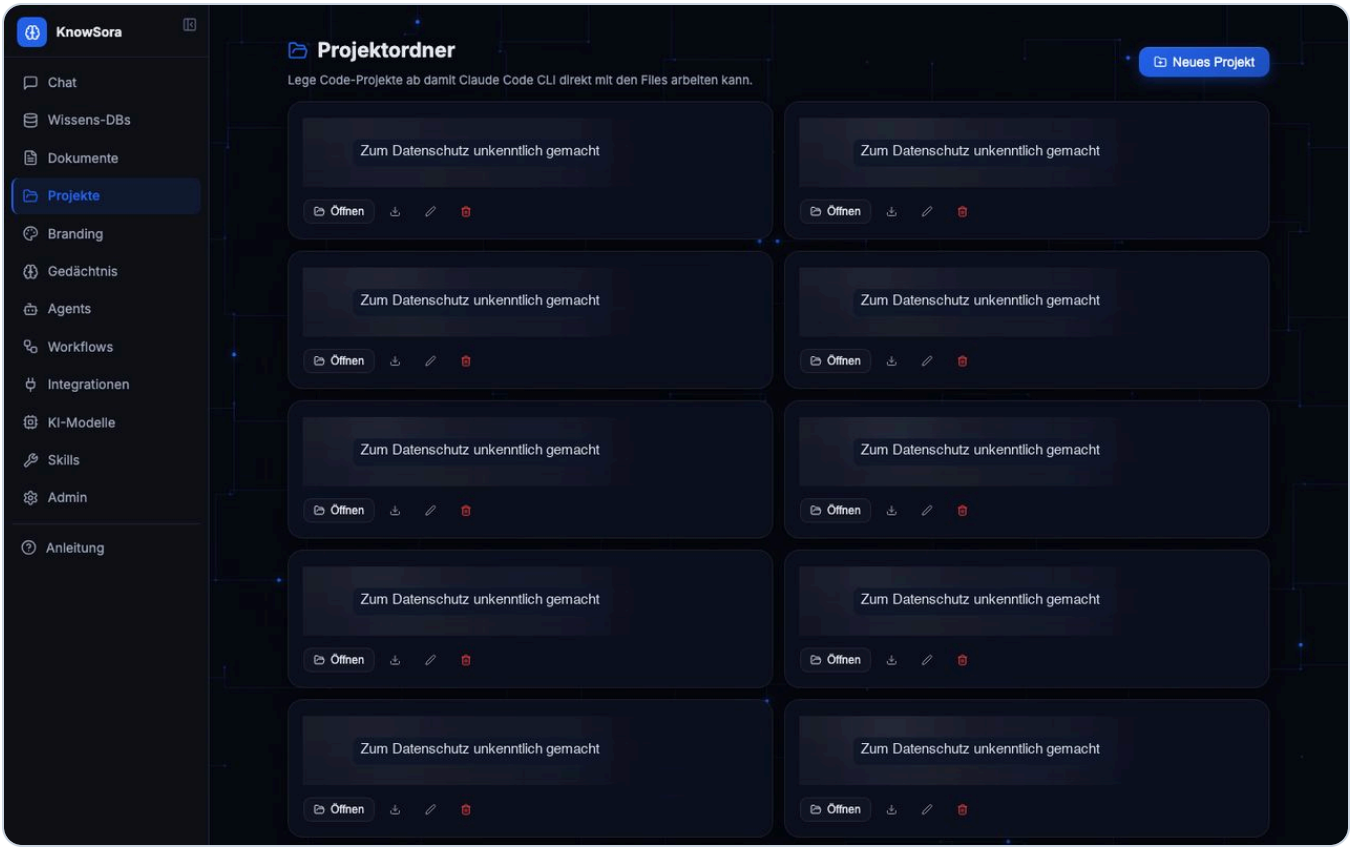
„Wartet…“ und „Angehalten“ sind zweierlei

Ein Dokument mit dem Vermerk **Wartet…** steht in der Schlange und kommt an die Reihe. Ein Dokument mit dem Vermerk **Angehalten** kommt *nicht* an die Reihe: Sie haben unter *KI-Modelle* einen laufenden Nachziehlauf gestoppt, und dieses Dokument gehört dazu.

1. Angehaltene Dokumente sind **bis zum Fortsetzen nicht durchsuchbar** — ihre Vektoren wurden vor dem Neu-Einlesen gelöscht. Die Anlage nennt Ihnen die Zahl im Klartext, damit eine vergessene Pause nicht still zu einer Lücke im Wissen wird.
2. Fortsetzen: *KI-Modelle* → *Nachziehlauf angehalten* → *Fortsetzen*.
3. **Was Sie selbst anstoßen, läuft trotzdem.** Hochladen, Wiederholen, Neu indexieren und der Projektordner-Abgleich warten nicht auf das Fortsetzen.

Gut zu wissen: Eine Ausnahme gibt es: Haben Sie gerade ein anderes **Einbettungs-Modell** aktiviert, steht bis zum Neustart wirklich alles — auch Uploads. Das ist Absicht, denn bis dahin würde jede Berechnung noch mit dem alten Modell laufen und wäre danach wertlos. Die Anlage schreibt in diesem Fall „Alles angehalten (Modellwechsel)“ statt „Nachziehlauf angehalten“.

4.4 Projekte



Projekte: eigener Ordner je Projekt, Dateizahl und Größe, Schreibrecht, Öffnen/Download/Bearbeiten.

1. **Projekte → Neues Projekt**: Name vergeben; ein Ordner wird angelegt.
2. Dateien hochladen — auch als ZIP (wird entpackt; Größenlimit 200 MB, entpackt 800 MB).
3. **Schreibbar** an/aus: mit Schreibrecht darf die KI Dateien ändern und Befehle im Projekt ausführen; ohne fragt sie nur.
4. Im Chat das Projekt anhängen — die KI arbeitet direkt mit den Dateien.

Gut zu wissen: Downloads aus Projekt-Chats: Als Download erscheinen Dateien, die der Lauf angelegt oder geändert hat, und alles, was die KI ausdrücklich verlinkt. Projektdateien, die die Antwort nur erwähnt, werden nicht mehr kopiert, und Dateien unter `outputs/` landen dort nicht ein zweites Mal. Für eine schon vorhandene Projektdatei bittest du um einen Link auf die Datei, zum Beispiel: *Gib mir den Link zu dist/release.zip.*

Projektordner durchsuchbar machen

Auf jeder Projektkarte sitzt der Schalter 🧠 **Als Neuronenkern durchsuchbar machen**. Er legt einen Neuronenkern `Projektname` an und nimmt die texttragenden Dateien des Ordners auf — danach findest du die Inhalte über Chat und Recherche wie in jedem anderen Kern. Der Kern trägt sich außerdem **von selbst in deine „Über mich"-Seite ein**.

1. Einschalten. Der erste Abgleich läuft im Hintergrund; die Karte zeigt, wo er steht.

2. **Abgleichen** nach Änderungen im Ordner: Es werden nur neue und geänderte Dateien eingelesen, verschwundene fliegen raus. Verglichen wird der Inhalt, nicht das Änderungsdatum — eine zurückgespielte Sicherung wird also richtig erkannt.

3. **Entkoppeln** trennt Projekt und Kern. Der Kern *bleibt bestehen* und bleibt durchsuchbar — löschen kannst du ihn unter *Neuronenkerne*.

Gut zu wissen: Was aufgenommen wird — und was nicht. Nur Dateien, die Text tragen: Markdown, Text, PDF, Word (.docx), Tabellen, JSON/YAML, HTML. Alte Office-Formate (.doc, .odt, .ods, .rtf) bleiben bewusst draußen — als Klartext gelesen wären sie Zeichensalat im Wissen; als .docx oder PDF gewandelt kommen sie mit. *Nicht* aufgenommen werden Archive, Bilder, Videos und die üblichen Ballastordner (`.git`, `node_modules`, `__pycache__`, `dist`). Nach jedem Lauf steht auf der Karte, wie viele Dateien übersprungen wurden und warum — ein halb eingelesener Ordner, den man für vollständig hält, wäre schlimmer als gar keiner.

Gut zu wissen: Eingebettete Bilder werden nicht mitgelesen. Eine HTML-Datei mit eingebetteten Bildern besteht fast nur aus Bilddaten — an einem echten Handbuch gemessen: 98,7 % Bilddaten, 0,8 % Text. Solche Daten werden vor dem Zerlegen durch einen Platzhalter ersetzt, der Typ und Größe nennt. Sonst stünden Bruchstücke davon später als Suchtreffer in Ihren Antworten.

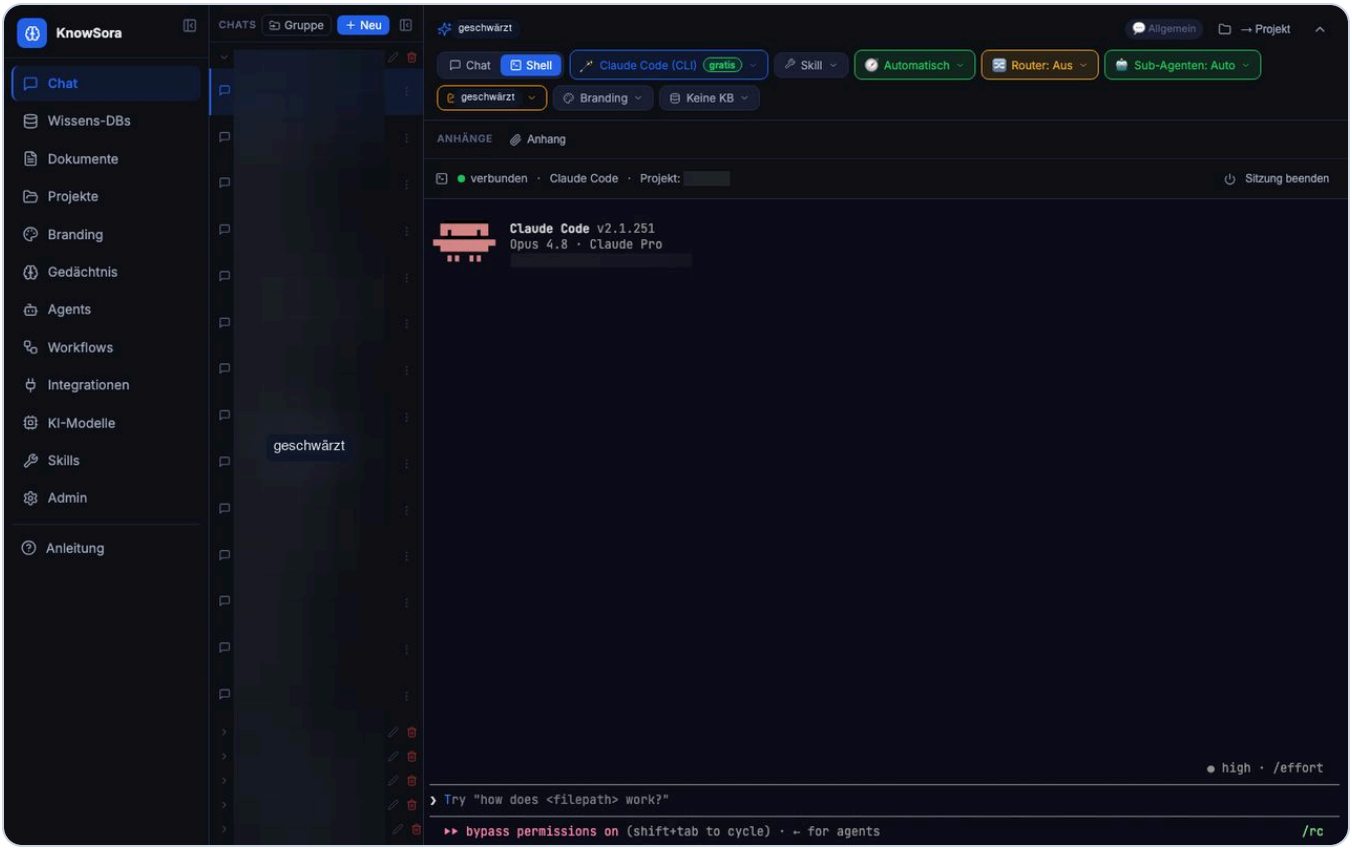
Tipp: Quellcode ist standardmäßig nicht dabei. Er zerfällt beim Zerlegen in Ausschnitte schlecht; für Code ist der Weg über Projektordner plus KI-Kommandozeile der bessere. Wer ihn trotzdem im Kern haben will, setzt vor dem Einschalten das Häkchen *auch Quellcode aufnehmen*.

Achtung: Die Kontextstufe bleibt bei Projekt-Kernen aus. Sie kostet je Ausschnitt einen Modellaufruf, und ein Projektordner hat schnell einige hundert. Du kannst sie am Kern selbst einschalten — dann siehst du auch, was du tust. Der Deckel von 500 Dateien bzw. 200 MB je Lauf zählt nur *neue* Arbeit — was schon im Kern liegt, läuft daran vorbei, mehrere Läufe arbeiten sich also durch jedes Projekt. Gelöscht wird nur, was der Abgleich selbst angelegt hat; von Hand hineingelegte Dokumente bleiben unberührt, und ein gedeckelter Lauf löscht gar nichts.

