WolfISMS

Wolf's Informations-Sicherheits-Management-System

Admindokumentation

Version 4.01

Stand: November 2026

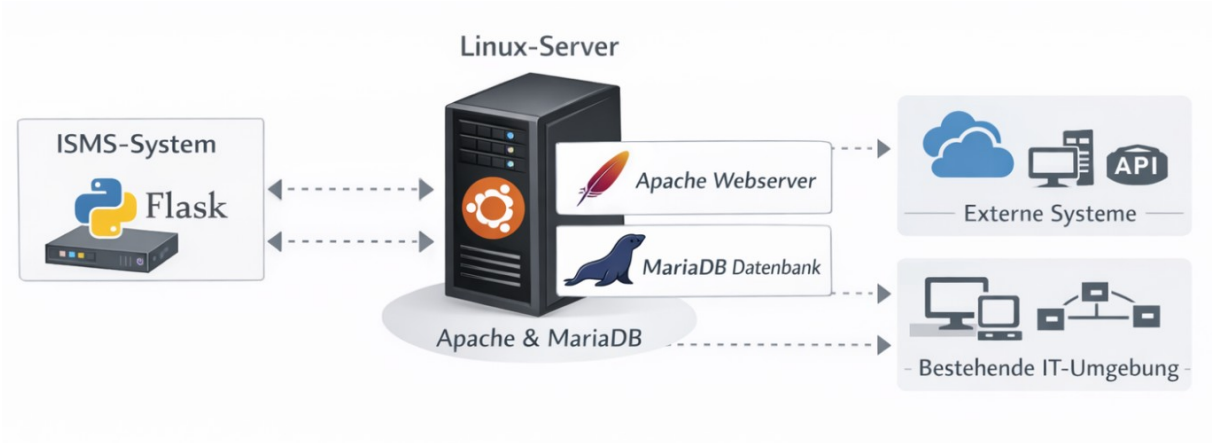
© 2012–2026 WolfInfoSec GmbH

1. Systemübersicht

WolfISMS ist ein webbasiertes Informationssicherheits-Managementsystem (ISMS), das auf dem Python-Framework Flask basiert. Die Anwendung unterstützt Organisationen bei der Umsetzung und Überwachung von Informationssicherheitsmaßnahmen nach gängigen Normen wie ISO/IEC 27001:2022, NIS-2, DSGVO und BSI IT-Grundschutz.

1.1 Technische Grunddaten

Komponente	Details
Framework	Python / Flask (Blueprint-Architektur)
Webserver	Apache / mod_wsgi (WSGI-Schnittstelle via isms.wsgi)
Datenbank	MariaDB/MySQL, lokal oder auf einem eigenen Server (siehe Datenbankanbindung) – Datenbank: "isms"
Version	4.01
Installationspfad	/var/www/isms/
Session-Verzeichnis	/var/www/isms/wolfisms/sessions/
Upload-Verzeichnis	/var/www/isms/wolfisms/uploads/
Request-Limit	512 MB pro Request (MAX_UPLOAD_MB) – bewusst groß, weil die Backup- Wiederherstellung ein komplettes Archiv entgegennimmt
Anhang-Limit	16 MB pro PDF-Anhang (MAX_PDF_MB, durchgesetzt von validate_pdf())
Rate Limiting	2000 Req/Tag, 500 Req/Stunde (global); REST-API ausgenommen (limiter.exempt)
Download-Quelle	https://update.wolfisms.de/download – Online-Framework-Katalog und Handbücher; ein Software-Update aus der Anwendung gibt es seit v4.00 nicht mehr



Systemarchitektur: Flask-Anwendung auf Apache und MariaDB, angebunden an bestehende IT und externe Systeme

1.2 Datenbankanbindung

config.py liest die Verbindung wie alle übrigen Werte aus der Umgebung bzw. aus /etc/wolfisms/secret.env. Seit v3.92 bündelt isms_db.py alles, was vom Datenbanktyp und vom

© 2012–2026 WolfInfoSec GmbH

Verbindungsweg abhängt: Treiber, Port, TLS und die Argumente für die Kommandozeilenwerkzeuge der Datensicherung. Damit lässt sich die Anwendung auch an einen entfernten Datenbankserver anbinden, etwa einen zentralen MariaDB-Cluster.

