

How-to-Handbuch

WAMAS Lift & Store **Angebotsgenerator – Sicherheit** **und Benutzeranleitung**

WAMAS Lift & Store Angebotsgenerator

Standard · Deutsch (de-DE)

Abgleich Secure Development Policy PO-00153-EN und Bedienung für
Anwender

Marek Sobona / SSI SCHÄFER

 **WAMAS®**

 **SSI SCHÄFER**

V 1.0 · 28.10.2026

Inhaltsverzeichnis

1 Über dieses Dokument.....	4
1.1 Zweck.....	4
1.2 Lesart.....	4
1.3 Dokumentinformationen.....	4
2 Systemüberblick.....	5
2.1 Was die App ist.....	5
2.2 Betrieb.....	5
2.3 Rollen.....	5
3 Sicherheitsstand – erfüllt.....	6
4 Sicherheitsstand – teilweise / offen.....	8
5 Bewertung für internen SSI-Betrieb.....	9
6 Benutzeranleitung Anwender.....	10
6.1 Anmelden.....	10
6.2 Fehlversuche und Sperre.....	10
6.3 Arbeitsablauf.....	10
6.4 Passwort ändern, Passwort vergessen, Abmelden.....	11
7 Kurz-Admin.....	12
8 Stolperfallen.....	13
9 Praxisbeispiele.....	14
9.1 Login-Sperre.....	14
9.2 Angebot als Editor.....	14
10 Quellenverzeichnis.....	15

1 Über dieses Dokument

1.1 Zweck

Dieses Dokument fasst den **Sicherheitsstand** des WAMAS Lift & Store Angebotsgenerators nach den Massnahmen P0 und P1 zusammen (erfüllt / teilweise / offen gemäss PO-00153-EN) und enthält eine **Benutzeranleitung** für Anwender sowie eine Kurzanleitung für Administratoren.

1.2 Lesart

- UI-Bezeichnungen stehen **fett**, Konfigurationsschlüssel und API-Pfade in `monospace`.
- Nicht belegbare Angaben sind als **LUECKE** markiert.
- Der Stand bezieht sich auf Commit 041e207 und Image `wamas-ls-angebot:ui-p1-041e207`.

1.3 Dokumentinformationen

Merkmal	Angabe
Titel	WAMAS Lift & Store Angebotsgenerator – Sicherheit und Benutzeranleitung
Produkt / Release	WAMAS Lift & Store Angebotsgenerator
Dokumentversion	1.0
Stand	28.10.2026
Autor	Marek Sobona / SSI SCHÄFER
Zielgruppe	Anwender, Key-User, Administratoren, Security-Review
Grundlage	CWA Security SW PO-00153-EN v1.00; Live-Stand Image ui-p1-041e207 (Commit 041e207); interner SSI-Betrieb (kein öffentliches Internet)

2 Systemüberblick

2.1 Was die App ist

Webanwendung (FastAPI, Python 3.12, Docker) mit drei Modulen:

1. **Lizenzkalkulator** – IC License Price List (Basic/Advanced, Add-ons, Clients, SLL-Rabatt)
2. **IT-Aufwand** – analog Kalkulation WAMAS Installation.xlsm
3. **Angebot** – druckfertiges Dokument (DOCX/PDF, Excel)

2.2 Betrieb

Element	Wert
URL	https://angebot.mareksobona.synology.me/
Container / Port	CH, Port 8100 hinter Synology-Reverse-Proxy (HTTPS)
Image	wamas-ls-angebot:ui-p1-041e207
Volume	wamas-ls-angebot-runtime
International	Port 8110, von diesem Stand nicht berührt
Scope	interner SSI-SCHÄFER-Betrieb, keine öffentliche Internet-Veröffentlichung

Tab. 1: Betriebsdaten

2.3 Rollen

Rolle	Rechte
viewer	lesen, Export, Vorschau; keine Bearbeitung
sales	nur Verkaufspreise, read-only
editor	wie viewer, zusätzlich speichern
admin	zusätzlich löschen, Benutzerverwaltung, Stammdaten