4.5 Chat-Modus und Shell-Modus

Bei den KI-Kommandozeilen (Claude Code, Codex, Gemini CLI) gibt es zwei Arbeitsweisen:

- **Chat** — Frage, Antwort, Werkzeuge im Hintergrund. Wie jeder andere Anbieter.
- **Shell** — die CLI läuft *interaktiv* als Terminal im Projektordner: Slash-Befehle wie `/model`, `/effort`, `/effort ultracode`, eigene Bash. Genau wie am eigenen Rechner.



Shell-Modus: Claude Code als echtes Terminal im Browser; die Sitzung überlebt ein Neuladen der Seite.

1. Voraussetzungen: ein CLI-Anbieter im Chat, ein angehängtes **Projekt** und die **Shell-Freigabe** (Admins immer; andere Nutzer per LocalEye-Benutzerverwaltung). Die Shell nutzt die CLI-Anmeldung des Betreibers — wer sie hat, kann diese Anmeldung lesen; dieselbe Freigabe braucht es für `python_exec` / `project_run_shell` in Skills, Agenten und Workflows.

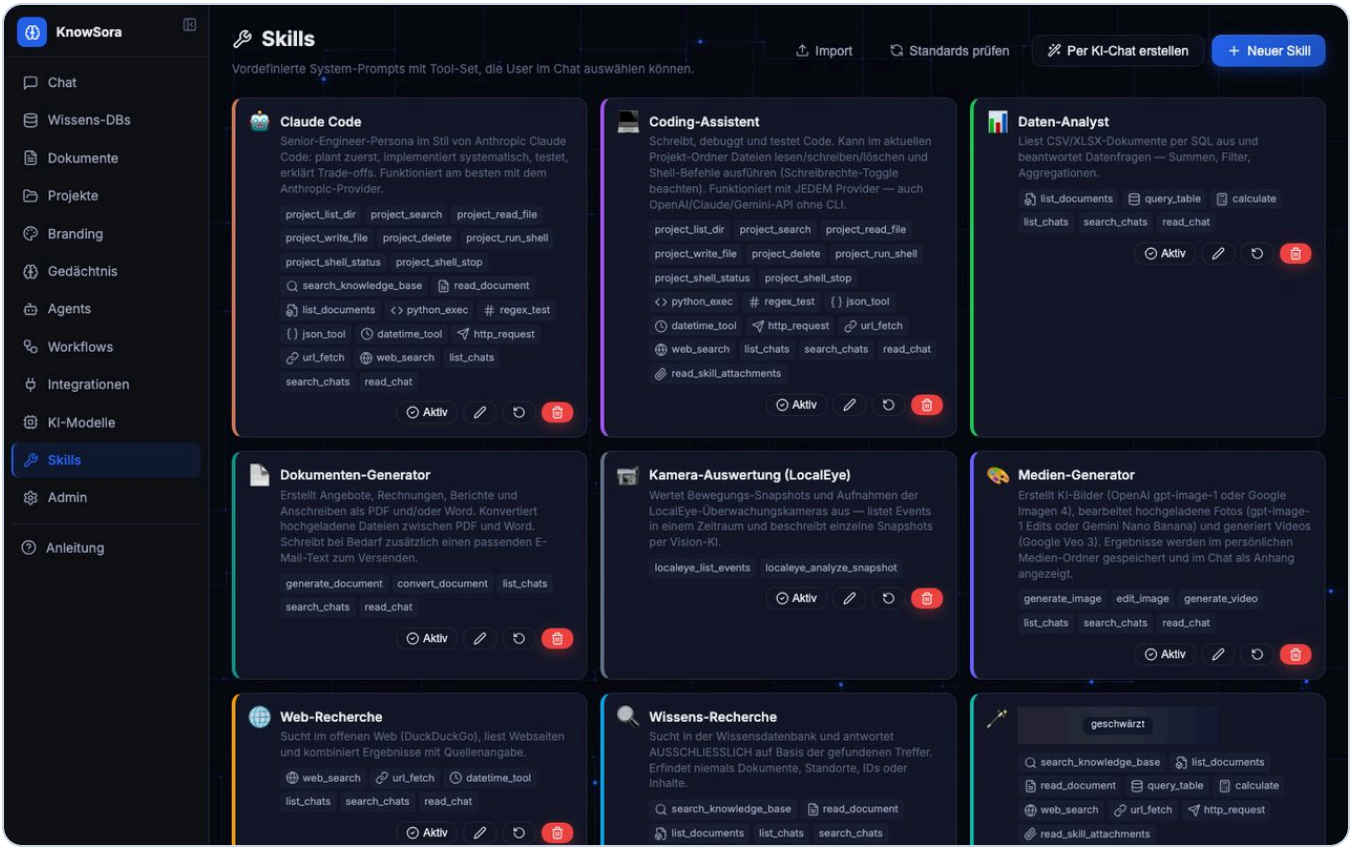
2. Oben links **Shell** wählen — die Sitzung startet im Projektordner.

3. Arbeiten wie gewohnt; **Sitzung beenden** schließt die CLI. Nach 60 Minuten ohne Verbindung endet sie automatisch.

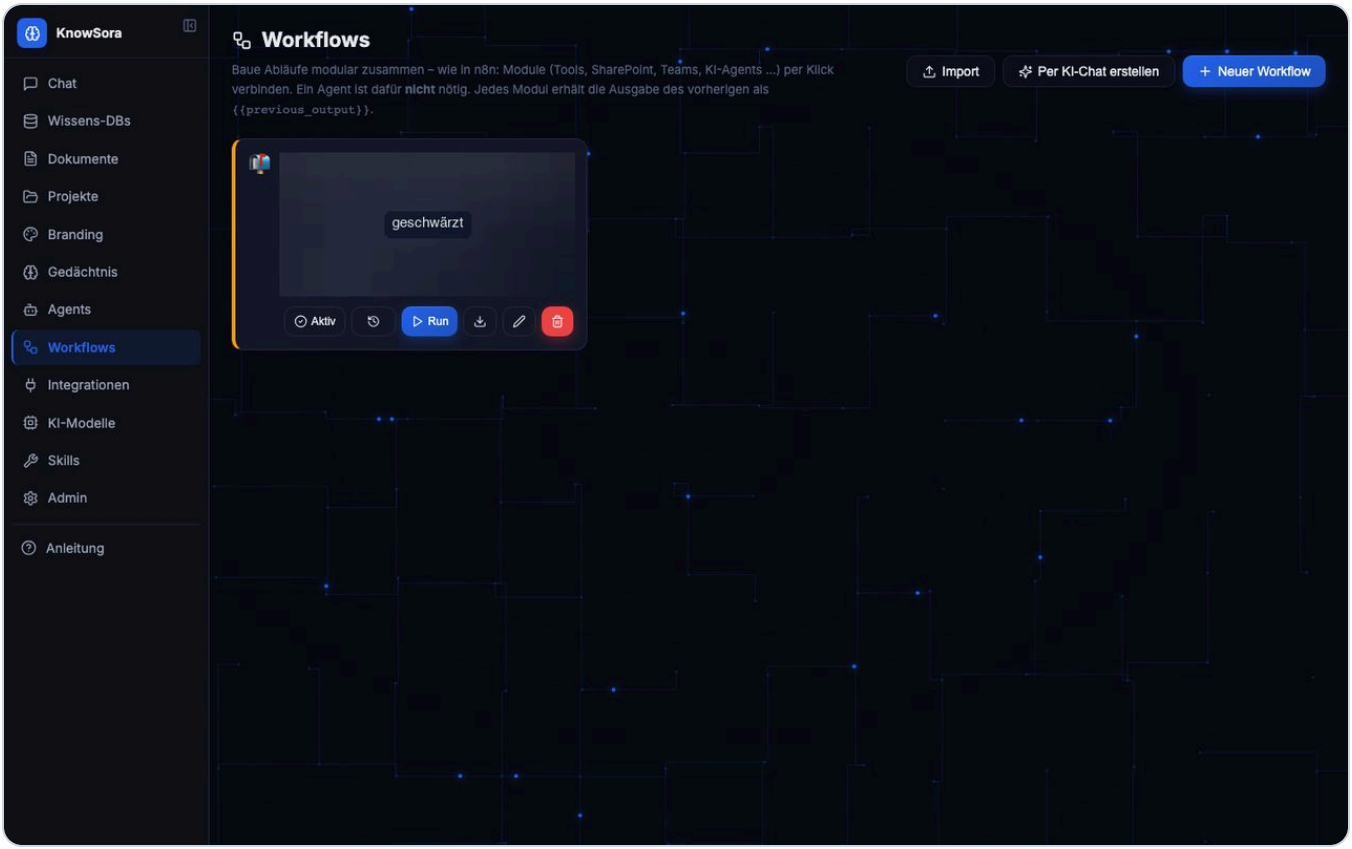
Gut zu wissen: Stockt die Ausgabe länger als 5 Sekunden, trennt KnowSora die Verbindung. Mit **Neu verbinden** geht es an derselben Stelle weiter, die bisherige Ausgabe wird nachgespielt.

Tipp: Der Denk-Modus *Ultracode* gilt für eine *neue* Sitzung; eine laufende wechselt in der TUI mit `/effort ultracode`.

4.6 Skills, Workflows, Agents, Integrationen, Gedächtnis



Skills: vordefinierte Rollen (System-Prompt + Werkzeuge), die der Chat automatisch oder auf Wunsch nutzt.