Variable	Vorgabe	Bedeutung
DB_TYPE	mariadb	mariadb oder mysql; mssql ist vorgesehen und wird derzeit beim Start abgelehnt
DB_HOST	localhost	Hostname oder IP-Adresse des Datenbankservers
DB_PORT	(leer)	Leer = Standardport des Typs (3306)
DB_NAME / DB_USER	isms / wolf	Datenbank und Benutzer
DB_PASSWORD	–	Pflicht, bewusst ohne Vorgabe
DB_SSL_MODE	(leer)	leer = Verhalten von Treiber und Client unverändert; disabled = ohne TLS; required = TLS ohne Zertifikatsprüfung; verify-full = TLS, Zertifikat und Hostname gegen DB_SSL_CA
DB_SSL_CA	(leer)	CA-Datei (PEM); Pflicht bei verify-full
DB_SSL_CERT / DB_SSL_KEY	(leer)	Client-Zertifikat und Schlüssel, falls der Server sie verlangt

- **Prüfung beim Start**: isms_db.aktivieren() wird am Ende von config.py aufgerufen. Ein unbekannter DB_TYPE, ein noch nicht unterstützter Typ, ein ungültiger DB_SSL_MODE oder verify-full ohne DB_SSL_CA bricht den Start mit einer klaren Meldung ab, statt später mit unverständlichen SQL-Fehlern
- **Gilt für jede Verbindung**: Die Module verbinden sich weiterhin selbst per pymysql.connect(host=config.dbhost, …). isms_db ergänzt dort Port und TLS, solange der Aufrufer sie nicht selbst setzt – aber nur für den konfigurierten Datenbankhost. Neuer Code nimmt isms_db.verbinden()
- **Datensicherung**: backup.py übergibt mariadb-dump und mariadb Host, Port, Benutzer und TLS-Optionen aus isms_db.client_argumente(); das Passwort geht wie bisher über MYSQL_PWD und nie über die Befehlszeile
- **Kontrolle**: check_config.py zeigt die aufgelösten Werte an, darunter dbtyp, dbport und die DB_SSL-Angaben

Weitere Datenbanktypen:
Ein neuer Typ ist ein Eintrag in isms_db.TYPEN samt Treiberverzweigung. Nutzbar wird er erst, wenn das SQL der Module darauf umgestellt ist – es ist durchgehend im MariaDB-Dialekt geschrieben. Im Docker-Betrieb schaltet DB_MODUS=extern den Datenbankcontainer ab; siehe Docker-Anleitung.

© 2012–2026 WolfInfoSec GmbH

2. Benutzerverwaltung

Die Benutzerverwaltung ist über den Adminbereich zugänglich. Nur Benutzer mit dem Flag admin=1 haben Zugriff auf diese Funktionen.

2.1 Benutzer-Datenbankstruktur

Die Tabelle users enthält folgende relevante Spalten:

Spaltenname	Bedeutung
id	Primärschlüssel (Auto-Increment)
email	E-Mail-Adresse / Login-Name
password_hash	Gehashtes Passwort (Werkzeug)
vorname / nachname	Name des Benutzers
admin	Administratorrechte (0/1)
readonly	Portal-Only-Zugriff (0/1)
master	Master-Berechtigung (0/1)
assets	Asset-Modul, Zugriffsmatrix und Changemanagement (0=kein, 1=lesen, 2=bearbeiten)
dok	Dokumente-Modul (0/1/2)
mass	Maßnahmen- und Projekte-Modul (0/1/2)
risk	Risiken-Modul (0/1/2)
vendor	Lieferanten-Modul (0/1/2)
audit	Audit-Modul und Managementbewertung (0/1/2)
norm	Controls und Rechtsregister (0/1/2)
pers	Personal-Modul (0/1/2)
play	Playbooks-Modul (0/1/2)
incidents	Incidents-Modul (0/1/2)
schwachstellen	Schwachstellenmodul (0/1/2); die Spalte legt schwachstellen_modul.py beim Laden selbst an
bcm	BCM-Modul (0/1/2)
kontakte	Kontaktmodul (0/1/2)
ziele	Ziele-Modul (0/1/2)
prozesslandkarte	Prozesslandkarte (0/1/2); die Spalte legt isms_helpers.kartenrechte_sicherstellen() selbst an
netzwerkzonen	Netzwerkzonen (0/1/2); die Spalte legt isms_helpers.kartenrechte_sicherstellen() selbst an
ki	KI-Assistenten nutzen (0/1, Vorgabe 0); die Spalte legt isms_helpers.kartenrechte_sicherstellen() selbst an
datenschutz	Datenschutzmodul (0/1/2, Vorgabe 0); die Spalte legt isms_helpers.kartenrechte_sicherstellen() selbst an
custom	Links-Modul (0=kein Zugriff, 1=Zugriff)
aufgabe	Aufgaben-Modul