Tab. 2: Rollenmodell

Quelle: README.md, static/index.html, Auftrag Marek (Scope)

3 Sicherheitsstand – erfüllt

Anforderung	Umsetzung	Beleg
Brute-Force-Schutz Login	5 Fehlversuche/Benutzer bzw. 30/IP in 15 Min. → HTTP 429, Sperre 15 Min., Audit	P0, Abb. 1–2
Passwort-Hashing	PBKDF2-SHA256 600 000 Iterationen, Upgrade beim Login	auth.py PBKDF2_ITERATIONS
Session-Cookie	HttpOnly, SameSite=Lax, Secure bei https-WAMAS_PUBLIC_URL	README, P0
HSTS	max-age=15552000 hinter HTTPS	Abb. 3
Security-Header	CSP (frame-ancestors 'none', script-src 'self'), X-Frame-Options DENY, nosniff, Referrer-Policy	http_security.py
API-Doku in Prod aus	/docs, /redoc, /openapi.json aus im Container bzw. bei https	http_security.py
Bootstrap-Passwort	Datei admin_bootstrap_password (0600), nicht im Log	README
Autorisierung	Rollen per Depends auf Fach-Endpoints, Tests	server.py, test_security_roles.py
Path-Schutz	Angebots-IDs auf [A-Za-z0-9-_] gefiltert	Abgleich 5.2
Container	Non-Root wamas (UID 1000), nicht privileged	Dockerfile
Secrets	nicht im Repo, nur ENV / Runtime-Volume	.gitignore
CORS	explizite Allowlist WAMAS_CORS_ORIGINS	README
CI / SAST / SCA	GitHub Actions: pytest, Bandit (high), pip-audit-Gate, Trivy (HIGH/CRITICAL)	ci.yml
Dependencies	requirements.txt mit Hashes (--require-hashes), Dev getrennt	README
Port-Bind	Standard 127.0.0.1:8100; Live zusätzlich LAN für Proxy	docker-compose.yml, P0
Passwort-Recovery	TLS-Pflicht, Rate-Limit, generische Fehler	Abgleich 5.2

Tab. 3: Erfüllte Anforderungen

Login Rate-Limit (HTTP 429)

Antwortkörper nach wiederholten Fehlversuchen:

```
{"detail": "Zu viele Anmeldeversuche. Bitte später erneut versuchen."}
```

Erwartung: HTTP 429 nach mehreren Fehlversuchen pro Benutzer.

Abb. 1: API-Nachweis Rate-Limit (HTTP 429)