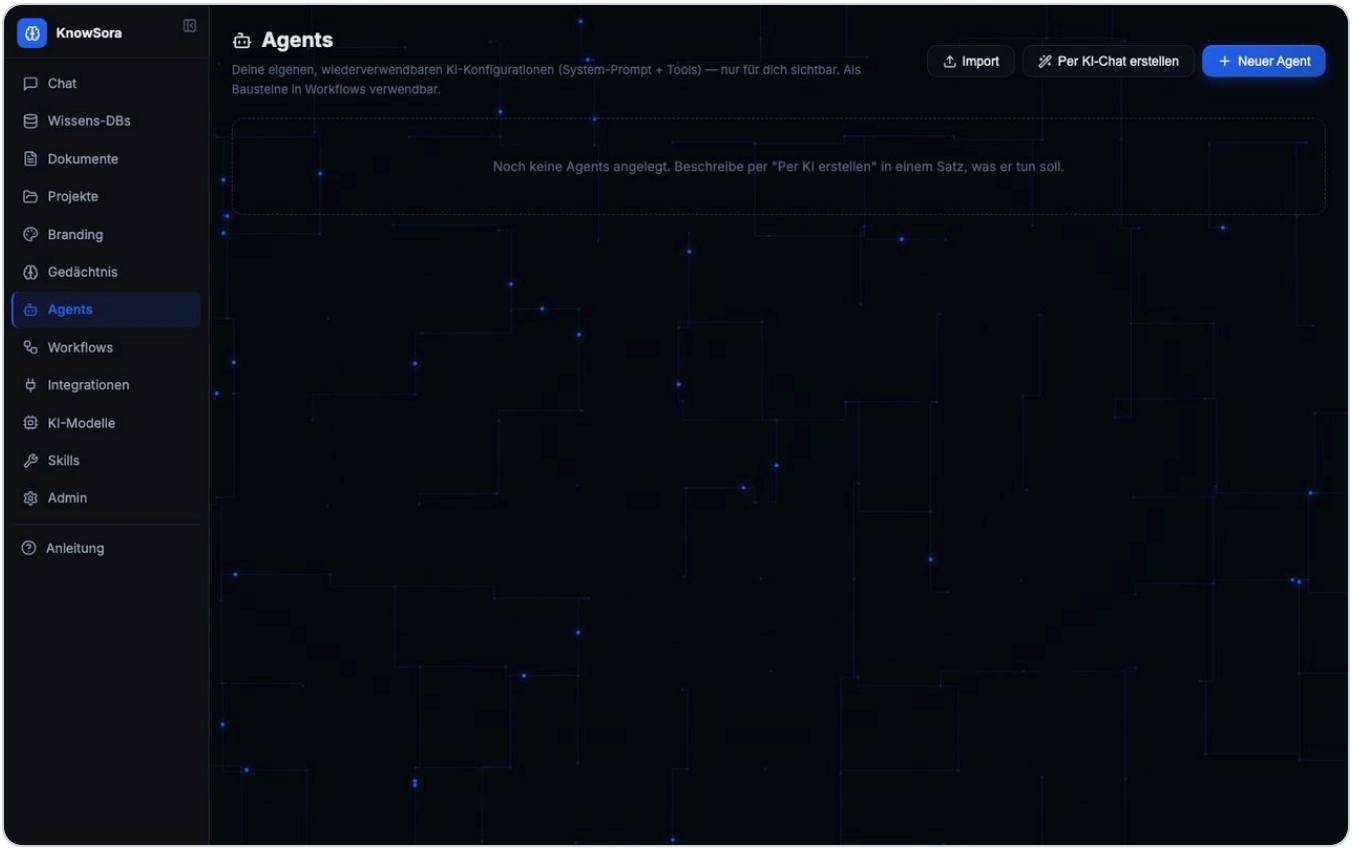


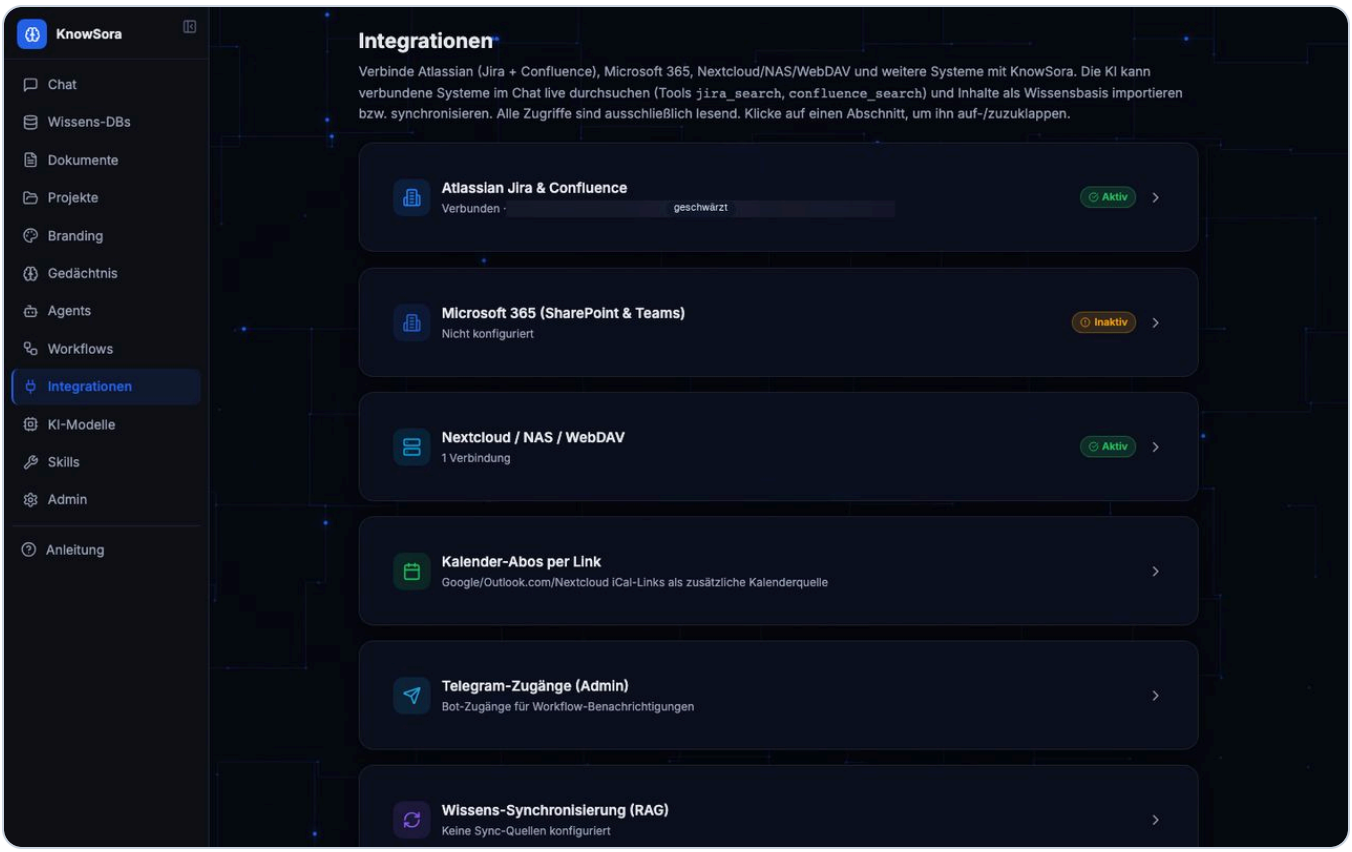
Workflows: zeitgesteuerte oder ausgelöste Abläufe, z. B. tägliche Zusammenfassungen per Telegram.

Skills, Agenten und Workflow-Schritte mit den Werkzeugen `python_exec` oder `project_run_shell` (Code bzw. Shell im Projektordner) kann seit dem 17.09.2026 nur anlegen und ausführen, wer die **Shell-Freigabe** hat (Admins immer, andere per LocalEye-Benutzerverwaltung) — ohne Freigabe lehnt KnowSora das Speichern mit einer klaren Meldung ab.

Zeitplan-Workflows holen verpasste Termine nach: Fällt die Startzeit (z. B. 08:00) oder ein Intervall-Termin in einen Neustart oder ein Update, läuft der Workflow innerhalb von 10 Minuten danach einmal nach. Ältere verpasste Termine entfallen, und es startet nie mehr als ein Lauf auf einmal. Ein neuer, wieder eingeschalteter oder geänderter Zeitplan läuft zum nächsten Termin. Liegt die Uhrzeit heute schon zurück, wird sie nicht nachgeholt. Die Uhrzeit eines Zeitplans liegt zwischen 00:00 und 23:59. Andere Werte nimmt KnowSora nicht an. Ein älterer Zeitplan mit einer solchen Uhrzeit ruht, bis er neu gespeichert wird.

Kamera-Skill: Arbeitet die Kameraüberwachung gerade mit Prüfhinweisen oder einer Griff-Prüfung am Bildmodell, wartet die Bildanalyse kurz oder bittet um einen neuen Versuch. Das ist gewollt: Die Kamera-KI hat Vorrang.





Integrationen: Kalender, Mail, Jira u. a. — Zugangsdaten werden verschlüsselt gespeichert.

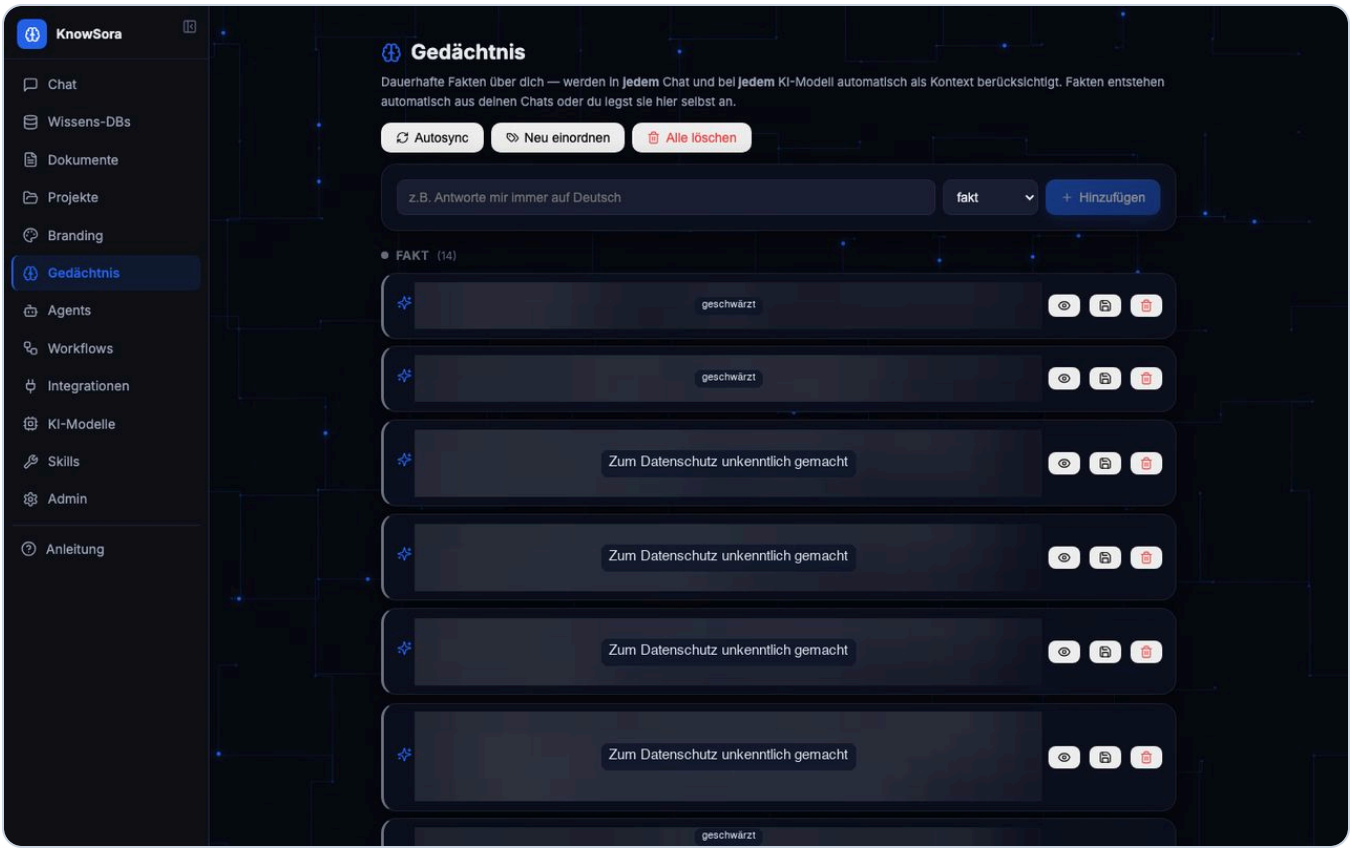
Firmen-Integrationen: Die hier hinterlegten Zugänge gelten für die ganze Firma. Wer sie nutzen darf, legt der Administrator je Konto in der Benutzerverwaltung fest (Abschnitt „Admin & Branding“). Aus einer Sitzung von außerhalb des Verwaltungsnetzes stehen sie auch Administratoren nicht zur Verfügung; geplante Workflows laufen weiter.

Neues Ziel, Geheimnis neu: Ein gespeichertes Passwort, ein Token oder ein Schlüssel gilt nur für das Ziel, für das er eingegeben wurde. Wer den WebDAV-Server, die Jira-Site oder bei Microsoft 365 den Tenant oder den Benutzernamen ändert, gibt Passwort oder Token im selben Schritt neu ein; sonst speichert KnowSora die Änderung nicht und meldet das. Ändert sich die Adresse eines KI-Anbieters, verwirft KnowSora den gespeicherten Schlüssel. Weicht die Adresse eines KI-Anbieters von der Vorgabe ab, zeigt seine Karte unter *KI-Modelle → AI-Provider → Konfigurieren* das Feld **Base URL (weicht von der Vorgabe ab)**; leer lassen stellt die Vorgabe wieder her. Ändern zwei Sitzungen dieselben Zugangsdaten gleichzeitig, verarbeitet KnowSora sie nacheinander: Ein neues Geheimnis gilt immer für das Ziel aus derselben Eingabe. Unter den Adressfeldern von Jira, WebDAV und der KI-Anbieter steht die Zeile *Zugangsdaten gehen an: …* mit dem Rechner in voller Länge; das Feld selbst zeigt bei einer langen Adresse nur einen Ausschnitt. Ist der Rechner nicht eindeutig zu erkennen (zum Beispiel Anmeldedaten vor „@“), steht dort in Fehlerfarbe *Der Rechner dieser Adresse ist nicht eindeutig zu erkennen. Bitte die Adresse neu eintragen.*, und Speichern weist die Adresse ab. In der Karte eines KI-Anbieters steht die Adresse vor dem API-Key. Eine eigene Adresse wirkt

nur bei OpenAI, xAI Grok, OpenRouter und *Custom*, im Chat ebenso wie bei Kontext-Erzeugung und Wiki-Verdichtung; bei den übrigen Anbietern sagt die Karte *Dieser Anbieter nutzt die eingetragene Adresse nicht.*

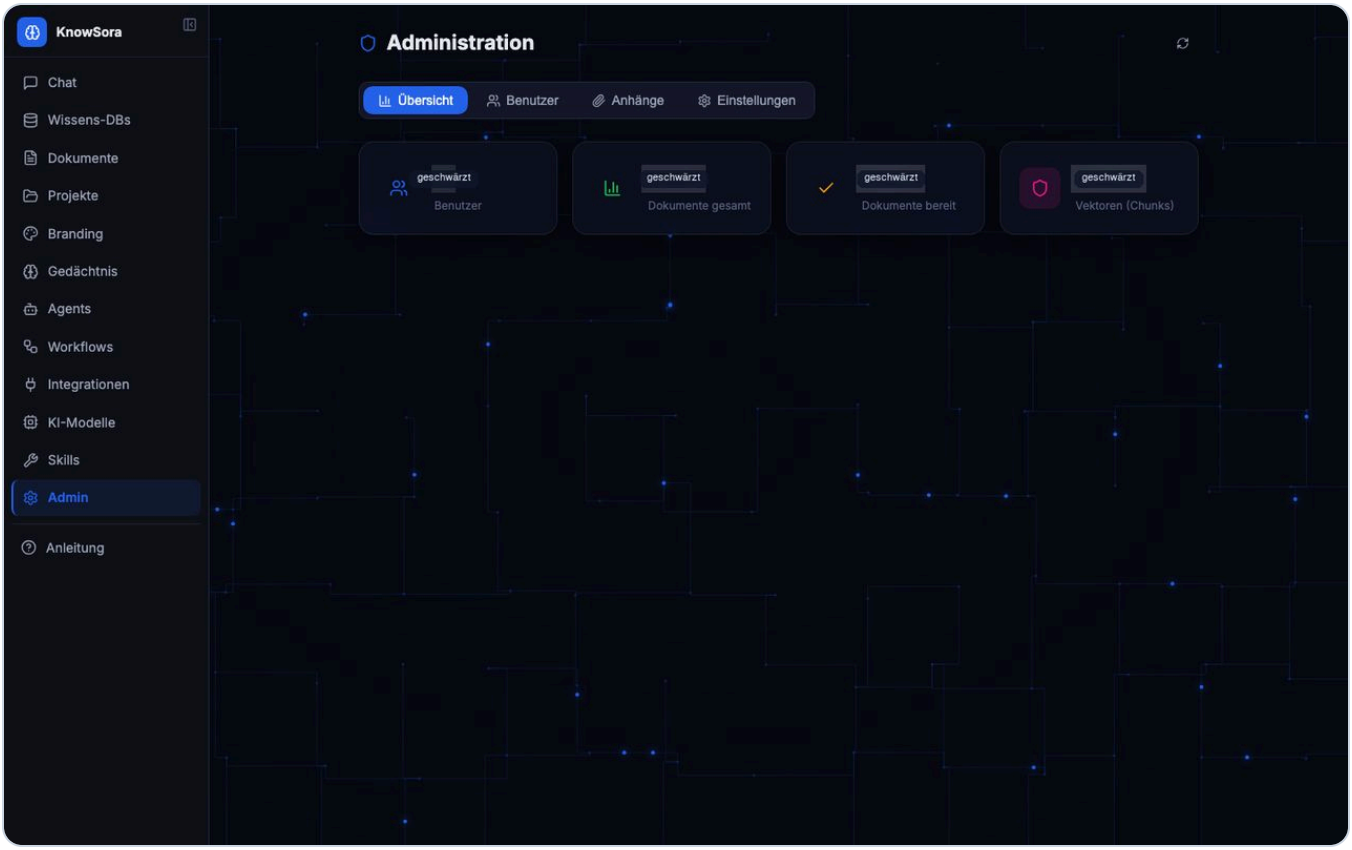
Workflows ohne Freigabe: Ohne Freigabe endet ein Workflow an dem Schritt, der eine Firmen-Integration nutzt, mit der Meldung *Die Firmen-Integration ist für dein Konto nicht freigegeben. Ein Administrator kann sie in der Benutzerverwaltung freigeben.* Das gilt auch für einen Agenten im Workflow, sobald er ein Werkzeug der Firmen-Integrationen aufruft. Danach erzeugt der Lauf keinen KI-Text mehr und verschickt keine Nachricht. Was der Agent vor dem gesperrten Werkzeug schon getan hat, bleibt getan. Ebenso endet ein Agent im Workflow, wenn er `python_exec` aufruft und das Konto keine Shell-Freigabe hat; die Meldung beginnt dann mit *python_exec führt Code auf dem Server aus*. Im Chat erklärt die KI die Meldung im Gespräch.

Kalender: Ohne Freigabe liefern die Kalender-Werkzeuge (`calendar_today` und `calendar_changes`) nur die Termine der eigenen Kalender-Abos. Ist Microsoft 365 eingerichtet, steht im Ergebnis zusätzlich der Hinweis, dass die Firmen-Integration für das Konto nicht freigegeben ist. Wer nur den Kalender „Microsoft 365“ abfragt, bekommt die Meldung statt einer leeren Liste.



Gedächtnis: was sich KnowSora über dich merkt, einsehbar und löschbar.

4.7 Admin & Branding



Admin: Nutzer, Provider-Freigaben, Betriebsmodus (Lokal-Modus), Statistiken.

Nutzer legst du in LocalEye an; hier siehst du ihren Stand und stellst den **Lokal-Modus** ein (blockiert jeden externen Verkehr — nur lokale Modelle). Im Lokal-Modus (Betriebsmodus *Nur lokale KI* oder *Vollständig lokal*) starten Claude Code, Codex und Gemini CLI nicht, auch nicht im Chat oder in der Shell-Sitzung.

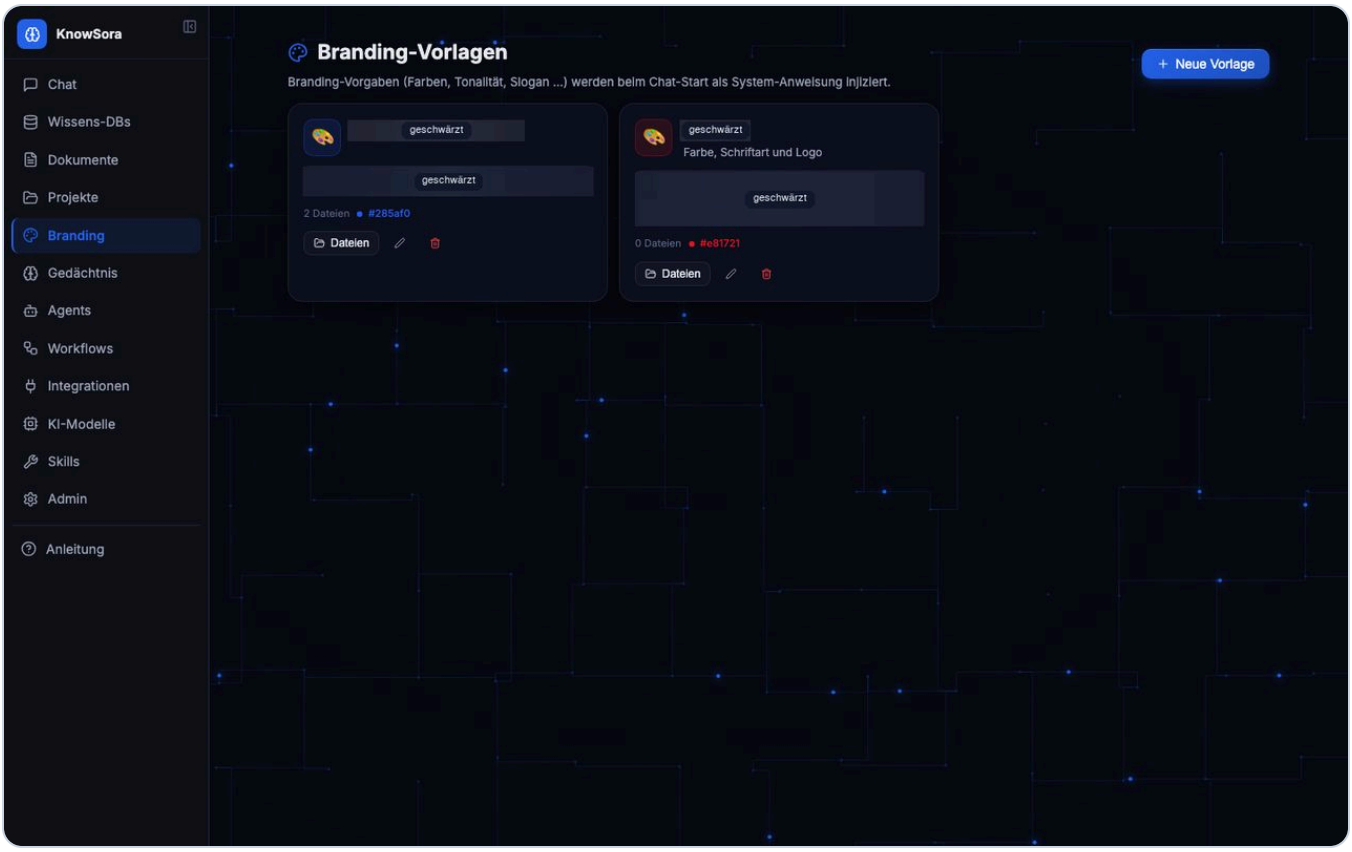
Unter **Admin → Benutzer** öffnet der Knopf **KI-Modelle und Firmen-Integrationen** den Dialog *Zugriff*. Der Schalter **Firmen-Integrationen nutzen** erlaubt einem Konto Postfach, Kalender, Teams und die übrigen Integrationen mit dem firmenweit hinterlegten Zugang. Die Freigabe wirkt sofort beim nächsten Aufruf, auch für vorhandene Skills, Agenten und Workflows. Bei Administrator-Konten gibt es keinen Schalter: Administratoren dürfen die Firmen-Integrationen immer nutzen. Das gilt nicht in einer Sitzung von außerhalb des Verwaltungsnetzes: Dort ist ein Administrator nur Betrachter.

Stillgelegte Konten: Konten mit der Endung „.alt-1“, „.alt-2“ … hat KnowSora stillgelegt, weil ihr Name in LocalEye an ein anderes Konto ging. Ihre Chats, Dokumente und Neuronenkerne bleiben erhalten. Wurde das LocalEye-Konto nur umbenannt, bleibt das Konto aktiv; sein Inhaber findet es bei der nächsten Anmeldung von selbst wieder, der Name zieht nach. Wurde das LocalEye-Konto gelöscht und für dieselbe Person neu angelegt, holt der Admin das alte

Konto zurück: **Admin** → **Benutzer**, am Konto **Wieder zuordnen** wählen und die Rückfrage bestätigen. Das Konto heißt danach wieder wie vorher und ist aktiv. Die Freigabe gilt für das LocalEye-Konto, das beim Zuordnen diesen Namen trägt: Es bekommt das Konto bei seiner nächsten Anmeldung mit allen Daten. Wird dieses LocalEye-Konto vorher gelöscht und der Name neu vergeben, bekommt der Nachfolger das Konto nicht; es wird wieder stillgelegt. Das inzwischen entstandene leere Konto macht Platz: Es wird umbenannt und stillgelegt, seine Daten bleiben erhalten. Ein Irrtum lässt sich auf demselben Weg zurücknehmen (**Wieder zuordnen** am anderen Konto). Umbenennen und Aktivieren von Hand genügt nicht.

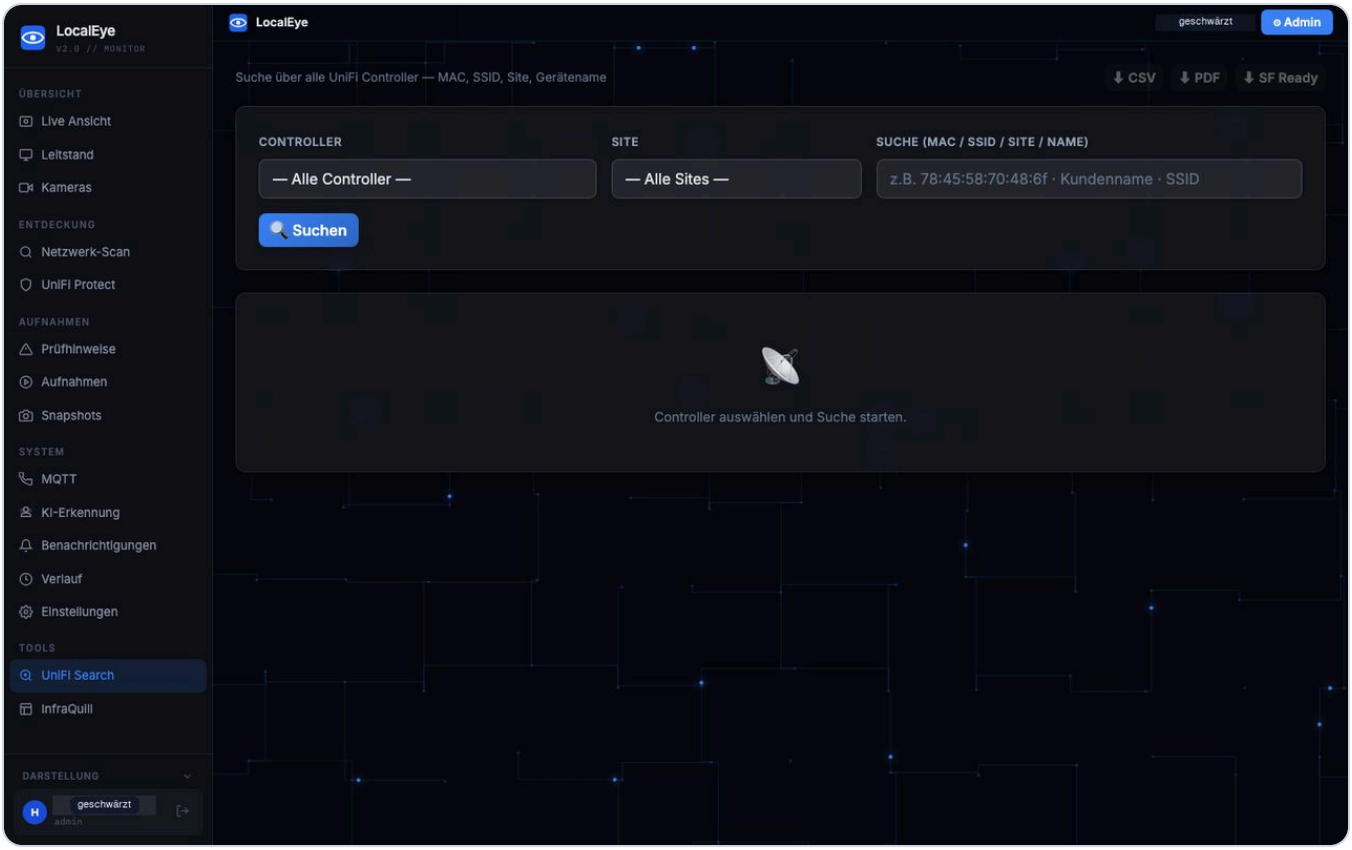
Wieder zuordnen erst wählen, wenn das LocalEye-Konto der Person den alten Namen trägt. KnowSora lehnt mit einer Erklärung ab, wenn LocalEye kein Konto mit diesem Namen führt oder wenn das alte Konto dort weiter jemandem gehört (nur umbenannt).