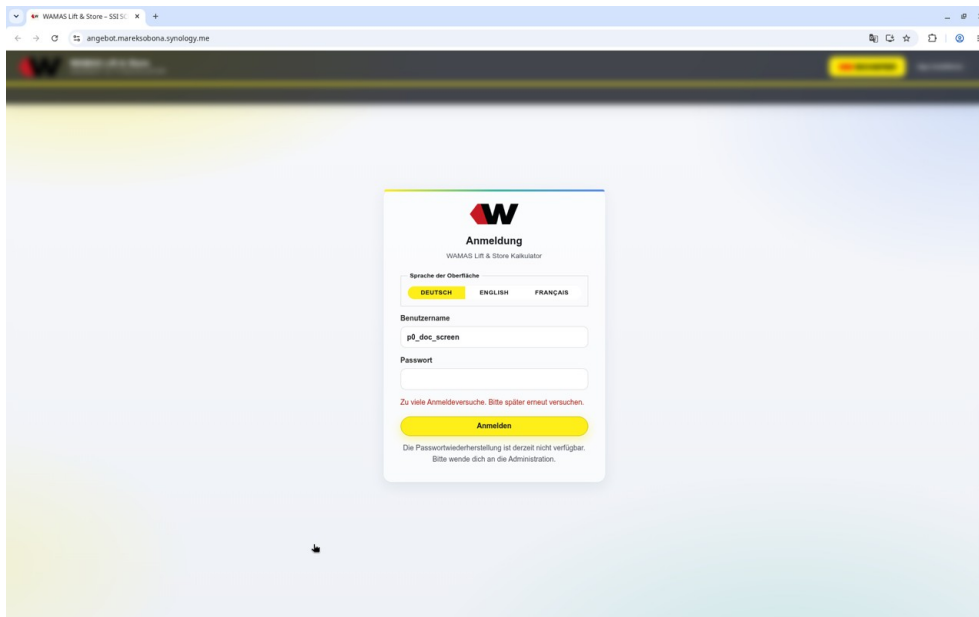


Abb. 2: Sperre in der Oberfläche

API Health + HSTS (öffentlich)

```
HTTP/2 200
server: nginx
date: Wed, 28 Oct 2026 11:43:34 GMT
content-type: application/json
content-length: 201
strict-transport-security: max-age=15552000

Body:
{"ok":true,"status":"ok","service":"WAMAS Lift & Store Calculator","modules":["license","it","offer"],"appVersion":"3.0.0","wamasProductVersion":"2.8","pdf":{"available":true,"provider":"libreoffice"}}
```

Abb. 3: Health 200 mit Strict-Transport-Security

Quelle: Code Commit 041e207, P0-Bericht v1.0, Live-Test 28.10.2026

4 Sicherheitsstand – teilweise / offen

Thema	Status	Bemerkung
Legacy-Hashes	teilweise	Schwache Varianten nur noch einmalig zum Upgrade unmigrierter Konten
Token-Transport	offen	Token im Login-Body; akzeptiert in mehreren Headern/Cookies (auth.py)
CSRF	teilweise	nur SameSite=Lax + CORS, kein CSRF-Token
Proxy / Netz	Hinweis	WAMAS_TRUST_PROXY=1 nur hinter Proxy; LAN-Bind 192.168.0.70:8100
Fehlertexte Export	offen	Inhalt der Fehlermeldungen nicht geprüft
GDPR	offen	kein Verarbeitungsverzeichnis, keine Retention
Backups	offen	Backup-Skripte ohne Verschlüsselung
Encryption at rest	offen	Angebote/Benutzer als JSON im Volume
Supply Chain	teilweise	Basis-Image ohne Digest, kein Dependabot, kein täglicher CI-Lauf
Container-Härtung	offen	kein read_only, cap_drop, no-new-privileges
Governance	offen	kein Threat Model, kein ASVS-Mapping, kein Pentest/DAST-Nachweis

Tab. 4: Offene Punkte

In den Unterlagen nicht beschrieben

Ob Branch-Protection und dokumentierte Reviews auf GitHub aktiv sind, ist nicht geprüft.

Quelle: Abgleich PO-00153 (Stand vor P0/P1), aktualisiert mit Commit 041e207

5 Bewertung für internen SSI-Betrieb

Für den **internen** Betrieb hinter HTTPS ist die App mit P0/P1 **technisch freigabefähig**. Der **Policy-Nachweis** (Prozess, Governance) ist nur **teilweise** erfüllt. Empfehlungen vor breitem Rollout:

1. Legacy-Hash-Pfad entfernen, sobald alle Konten migriert sind.
2. WAMAS_TRUST_PROXY nur hinter dem Reverse Proxy setzen.
3. Export-Fehlertexte prüfen (keine internen Details).
4. Token nur noch per HttpOnly-Cookie; CSRF-Schutz ergänzen.
5. Backups verschlüsseln, GDPR-Verzeichnis und Retention festlegen.

Quelle: Kap. 3 und 4

6 Benutzeranleitung Anwender

6.1 Anmelden

1. <https://angebot.mareksobona.synology.me/> öffnen.
2. Benutzername und Passwort eingeben, **Anmelden** klicken.