Ein lokales KnowSora-Konto, das noch an kein LocalEye-Konto gebunden ist, geht nicht an ein neues LocalEye-Konto gleichen Namens, solange das frühere LocalEye-Konto dieses Namens (inzwischen umbenannt) noch besteht. Es bleibt dessen Inhaber erhalten.



Branding (Whitelabel): Name, Logo, Farben, Hintergrund — passend zu LocalEye.

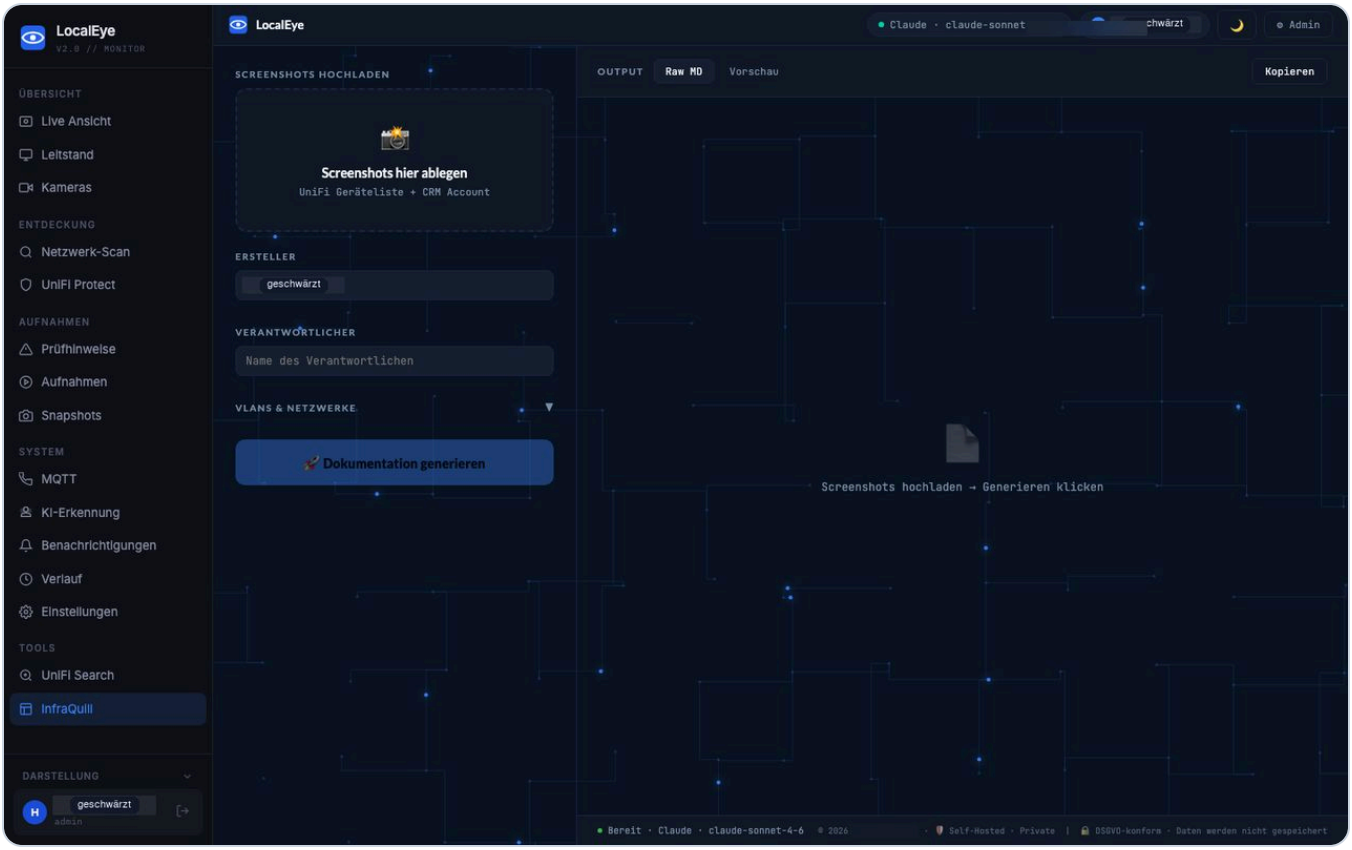
5 SearchTool



SearchTool im Reiter „UniFi Search“: Controller und Site wählen, nach MAC, SSID oder Name suchen; Export als CSV, PDF oder SF-Ready.

1. In LocalEye links **Tools** → **UniFi Search** öffnen (Login übernimmt das SSO).
2. Oben rechts **Admin**: Controller hinterlegen — Adresse, Benutzer, Passwort (nur-Lese-Konto empfohlen).
3. Controller und Site wählen, Suchbegriff eintippen (MAC-Adresse, SSID, Kundenname, Gerätename) und **Suchen**.
4. Treffer als CSV/PDF exportieren.

6 InfraQuill



InfraQuill: Screenshots ablegen, Ersteller/Verantwortlichen eintragen, VLANs ergänzen, Dokumentation generieren — Ausgabe als Markdown mit Vorschau.

1. In LocalEye links **Tools** → **InfraQuill** öffnen.

2. Oben rechts den **Anbieter** wählen: *Lokal – Texterkennung* (ohne KI und ohne Internet, immer verfügbar) oder einen Cloud-Anbieter (Schlüssel unter *Admin*).

3. Unter **Geräte aus SearchTool** Controller und Standort wählen – die Geräte kommen dann exakt aus dem Controller, mit Uplinks und VLANs. Für Controller, die nicht in SearchTool eingebunden sind, Screenshots der UniFi-Geräteliste in das Feld ziehen (beides lässt sich kombinieren).

4. Ersteller, Verantwortlichen und VLANs/Netzwerke eintragen, dann **Dokumentation generieren**.

5 Ergebnis als Markdown kopieren oder in der Vorschau lesen. Es werden keine Daten gespeichert.

Gut zu wissen: Lokal – Texterkennung liest eine UniFi-Geräteliste in wenigen Sekunden (Name, MAC, IP, Modell, Status); unsichere Zeilen markiert InfraQuill mit ⚠. Kundenname und -nummer trägst du dabei in die Felder unter *Kunde* ein. Im *Admin* steht sie als feste Karte; nicht mehr benötigte Anbieter-Karten entfernst du mit 🗑 *Löschen* (zweiter Klick bestätigt). Ein lokales Bildmodell über **Ollama** ist abgeschaltet, weil es auf typischer NAS-Hardware viele Minuten je Screenshot braucht – bei Bedarf mit `COMPOSE_PROFILES=…,infraquill-ollama` in der `.env` einschalten.

7 Anlage verwalten

Die **Anlagen-Oberfläche** unter `http://<IP>:8987/anlage/` ist die Verwaltung der ganzen Installation im Browser — auf Handy, Mac und PC genauso wie auf dem Bildschirm am Gerät. Sie führt zuerst durch die Einrichtung (Lizenz, Admin, Abschluss) und zeigt danach die Verwaltungskarten. Anmeldung mit einem Admin-Konto der Suite. Über den Fernzugriff ist die Seite gesperrt.

Ist die Einrichtung abgeschlossen, öffnet der Knopf **„Weboberfläche öffnen ↗“** oben rechts die Suite in einem neuen Browser-Tab; auf dem Bildschirm am Gerät öffnet er im selben Fenster.

Umgekehrt führt in LocalEye der Eintrag **„Anlage verwalten ↗“** (Seitenleiste, Abschnitt *System*, nur für Admins) hierher – ebenfalls in einem neuen Tab.

Status

Zeigt die Dienste (laufen/gestoppt), belegten Plattenplatz, RAM und die Lizenz. Der Knopf *Systemprüfung* startet die gleiche Prüfung wie `manage.sh system` und zeigt das Ergebnis im Live-Log. Der *Selbsttest* (`manage.sh selbsttest`) spricht jede App an und prüft Texterkennung, Modelle samt Download-Quellen (Hugging Face und Shop-Spiegel) und den freien Platz – ideal direkt nach einem Update. Er hat auch den Abschnitt *Sicherung*: Liegt das Sicherungsziel im Netz und entstehen die Sicherungen unverschlüsselt, warnt er: *Sicherung am Netzziel unverschlüsselt: Sicherungs-Passwort setzen*. Im Datenschutzprofil prüft er zusätzlich Sicherungs-Passwort, Frist und täglichen Fristlauf der Sicherungen sowie den Adressbereich der Verwaltung, nennt die Zahl der verworfenen Kamera-Ereignisse und warnt vor einem `SNAPSHOT_TOKEN` in der `.env`.

Update

Jedes Release ist signiert: Die Anlage prüft Signatur und Dateien, bevor sie etwas einspielt, und lehnt Verändertes ab. **Online-Update:** Liegt im Shop eine neuere Version, zeigt die Karte sie samt Hinweis an. *Jetzt aktualisieren* lädt sie mit dem Lizenzschlüssel (ohne das Download-Kontingent), prüft die Prüfsumme, spielt sie ein und lässt danach den Selbsttest laufen. In LocalEye trägt der Eintrag *Anlage verwalten ↗* dann das Abzeichen *Neu*. Alternativ eine Release-ZIP vom eigenen Gerät hochladen (mit Fortschrittsanzeige) und einspielen. Der Schalter *vollständig neu bauen* baut alle Images ohne Zwischenspeicher — dauert länger, hilft bei hartnäckigen Fehlern.

Lizenz

Lizenzschlüssel eintragen oder wechseln. Die Rückmeldung des Lizenzservers erscheint im Klartext; der Schlüssel selbst wird nirgends protokolliert.

Datenschutzprofil

Schalter für strenge Vorgaben: 72 Stunden Aufbewahrung, Meldungen ohne Bild, kein Ton, nur freigegebene Kanäle, Zusatz-Apps aus. Ein Tipp auf eine Meldung öffnet das Ereignis samt Bild nach der Anmeldung in LocalEye. Beim Einschalten wird alles Ältere endgültig gelöscht, gehaltene Aufnahmen bleiben. E-Mail-Meldungen gehen nur über Mailserver, die du in derselben Karte unter *Erlaubte Mailserver* einträgst und speicherst. Ausnahmen, auch einzelne Zusatz-Apps, gibst du in LocalEye unter *Einstellungen → Funktionen* mit Begründung frei. Die Funktion geht damit gleich an (den Lernlauf schaltet der Admin danach selbst ein), die Freigabe steht im Protokoll. Für Griff-Wache, Prüfhinweise, Person verfolgen, Kassen-Abgleich, Clip im Prüflink, Lernlauf, Texterkennung, Bildbeschreibung und die Zusatz-Apps braucht eine Freigabe zusätzlich eine Grundlage wie DSFA oder Betriebsvereinbarung; Clip im Prüflink und Lernlauf gibt es nur befristet mit Enddatum. Im Profil stoppt der Fernzugriff, solange ein Admin keinen zweiten Faktor hat. Neue Admins legst du deshalb im lokalen Netz an. Alle Admin-Funktionen samt Einstellungen, Benutzern, Meldekanälen und Scharf/Unscharf, auch per API-Token, sind im Profil über den Fernzugriff gesperrt. Aus dem Netz gehen sie nur aus dem Bereich im Feld *Verwaltung erlaubt aus (Adressbereich)* (`VERWALTUNG_LAN_CIDRS`), direkt am Gerät immer. Ohne gültigen Bereich ist die Verwaltung nur am Gerät erreichbar. Aus dem Netz speichert die Karte nur einen Bereich, der die eigene Adresse enthält; leeren geht nur am Gerät. Ton und Trainings-Export sind gesperrt; die Bildbeschreibung und bei KI-Erkennung und Zonen-Alarm ein Vorschaubild in E-Mail und Push, verpixelt wie das Prüfbild, gibt es nur mit Freigabe. Im Profil öffnet kein Token ein Kamerabild (Kapitel

„Schnappschuss-Token“), jede Kamera zählt nur Personen, und Sicherungen entstehen nur verschlüsselt (Karte *Sichern*). Beim Abschalten kommen Fristen, Kanäle und Schalter zurück, Gelöschtes nicht. Im EDEKA-Betrieb (`PRIVACY_LOCK`, Vorlage `.env.edeka.example`) bleibt das Profil aktiv, bis ein Admin in LocalEye unter *Einstellungen → Funktionen*, Karte Datenschutzprofil, eine Abschalt-Freigabe mit Begründung erteilt; vorher verweigert diese Karte das Abschalten. Die Freigabe gilt für ein Abschalten. Mit Profilsperre zeigt die Karte das Profil auch bei geleertem `PRIVACY_PROFILE` als eingeschaltet, und LocalEye, diese Oberfläche und `manage.sh` wenden dieselbe Regel an. `./manage.sh profil edeka` überträgt die Vorlage in die `.env`, `./manage.sh abnahme` prüft das Ergebnis Vorgabe für Vorgabe.

KI-Training

Zwei Schalter, beide standardmäßig aus: *Prüfhinweise verpixelt an den Hersteller senden* und *Gelerntes aus dem Pool automatisch übernehmen* (je einmal pro Woche, Einzelheiten unter LocalEye → Gelerntes). Die Karte zeigt, wie viele Normal- und Pool-Muster die KI kennt, die Neuerungen der letzten 30 Tage und wann zuletzt gesendet und abgerufen wurde.

Sichern

Vollsicherung (Datenbanken, Einstellungen, Projekte, KI-Anmeldungen) mit optionaler Bezeichnung. Unter *Sicherungsziel* wählst du, wohin sie geht: lokal nach `backups/`, in einen Ordner auf dem Gerät (z. B. USB-Festplatte), auf eine SMB-Freigabe (Windows, NAS) oder eine NFS-Freigabe – das entlastet den lokalen Speicher. Freigaben werden nur während einer Sicherung eingehängt; *Speichern und prüfen* testet das Ziel und zeigt den freien Platz. Unter *Sicherungs-Passwort* legst du ein Passwort fest. Damit werden neue Sicherungen verschlüsselt und lassen sich auch auf einem Ersatzgerät prüfen und einspielen. Ohne dieses Passwort ist eine verschlüsselte Sicherung nicht wiederherstellbar, bewahre es also getrennt von der Anlage auf. Wer eine Sicherung entschlüsseln kann, hat alles darin (Konten samt zweitem Faktor, `.env`, Schlüssel des Verbunds): Schütze das Passwort wie den Zugang zur Anlage. Liegt das Ziel im Netz und ist kein Passwort gesetzt, warnt die Karte. Im **Datenschutzprofil** entsteht ohne Sicherungs-Passwort keine Sicherung, auch nicht vor dem Zurückspielen. Dort löscht die Anlage eigene Sicherungen nach `SICHERUNG_FRIST_TAGE` (leer = 7 Tage); den Fristlauf stößt *Anlage verwalten* einmal am Tag selbst an, von Hand mit `sudo ./manage.sh sicherung-frist`. Die Frist rechnet mit der Uhr der Anlage: Springt sie um mehr als die Frist nach vorn, löscht der Fristlauf alle Sicherungen der Anlage. Sicherungen eines Vorgängergeräts am selben Ziel löscht die Frist nicht. Ändern oder Entfernen des Passworts und Änderungen am Sicherungsziel verlangen hier im Profil den Code der Authenticator-App; das erste Setzen geht ohne. Fehlt eine USB-Platte später, bricht die Sicherung ab, statt die lokale Platte zu füllen. Ein eigener Ordner oder eine Freigabe

behält seine Rechte; die Sicherungsdateien selbst sind nur für root lesbar. Darunter die Liste der Sicherungen (am Ziel mit dem Abzeichen *Ziel*). Jede Sicherung trägt ein Anlagenkürzel im Namen und einen Echtheitsnachweis (Datei `.nachweis` daneben). Zurückspielen lassen sich hier nur Sicherungen mit gültigem Nachweis. Sicherungen anderer Anlagen am selben Ziel zeigt die Karte nicht und löscht sie nie. Einzelheiten stehen unten unter „Sicherungen im Einzelnen“.

Zurückspielen

Eine Sicherung auswählen und zurückspielen. Der Schalter *mit Einstellungen* nimmt auch `.env` und Zugangsdaten mit. `docker-compose.yml` und `nginx.conf` bleiben auf dem installierten Stand; die gesicherte Fassung liegt als `*.aus-sicherung-<zeit>` daneben. Vor dem Start erscheint ein Bestätigungsdialog mit Klartext-Warnung — der aktuelle Stand wird dabei überschrieben. Im **Datenschutzprofil** bleibt die `.env` der Anlage auch *mit Einstellungen* stehen; das Protokoll nennt nur die Namen der Werte, die die Sicherung anders hat. Konfiguration und Konten des Push-Servers (ntfy) kommen auch im Profil aus der Sicherung, deshalb keine Sicherung aus fremder Hand zurückspielen. Stammt die Sicherung von dieser Anlage, wird ihr Journal weitergeführt; auf einem Ersatzgerät kommt das Journal der Sicherung zurück, und die Einträge des Ersatzgeräts hängen dahinter; ist die Lage unklar, wird nicht zurückgespielt. Ereignisbilder und Berichte der Anlage bleiben, und eine Abschalt-Freigabe kommt nicht aus der Sicherung.

Betrieb

Alle Container starten, stoppen oder neu starten (Stopp und Neustart mit Bestätigung, weil währenddessen auch Kameras und Apps nicht erreichbar sind). Dazu das Protokoll jedes Dienstes (letzte 300 Zeilen) zum Nachlesen.

GPU

Nur sichtbar, wenn eine NVIDIA-Karte erkannt ist: Erkennung anzeigen, Automatik einschalten oder auf CPU zurückschalten (`manage.sh gpu`).

Ports

Liste aller vergebenen Ports. Ein Port lässt sich einzeln ändern — mit Warnung, denn Lesezeichen, Firewall-Regeln und Reverse-Proxy müssen danach angepasst werden; ein Neustart folgt automatisch.

Notzugang

Nur im Verbund als Slave: die Not-Oberfläche ein- oder ausschalten, falls der Master nicht erreichbar ist.

Sicherungen im Einzelnen

- **Sicherungsziel:** *Speichern und prüfen* hängt die Freigabe immer selbst ein. Läuft gerade eine Sicherung auf dieselbe Freigabe, wartet die Prüfung bis zu einer Minute und meldet dann, dass das Sicherungsziel belegt ist. Ebenso wartet eine zweite Sicherung bis zu einer Minute und bricht sonst mit dieser Meldung ab: *Das Sicherungsziel ist gerade von einem laufenden Auftrag belegt – bitte später erneut versuchen.* Wer im Zielordner schreiben darf, kann Sicherungen dort löschen; ein Austausch fällt beim Einspielen am Echtheitsnachweis auf.
- **Arbeitsordner:** Während eine Sicherung läuft, liegt sie in einem Arbeitsordner `.allinone-sicherung-…` im Ziel und erscheint erst fertig in der Liste. Bleibt nach einem Abbruch ein solcher Ordner liegen, entfernt ihn die nächste Sicherung nach 24 Stunden.
- **Sicherungen von vor dem Update:** Sicherungen von vor dem Update im Ordner `backups/` bekommen ihren Echtheitsnachweis einmalig bei der ersten Sicherung oder beim ersten Zurückspielen nach dem Update. Bis dahin tragen sie in der Karte das Kennzeichen *Nachweis folgt*. Später in den Ordner kopierte Sicherungen ohne Nachweis (Kennzeichen *Ohne Nachweis*) lassen sich nur am Terminal nach Bestätigung einspielen: `sudo ./manage.sh restore <datei>`.
- **Echtheitsnachweis:** Der Echtheitsnachweis gilt für genau den Dateinamen der Sicherung: Eine umbenannte Sicherung gilt nicht mehr als echt. Sicherungen einer anderen Anlage lassen sich nur am Terminal einspielen und brauchen dort die getippte Bestätigung `ANDERE ANLAGE`.
- **Verschlüsselung:** Verschlüsselt wird nur, wenn auf dem Gerät openssl ab Version 1.1.1 vorhanden ist. Fehlt es oder ist es älter, entstehen Sicherungen ohne Datenschutzprofil

trotz Sicherungs-Passwort unverschlüsselt; die Karte Sichern und der Selbsttest weisen darauf hin. Im Datenschutzprofil entsteht dann keine Sicherung.

- **Altes Namensformat:** Sicherungen im alten Namensformat (Kennzeichen *Altes Format*) an einem Sicherungsziel außerhalb (Ordner, Freigabe) räumt die Anlage nach dem Update nicht mehr automatisch auf, weil sie dort nicht sicher zu ihr gehören. Die Karte Sichern nennt ihre Anzahl und Gesamtgröße; bei knappem Platz bitte von Hand am Ziel löschen.
- **Kopierte Installation:** Wird ein Installordner kopiert und mit anderem `INSTANCE_SUFFIX` betrieben, gilt die Kopie als eigene Anlage: Bei ihrer ersten Sicherung bildet sie ein eigenes Kürzel und einen eigenen Geräteschlüssel. Die mitkopierten Sicherungen der Quelle zählen dort als Sicherungen einer anderen Anlage. Dasselbe geschieht, wenn das `INSTANCE_SUFFIX` einer Anlage nachträglich geändert wird.

Aufträge und Live-Log

Jede Karte startet einen **Auftrag**, dessen Ausgabe live auf der Seite mitläuft (Zustand *läuft / fertig / fehler / abgebrochen*). Aufträge laufen losgelöst auf dem Gerät weiter — auch das Update, das die Anlagen-Oberfläche selbst neu startet, läuft durch; die Seite meldet dann kurz „Verbindung wird wiederhergestellt …“ und findet den Auftrag danach wieder. Es läuft immer nur ein Auftrag gleichzeitig; die letzten 30 bleiben mit Protokoll erhalten. Öffnest du ein älteres Protokoll aus *Letzte Aufträge* oder das Dienst-Log, während ein Auftrag läuft, bleibt es stehen; der laufende Auftrag steht als Zeile darüber (*Im Hintergrund läuft: …*, am Ende mit dem Ergebnis) mit dem Knopf **Verlauf zeigen**.

Einrichtungscode und Anmeldung

1. Solange noch **kein Admin** existiert, verlangt die Seite aus dem Netz einen achtstelligen **Einrichtungscode** (steht in der Ausgabe von `./manage.sh update` und `./manage.sh config`, beim Gerät mit Bildschirm auch auf der Konsole). Am Gerät selbst ist kein Code nötig.

2. Nach dem Anlegen des Admins gilt nur noch die **Anmeldung** mit Benutzername und Passwort eines Admin-Kontos — dieselben Daten wie in LocalEye.

3. Fünf Fehlversuche (Code oder Passwort) sperren die Adresse für zehn Minuten.

Gut zu wissen: Hat das Admin-Konto in LocalEye die Zwei-Faktor-Anmeldung eingerichtet, fragt auch die Anlagen-Seite den Code ab. Ein Code aus der Authenticator-App gilt nur einmal, auch zwischen LocalEye und „Anlage verwalten“. Wer sich gerade in LocalEye angemeldet hat, nimmt für „Anlage verwalten“ den nächsten Code (höchstens 30 Sekunden warten). Sonst erscheint: „Dieser Code wurde schon verwendet. Bitte den nächsten Code abwarten.“ Für die erste Anmeldung nach dem Einrichten gilt der nächste Code der App, nicht der eben zur Bestätigung eingegebene. Die Seite ist nur im lokalen Netz erreichbar.

Gut zu wissen: Adresse und (bei Bedarf neuen) Einrichtungscode zeigt `./manage.sh config`; der Befehl startet die Anlagen-Oberfläche auch auf älteren Installationen nach. Ist LocalEye einmal nicht erreichbar (Einrichtung, laufendes Update), leitet die Startseite von selbst auf `/anlage/` weiter.

8 Bildschirm am Gerät

Hängt ein Monitor am Gerät, kann die Suite als **Vollbild-Browser** direkt dort laufen — Einrichtung, Kameras und Verwaltung mit Tastatur und Maus, genau wie aus dem Netz, nur ohne Einrichtungscode. Die Appliance-ISO schaltet das beim ersten Start von selbst ein, sobald ein Monitor erkannt wird.

```
cd /opt/allinone
sudo ./manage.sh webgui on       # Vollbild-Browser auf dem Monitor einsch
sudo ./manage.sh webgui off      # zurück zur Textkonsole
./manage.sh webgui status        # an/aus, Browser, Adresse, Monitor erkan
```

1. `webgui on` installiert den schlanken Fenstermanager `cage` und einen Browser (Chromium oder Epiphany), legt den Systembenutzer `allinone-kiosk` an und meldet ihn auf der ersten Konsole automatisch an.