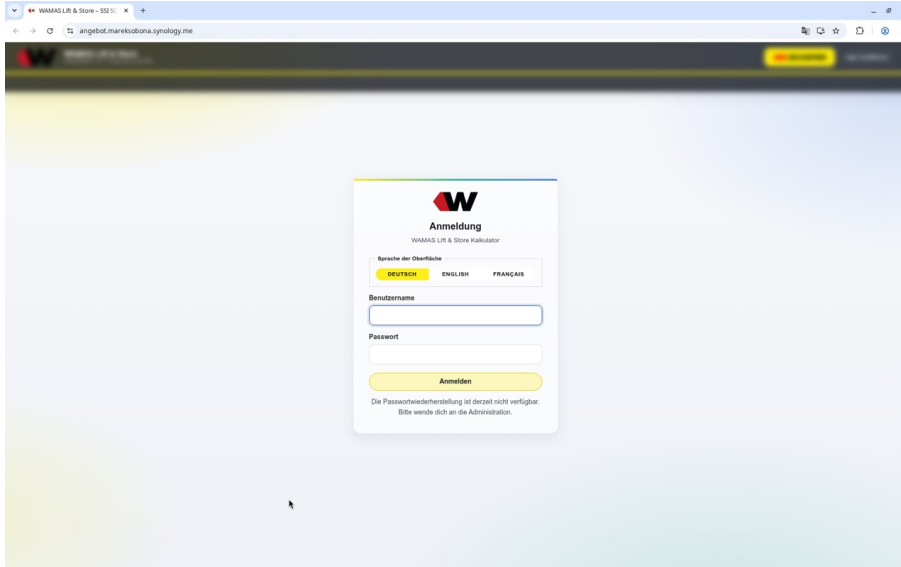


Abb. 4: Anmeldemaske

6.2 Fehlversuche und Sperre

Bei falschen Daten erscheint **Benutzername oder Passwort falsch**. Nach 5 Fehlversuchen: **Zu viele Anmeldeversuche. Bitte später erneut versuchen. – 15 Minuten warten.**

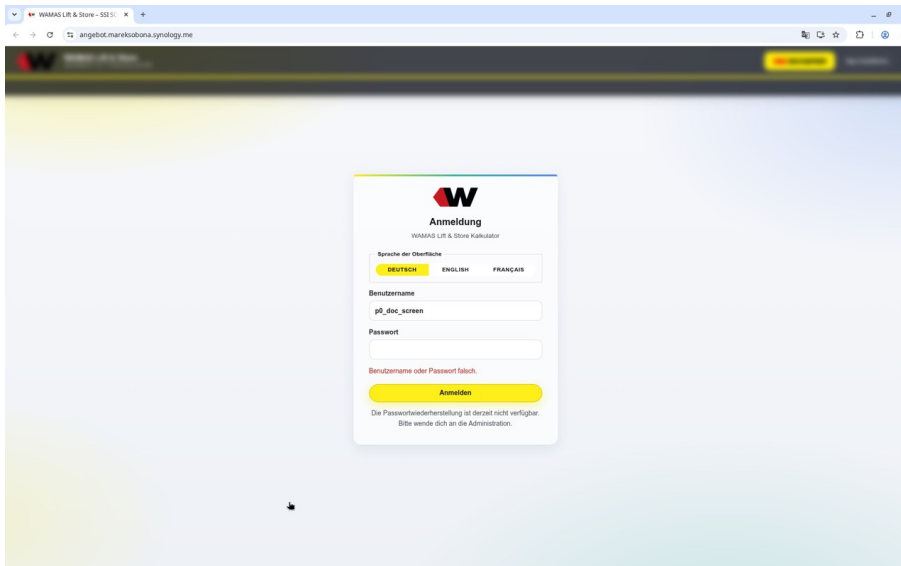


Abb. 5: Fehlermeldung

6.3 Arbeitsablauf

Die obere Navigation führt durch **Lizenzkalkulator** → **IT-Aufwand** → **Angebot**, dazu **Archiv**.