2. Der Browser startet im Kiosk-Modus mit der Suite-Adresse des Geräts; stürzt er ab, startet er nach drei Sekunden neu.

3. `webgui off` nimmt die Automatik zurück; die Pakete bleiben installiert, ein erneutes `on` geht schnell.

Gut zu wissen: Auf einem NAS (kein systemd, kein apt-get), ohne root-Rechte oder ohne erkannten Monitor installiert `webgui on` nichts: Es nennt den Grund, gibt die Adresse für andere Geräte aus und endet ohne Fehler. Die Suite läuft dann wie gewohnt weiter — bedienbar von jedem Browser im Netz.

9 Oberfläche aus der Shell (ohne Monitor)

Kein Monitor am Gerät, etwa beim NAS? Dann öffnet `./manage.sh webgui` in einer **SSH-Sitzung** die Oberfläche **direkt im Terminal** — mit Maus und Tastatur, ohne Einrichtungscode, weil die Seite „am Gerät“ geöffnet wird.

```
./manage.sh webgui                # per SSH: Oberfläche im Terminal
./manage.sh webgui terminal       # Terminal-Browser erzwingen
./manage.sh webgui tunnel         # nur den SSH-Tunnel-Befehl ausgeben
```

1. Klicken und Mausrad funktionieren wie gewohnt; **Tab** oder **Pfeil runter** springt zum nächsten Feld, **Pfeil hoch** bzw. **Esc + Tab** zurück, **Enter** führt aus, **Leertaste** schaltet Kästchen, **Strg+C** beendet.

2. Über dem Terminal-Browser steht jedes Mal der fertige Tunnel-Befehl (`ssh -L 8987:127.0.0.1:8987 …`). Damit öffnet der Browser des eigenen Geräts unter `http://localhost:8987/` die vollständige Oberfläche mit Livebildern — Terminus/Termius speichert das als „Port Forwarding“.

3. Der Terminal-Browser ist eingezäunt: nur Adressen dieses Geräts, schreibgeschützt, ohne Zusatzrechte, höchstens 1 GB RAM und niedrige CPU-Priorität. Bricht die SSH-Verbindung ab, wird er automatisch beendet.

Gut zu wissen: Umlaute, €, ß und alle Sonderzeichen kommen im Terminal richtig an. Schickt das Terminal-Programm die ⌥-Taste als „Meta“ (Voreinstellung in Termius), übersetzt die Oberfläche ⌥L, ⌥E, ⌥6–9, ⌥N und ⌥⇧7 nach deutscher Mac-Tastatur in @, €,], |, {, }, ~ und \ und blendet einen Hinweis ein; sauberer ist es, in Termius unter *Einstellungen → Terminal* „Option als Meta-Taste“ auszuschalten.

Gut zu wissen: Im Terminal zeigt die Oberfläche bewusst eine ruhige Darstellung: kein bewegter Hintergrund, keine Schatten, rechteckige Knöpfe, Menü ohne Symbole — so bleibt der Text scharf und der Terminal-Browser braucht kaum Rechenzeit. Kamerabilder sind im Terminal nur grob zu erkennen — für den Kamerablick den Tunnel oder einen normalen Browser im Netz nutzen. Das Terminal sollte 256 Farben und Mausmeldungen können (Terminus/Termius, iTerm, Windows Terminal).

10 Betrieb: Update, Backup, Restore

Alles Wichtige steht im Browser unter `/anlage/` (Kapitel „Anlage verwalten“) — und weiterhin per `manage.sh` im Installationsordner:

BEFEHL	WAS ER TUT
`./manage.sh update`	Neue Release-ZIP einspielen: Code aktualisieren, Konfiguration migrieren, Images bauen, Container rollierend neu starten. Deine Daten bleiben unangetastet. In der `.env` ergänzt das Update nur eigene Verwaltungswerte.
`./manage.sh watch`	Wartet auf neue ZIPs im Ordner und aktualisiert automatisch.
`./manage.sh backup`	Vollsicherung nach `backups/`: Datenbank (konsistenter Snapshot), Konfiguration, Aufnahmen-Metadaten, Projekte, KI-Anmeldungen. Behält 15 eigene Archive (`BACKUP_ZIEL_BEHALTEN`). Archive anderer Anlagen am selben Ziel bleiben liegen. Neben jedem Archiv liegt ein Echtheitsnachweis (`.nachweis`); mit Sicherungs-Passwort ist das Archiv verschlüsselt (`.tar.gz.enc`).
`./manage.sh restore <archiv>`	Sicherung zurückspielen (fragt nach; vorher wird der aktuelle Stand gesichert). Mit `--with-config` auch `.env` und Zugangsdaten; im Datenschutzprofil bleibt die `.env` der Anlage dabei stehen. `docker-compose.yml` und `nginx.conf` bleiben auf dem installierten Stand; die gesicherte Fassung liegt als `*.aus-sicherung-<zeit>` daneben. Vorher wird der Echtheitsnachweis geprüft. Fehlt er oder passt er nicht (z. B. Sicherung von vor dem Update), geht das Einspielen nur hier am Terminal nach Eingabe von `UNGEPRUEFT`, mit `--with-config` zusätzlich `KONFIGURATION`. Fehlt einer verschlüsselten Sicherung der Echtheitsnachweis, fragt das Zurückspielen am Terminal nach ihrem Sicherungs-Passwort (Eingabetaste nimmt das gespeicherte) und prüft es sofort. Nach drei falschen Versuchen endet es, ohne etwas zu verändern.

`./manage.sh sicherung-passwort [entfernen]`	Sicherungs-Passwort setzen, ändern oder entfernen. Ein neues Passwort gilt für neue Sicherungen; ältere brauchen ihr bisheriges. Im Datenschutzprofil Pflicht: ohne Passwort keine Sicherung.
`sudo ./manage.sh sicherung-frist`	Nur im Datenschutzprofil: eigene Sicherungen älter als `SICHERUNG_FRIST_TAGE` (leer = 7 Tage) löschen, mit Eintrag im Protokoll. Läuft auch von selbst mit jeder neuen Sicherung und einmal am Tag über *Anlage verwalten*.
`./manage.sh ki-sandbox [absichern|zurueck]`	`./manage.sh ki-sandbox` zeigt, wie die Docker-Sandbox der KI läuft. `./manage.sh ki-sandbox absichern` stellt sie auf einen Docker-Daemon ohne Root um: Eine KI-Sitzung kann dann über die Sandbox nicht mehr Root auf dem Gerät werden. Images, Container und Volumes, die KI-Sitzungen bisher in der Sandbox angelegt haben, bleiben im alten Speicher; `./manage.sh ki-sandbox zurueck` holt sie zurück. Neue Installationen wählen die abgesicherte Art beim ersten Start selbst, wenn das Gerät sie unterstützt. Die Offline-ISO bringt die KI-Sandbox in beiden Arten mit. Beim ersten Start erprobt die Anlage die abgesicherte Art ohne Internet und wählt sie, wenn das Gerät sie unterstützt. `./manage.sh ki-sandbox absichern` und `zurueck` brauchen auf einer solchen Anlage kein Internet. Der Docker-Daemon der KI-Sandbox ist nur über den Socket erreichbar, den KnowSora nutzt, nicht über das Netz. Der Selbsttest prüft das.
`./manage.sh start | stop | restart | status | logs`	Container steuern und beobachten.
`./manage.sh activate <schlüssel>`	Lizenzschlüssel setzen/ändern.
`./manage.sh port`	Ports neu zuweisen, falls sich am System etwas geändert hat.
`./manage.sh config`	Anlagen-Oberfläche starten/nachrüsten, Adresse und (ohne Admin) einen neuen Einrichtungscode ausgeben.
`./manage.sh webgui`	Per SSH: Oberfläche direkt im Terminal (Maus + Tastatur), darüber der fertige Tunnel-Befehl; am Gerät: Vollbild-Kiosk.

`./manage.sh webgui terminal|tunnel`	Terminal-Browser erzwingen bzw. nur den SSH-Tunnel-Befehl ausgeben.
`sudo ./manage.sh webgui on|off|status`	Vollbild-Browser am Gerätemonitor ein-/ausschalten (Kapitel „Bildschirm am Gerät“).
`./manage.sh profil <name> [--ja]`	Voreinstellung aus `.env.<name>.example` übernehmen, etwa `edeka`: zeigt vorher, was sich ändert, und fragt nach; `--ja` ohne Rückfrage. Ports, Lizenz und Mailserver bleiben. Ist `VERWALTUNG_LAN_CIDRS` leer oder ungültig, fragt der Befehl am Terminal danach und schlägt das Netz der eigenen Schnittstelle vor; leer heißt im Profil: nur am Gerät verwaltbar.
`./manage.sh abnahme`	Abnahmeprotokoll für das Datenschutzprofil: jede Vorgabe mit OK, ABWEICHUNG oder NICHT PRÜFBAR, Ablage unter `logs/`, Exit-Code 1 bei Abweichungen. Ändert nichts. Neu sind unter anderem die Zeilen „Bilder über Token“, „Schnappschuss-Token-Datei“, „Lernlauf“, „Clip im Prüflink“, „Aktive Freigaben“, „Leitstellen-Zugang“, „Version im laufenden Image“, „Sicherungs-Passwort gesetzt“, „Sicherungen älter als Frist“ und „Fristlauf“. Läuft noch ein älteres Image mit Kennzeichenerkennung, steht dort eine ABWEICHUNG, bis `./manage.sh update` neu gebaut hat.

1. **Update:** neue ZIP in den Installationsordner legen (auch „(1)“-Kopien werden erkannt), `./manage.sh update` ausführen, Ausgabe lesen. Die KI-Kommandozeilen aktualisieren sich unabhängig davon täglich selbst.

2. **Backup:** `./manage.sh backup` (als root/sudo). Das Archiv liegt unter `backups/allinone_backup_<kürzel>_<datum>_<bezeichnung>.tar.gz`, daneben der Echtheitsnachweis `.nachweis`. Für ein anderes System beide Dateien mitnehmen und dort dasselbe Sicherungs-Passwort setzen.

3. **Restore:** `./manage.sh restore backups/<archiv>`, mit `JA` bestätigen. Bei einer Sicherung ohne gültigen Nachweis zusätzlich `UNGEPRUEFT` eingeben.

4 **Ersatzgerät:** Nach der Neuinstallation dasselbe Sicherungsziel eintragen. Die Sicherungen der ausgefallenen Anlage zeigt `sudo ./manage.sh restore ziel:` an; die Karte Zurückspielen zeigt sie nicht. Eingespielt wird mit `sudo ./manage.sh restore ziel:<datei>`, dem Sicherungs-Passwort und der Bestätigung `ANDERE ANLAGE`. Wurde ohne Sicherungs-Passwort gesichert, kann das Ersatzgerät den Echtheitsnachweis nicht prüfen; dann verlangt das Einspielen zusätzlich `UNGEPRUEFT`.

Achtung: Aufnahmen (Videos) sind bewusst *nicht* im Backup — sie sind groß und liegen in deinem Aufnahme-Ordner. Sichere diesen Ordner separat, wenn du ihn brauchst.

11 Fehlerbehebung

SYMPTOM	URSACHE / LÖSUNG
Login klappt, danach wieder die Anmeldemaske	Cookie-Einstellung: Zugriff über `http://IP` mit `SESSION_COOKIE_SECURE=true`. In der `.env` `SESSION_COOKIE_SECURE=auto` setzen (Standard bei Neuinstallationen) und `./manage.sh restart`. Oder per HTTPS (Reverse Proxy) zugreifen.
Kachel zeigt „⚠ Bild von …“	Das Snapshot-Bild ist älter als zwei Minuten – die Kamera liefert gerade nichts. Strom und Netz der Kamera prüfen und in der Anlage den *Selbsttest* starten. Nach Bildschirmsperre oder Tab-Wechsel lädt LocalEye die Kacheln selbst nach (seit 06.10.2026).
Live-Bild bleibt schwarz oder lädt endlos, Snapshot geht	Meist der Codec: Haupt- und Unterstream an der Kamera auf **H.264** stellen, „Smart Codec“ aus. Danach die Kamera in LocalEye einmal öffnen und speichern – das baut den Live-Dienst neu auf. Fehlen im Unterstream die Zugangsdaten, übernimmt LocalEye sie dabei aus dem Hauptstream. Hilft das nicht: `./manage.sh update` zieht die nginx-Konfiguration nach; bei RTSP-Fehlern den Kamera-Test in der Kameraverwaltung nutzen.
Scan-Import meldet „ONVIF fehlgeschlagen“	Die Meldung nennt den Grund je Port. *Anmeldung abgelehnt*: ONVIF-Benutzer und -Passwort an der Kamera prüfen (Herstellerhinweise in 3.2). *verweigert*: auf diesem Port kein ONVIF – unkritisch, solange ein anderer Port antwortet. *HTTPS: …*: TLS-Problem der Kamera; alte TLS-Versionen und selbstsignierte Zertifikate akzeptiert LocalEye bereits. Mit *Manuell per RTSP fortfahren* läuft das Videobild auch ohne ONVIF.
Live-Bild meldet „401 Unauthorized“	Zugangsdaten fehlen oder stimmen nicht. Bei Eingabe von Hand Sonderzeichen im Passwort kodieren: `@` → `%40`, `#` → `%23`, `:` → `%3A`, `/` → `%2F`, `?` → `%3F`.

Ursache bei ONVIF unklar	Prüfskript auf dem LocalEye-Rechner: onvif-test.py herunterladen und `python3 onvif-test.py <Kamera-IP>` starten. Es zeigt je Port, ob ONVIF antwortet, umleitet, die Anmeldung ablehnt oder die Kamerauhr abweicht.
Container startet nicht, „SECRET_KEY“/„readonly database“	Datenordner gehört dem falschen Benutzer. `sudo chown -R $(id -u):$(id -g) knowsora/data`, dann `./manage.sh restart`.
Port belegt	`./manage.sh port` sucht freie Ports; die Adresse steht danach in der Ausgabe und in der `.env`.
KI-CLI meldet „not logged in“	In KnowSora unter *AI-Provider* beim Eintrag **Login** ausführen, ggf. vorher **Reparieren**.
Zusatz-App zeigt statt ihrer Oberfläche eine Anmeldeseite („Bitte über LocalEye anmelden …“, in KnowSora „Die Anmeldung erfolgt ausschließlich über LocalEye.“)	LocalEye neu laden, ein Klick auf die Kachel genügt nicht. Bleibt es dabei: in LocalEye ab- und wieder anmelden, das erneuert die Bindung an den Browser. Wurde dieselbe Anlage im selben Browser über https und über http benutzt, die Cookies der Adresse im Browser löschen. Ein eigener Reverse-Proxy davor muss Cookies unverändert durchreichen. Hilft das nicht, stellt die Zeile `SSO_BINDUNG=aus` in der `.env` das frühere Verhalten wieder her (greift ohne Neustart, sonst `./manage.sh restart`).
KnowSora ordnet nach einem Update Konten falsch zu	Im Installationsordner die leere Datei `knowsora/data/konto_kennung_aus` anlegen: `touch knowsora/data/konto_kennung_aus`. KnowSora ordnet Konten dann wieder nur über den Namen zu, ohne Neustart. Löschen der Datei schaltet die feste Kennung wieder ein. Der Rückschalter wirkt nur vorbeugend: Er muss liegen, bevor sich das betroffene Konto das erste Mal anmeldet. Ein schon stillgelegtes Konto (Endung „.alt-1“) holt er nicht zurück. Dafür gibt es **Wieder zuordnen** unter *Admin → Benutzer*.

Stillgelegtes KnowSora-Konto zurückholen, ohne Oberfläche	`docker exec knowsora-suite-backend python3 db_admin.py le-freigeben kasse.alt-1 --platz-machen`. Ohne `--platz-machen` bricht der Befehl ab, wenn der Name inzwischen einem neuen Konto gehört. Das Notfall-Token (`docker exec knowsora-suite-backend python3 db_admin.py admin-token`) gehört dem ersten aktiven Admin.
KnowSora meldet beim Speichern eines API-Keys „Schlüssel und Adresse gehören zusammen. Bitte die Seite neu laden und erneut speichern.“	Die offene Seite stammt noch aus der Zeit vor dem Update. Seite neu laden, den API-Key noch einmal eintragen und speichern.
KnowSora-Workflow steht im Verlauf als fehlgeschlagen mit „Die Firmen-Integration ist für dein Konto nicht freigegeben“	Ein Schritt oder ein Agent des Workflows nutzt Postfach, Kalender, Teams, SharePoint, Jira, Confluence oder Telegram, und das Konto hat keine Freigabe. Der Administrator gibt das Konto unter *Admin → Benutzer* mit dem Knopf **KI-Modelle und Firmen-Integrationen** frei. Sonst das Werkzeug aus dem Agenten oder dem Workflow nehmen oder den Workflow abschalten. Steht dort stattdessen „python_exec führt Code auf dem Server aus …“, nutzt ein Schritt oder ein Agent des Workflows `python_exec`, und dem Konto fehlt die Shell-Freigabe: Shell-Freigabe in der Benutzerverwaltung von LocalEye erteilen (gilt ab der nächsten Anmeldung) oder `python_exec` aus dem Agenten oder dem Workflow nehmen.
Selbsttest meldet „KI-Sandbox laeuft privilegiert“	Bestehende Anlage, die Sandbox läuft noch mit einem Docker-Daemon als root. Beheben mit `./manage.sh ki-sandbox absichern`. Steht dabei „Image der abgesicherten Art liegt nicht vor und die Image-Quelle ist nicht erreichbar“, hat das Gerät kein Internet und das Image fehlt: mit Internet `./manage.sh ki-sandbox absichern` ausführen.

Selbsttest meldet „Docker-API der KI-Sandbox antwortet im Netz des Hosts“	Der Docker-Daemon der KI-Sandbox lauscht zusätzlich auf einem Netz-Port (2375 oder 2376). Das Update stellt den Dienst `knowsora-ai-dind` in der `docker-compose.yml` um. Findet es dort die erwartete Startzeile nicht, weil die Datei von Hand geändert wurde, bleibt der Port offen. Das Update erneut einspielen. Bleibt die Meldung, den Dienst `knowsora-ai-dind` in der `docker-compose.yml` mit der `docker-compose.yml.example` vergleichen, die Zeile mit `dockerd-entrypoint.sh` von dort übernehmen und `./manage.sh restart` ausführen. Meldet der Selbsttest stattdessen als Warnung „nicht pruefbar, ob sie im Netz des Hosts erreichbar ist“, hat der Container der KI-Sandbox keine eigene Adresse oder auf dem Gerät fehlt `python3`; dann dieselbe Zeile von Hand mit der Vorlage vergleichen.
Selbsttest meldet „Netzregeln nicht verfügbar“	Der Kernel kann Netzregeln für den Sandkasten nicht setzen (vor 6.7). KI-Läufe erreichen die Dienste der Plattform weiterhin.
Selbsttest meldet „Claude-Haken der verwalteten Vorlage rufen Dateien auf, die jede Shell-Sitzung in KnowSora ändern kann“	Das genannte Skript an einen Ort legen, den KI-Läufe nicht beschreiben können, und den Pfad in der Vorlage ändern, oder den Haken entfernen.
Selbsttest meldet „Schluessel-Vermittler der KI-Kommandozeilen laeuft nicht“	Der Schlüssel-Vermittler im KnowSora-Backend läuft nicht; den Grund nennt die Meldung in Klammern. Läufe von Claude Code, Codex und Gemini CLI mit API-Key starten dann nicht, Läufe mit Abo-Anmeldung gehen weiter. Das KnowSora-Backend neu starten (`./manage.sh restart`). Den Zustand zeigt auch der Kasten *Version der KI-CLIs* unter *KI-Modelle → AI-Provider*.

Selbsttest meldet „Zielbindung der KI-Kommandozeilen unvollstaendig“	Was fehlt, steht in Klammern und gelb im Kasten *Version der KI-CLIs*. Steht dort *Gemini CLI startet nicht*, fehlen dessen Einstellungen im Abbild des KnowSora-Backends (sie entstehen beim Bau des Abbilds): `./manage.sh update` baut es neu; bis dahin startet Gemini CLI nicht. Nennt die Meldung Claude Code, das Backend neu starten: `./manage.sh restart`. Nennt sie Codex und `KS_SANDKASTEN_TMP=gemeinsam`, laden Codex-Chat-Läufe wegen dieses Rückschalters die .env des Login-Ordners wieder.
Chat mit Gemini CLI meldet „Gemini CLI startet nicht: …“	Die Einstellungen der Zielbindung für Gemini fehlen im Abbild des KnowSora-Backends oder passen nicht zum Stand. Das Update erneut einspielen (`./manage.sh update` baut das Abbild neu).
Chat mit Claude Code, Codex oder Gemini CLI meldet „Der hinterlegte API-Key wurde vom Hersteller abgelehnt“	Der Hersteller nimmt den API-Key dieses Eintrags nicht an. Unter *KI-Modelle → AI-Provider* den Schlüssel prüfen und gegebenenfalls neu eintragen. Vor dem Update stand an dieser Stelle „Anmeldung fehlt, neu einloggen“. Meldet der Lauf stattdessen „Der API-Key dieses Eintrags wurde inzwischen geändert oder entfernt“, den Lauf neu starten.
Eigener Proxy oder eigene Adresse einer KI-Kommandozeile wirkt nach dem Update nicht mehr	Die Zielbindung lässt Adresse, Proxy und Zertifikate aus Einstellungsdateien im Projekt- und Login-Ordner nicht mehr gelten. Den Wert in die Umgebung des KnowSora-Backends eintragen: Dienst `knowsora-backend` in der `docker-compose.yml` bzw. `docker-compose.override.yml`, mit `HTTPS_PROXY`, `NO_PROXY`, `NODE_EXTRA_CA_CERTS` und für ein eigenes Gateway `ANTHROPIC_BASE_URL` (Claude Code) bzw. `GOOGLE_GEMINI_BASE_URL` (Gemini CLI). Ein Zertifikat muss im Container unter `/etc` liegen. Danach `./manage.sh restart`.
Menüpunkt „Kennzeichen“ und die Option „Kennzeichen lesen“ fehlen nach dem Update	Gewollt: Die Kennzeichenerkennung ist entfernt (Kapitel „Schnappschuss-Token“). Der erste Start nach dem Update hat die gespeicherten Kennzeichen-Daten gelöscht.

Meldung „Kein Adressbereich für die Verwaltung festgelegt. Bitte VERWALTUNG_LAN_CIDRS eintragen.“	Im Datenschutzprofil ist kein gültiger Adressbereich eingetragen, die Verwaltung geht dann nur am Gerät. Am Bildschirm der Anlage arbeiten oder den Bereich eintragen: `./manage.sh profil edeka` fragt am Terminal danach, sonst `VERWALTUNG_LAN_CIDRS` in der `.env` setzen.
Meldung „Bindung der Leitstelle gelöst, bitte neu zuordnen“	Ein SIA-Kanal, an den ein Leitstellen-Konto gebunden war, wurde geändert oder gelöscht. Unter **Einstellungen** → **Benutzer** das Konto der Leitstelle wieder an den Kanal binden.
Etwas anderes	`./manage.sh logs` (alle Container) und `./diagnose.sh` liefern die Details für den Support.

Gut zu wissen: Rückschalter Sandkasten: Alle Rückschalter des Sandkastens stehen in einer Datei: `knowsora/data/sandkasten-schalter.env`, eine Zeile je Schalter, danach das KnowSora-Backend neu starten. `KS_SANDBOX=off` (ganzer Sandkasten aus), `KS_SANDBOX_NETZ=off` (Netzregeln aus), `KS_SANDBOX_NETZ_FILTER=einfach` (Filter nur gegen die direkten Aufrufe, io_uring bleibt frei) oder `KS_SANDBOX_NETZ_FILTER=off` (kein Filter gegen die Umwege, die Netzregeln bleiben), `KS_SANDKASTEN_TMP=gemeinsam` (gemeinsames /tmp wie früher), `KS_CLI_SCHUTZ=aus` (Einstellungen der CLI-Ablage wieder laden), `KS_CLI_MITSCHRIFT=an` (Chat-Läufe speichern wieder Mitschriften), `KS_LAUF_ABBRUCH=aus` (Stopp und Zeitlimit beenden wie früher nur die KI-Kommandozeile samt ihrer Prozessgruppe), `KS_CLI_VERMITTLER=aus` (KI-Kommandozeilen bekommen den API-Key wieder selbst), `KS_CLI_ZIELBINDUNG=aus` (Einstellungsdateien im Projekt- und Login-Ordner können das Ziel der Anfragen wieder ändern). Steht `KS_SANDBOX_NETZ` in der Datei, gilt die Datei; der Eintrag in der `.env` gilt nur, wenn die Datei den Schalter nicht nennt.

12 Sicherheitshinweise

- **Passwort** des Admin nach dem ersten Login ändern; für jeden Menschen ein eigenes Konto.
- **Viewer** nur die Kameras und Apps geben, die sie brauchen; **Shell-Mode** nur, wenn der Nutzer Code auf dem System ausführen darf.
- Zugriff aus dem Internet nur über einen **HTTPS-Reverse-Proxy** (z. B. Nginx Proxy Manager); die Suite selbst spricht HTTP im LAN.
- **Verbund:** Vor dem Koppeln HTTPS einrichten und `LAN_HTTP=aus` in die `.env` eintragen. Eine Kopplung über HTTP lässt sich im lokalen Netz mitschneiden.
- Regelmäßig `./manage.sh backup` — und einmal probehalber ein Restore auf einem Testsystem.
- Alle Geheimnisse (Schlüssel, Passwörter) liegen ausschließlich in der `.env` und im Datenordner, nie in der Release-ZIP.

© 2026 Fabian Hammer · AllInOne-Suite · Handbuch erzeugt mit tools/handbuch_bauen.py