1. **Lizenzkalkulator:** Schritte **1 Kunde**, **2 Angebotsart**, **3 Instanz**, **4 Add-ons**, **5 Clients & Upgrade**; mit **Weiter/Zurück** wechseln; **Lizenzkalkulation speichern** (ab editor). Export **Excel** / **Drucken**, sobald berechnet.
2. **IT-Aufwand:** Schritte **1 Kunde** bis **6 Reise**.
3. **Angebot:** Assistent mit **Weiter/Zurück**; Export als DOCX/PDF/Excel.
4. **Archiv:** gespeicherte Angebote öffnen; **Auswahl löschen** nur für admin.

In den Unterlagen nicht beschrieben

Die genauen Felder und Export-Buttons im Modul **Angebot** sind ohne Screenshot nicht belegt.

6.4 Passwort ändern, Passwort vergessen, Abmelden

- **Passwort ändern** in der Kopfzeile.
- **Passwort vergessen?** auf der Anmeldemaske (nur aktiv, wenn Recovery/Mailversand konfiguriert ist).
- **Abmelden** in der Kopfzeile.

Quelle: static/index.html, server.py, Live-Screenshots 28.10.2026

7 Kurz-Admin

- **Benutzer** (nur admin): anlegen, bearbeiten, Passwort setzen, deaktivieren/löschen. Ohne UI: `python scripts/set_user_password.py <user> <pw>`.
- **Stammdaten** (nur admin).
- **Deploy**: `docker compose up -d --build`; Image-Tag und Origins über `.env`.
- **Health**: `/api/health` → HTTP 200.
- **Backup**: gesamtes Runtime-Volume per `deploy/backup/backup-runtime.sh`, Restore mit `restore-runtime.sh`; Restore-Test quartalsweise.

Deploy-Stand CH 8100

```
Image: wamas-ls-angebot:ui-p0-5db45ac
Ports: 127.0.0.1:8100 und 192.168.0.70:8100
Volume: wamas-ls-angebot-runtime (unverändert, 49 Dateien)
International 8110: unberührt
WAMAS_PUBLIC_URL=https://angebot.mareksobona.synology.me/
WAMAS_TRUST_PROXY=1
PR: https://github.com/msobona/Lift-Store-Angebot-/pull/2 (gemergt)
```

Abb. 6: Deploy-Stand (P0)

In den Unterlagen nicht beschrieben

Abb. 6 zeigt den Stand P0 (`ui-p0-5db45ac`); ein Screenshot zum aktuellen Stand `ui-p1-041e207` fehlt.

Quelle: README.md, deploy/docs/02-volumes-backup.md, 04-runbook.md

8 Stolperfallen

- Tests mit dem eigenen Admin-Konto können 15 Minuten aussperren.
- Lokal über HTTP sind Secure-Cookie/HSTS aus; /docs lokal an.
- Laufender Stack muss neu erzeugt werden, damit Port-Bind gilt.
- Backups enthalten auch gelöschte Angebote.

9 Praxisbeispiele

9.1 Login-Sperre

Ein Benutzer gibt fünfmal ein falsches Passwort ein. Die App meldet **Zu viele Anmeldeversuche...** (HTTP 429). Nach 15 Minuten ist die Anmeldung wieder möglich.

9.2 Angebot als Editor

1. Anmelden als editor.
2. Lizenzkalkulator ausfüllen und speichern.
3. IT-Aufwand erfassen.
4. Im Modul **Angebot** Dokument erzeugen und als PDF exportieren; im **Archiv** wiederfinden.

10 Quellenverzeichnis

Kürzel	Dokument / Artefakt
PO-00153	CWA Security SW – Secure Development Policy v1.00
ABGL	WLS_Angebot_Security_Abgleich_PO-00153.md
P0	WLS_Angebot_Security_P0_Technischer_Bericht_deDE_v1.0
CODE	Commit 041e207: README.md, auth.py, http_security.py, server.py, static/index.html, ci.yml, Dockerfile, docker-compose.yml
DEPLOY	deploy/docs/02-volumes-backup.md, 03-network-https.md, 04-runbook.md