© 2012–2026 WolfInfoSec GmbH

Spaltenname	Bedeutung
abteilung	Abteilungsname (Freitext, wird weiterhin mitgeschrieben)
abteilung_id	Fremdschlüssel auf Abteilungen – maßgeblich für Auswertungen
rolle	Erste zugewiesene Rolle (Fremdschlüssel Rollen). Seit v3.71 nur noch Spiegel der Zuordnungstabelle user_rollen – maßgeblich ist diese
persnr	Personalnummer
lang	Oberflächensprache (de/en)
dok_read	Kommagetrennte IDs der als gelesen bestätigten Dokumente (Lesebestätigung)
totp_secret	TOTP-Geheimschlüssel (Base32, NULL wenn 2FA nicht eingerichtet)
totp_active	2FA aktiv für diesen Benutzer (0/1)
aktiv	Benutzerkonto aktiv (1) oder gesperrt (0); gesperrte Benutzer können sich nicht anmelden
abwesend_von / abwesend_bis / vertreter_id	Abwesenheit mit Vertreter; siehe Abschnitt Abwesenheit und Vertretung

Migrationen bei Update einer Altinstallation:
ALTER TABLE users ADD COLUMN IF NOT EXISTS ziele TINYINT DEFAULT 0 AFTER kontakte;
ALTER TABLE users ADD COLUMN IF NOT EXISTS custom TINYINT DEFAULT 0;
ALTER TABLE users ADD COLUMN IF NOT EXISTS aktiv TINYINT DEFAULT 1;
Die Spalte abteilung_id wird von isms_helpers.ensure_abteilung_id_spalte() automatisch ergänzt und aus dem bisherigen Freitext befüllt.

2.2 Berechtigungsübersicht

Kürzel	Berechtigung	Beschreibung
ADM	Administrator	Vollzugriff inkl. Benutzerverwaltung und Systemeinstellungen
RO	Portal Only	Nur Lesezugriff über das Benutzerportal, kein Admin-Bereich
MST	Master	Erweiterte Rechte (modulübergreifend): systemweite Aufgaben-/Operational-Plan-Sicht, Rollenpflege, Lesepflicht-Matrix, Lesestatus-Reset
KI	KI-Nutzung	KI-Assistenten nutzen; Administratoren dürfen immer
ASS	Assets	Asset-Management – schließt Zugriffsmatrix und Changemanagement ein
DOK	Dokumente	Dokumentenmanagement
MAS	Maßnahmen	Maßnahmen- und Projektmodul
RSK	Risiken	Risikomanagement
VEN	Lieferanten	Lieferantenmodul
AUD	Audit	Audit-Modul – schließt die Managementbewertung ein
NRM	Controls & Rechtsregister	Controls und rechtliche Anforderungen

© 2012–2026 WolfInfoSec GmbH

Kürzel	Berechtigung	Beschreibung
PER	Personal	Personalmodul
PLY	Playbooks	Playbooks / Notfallpläne
INC	Incidents	Incident-Management
SWM	Schwachstellen	Schwachstellenmodul (CVSS-Score, Quellen, Behandlung)
KON	Kontakte	Kontaktmodul
BCM	BCM	BCM-Modul (BIA je Asset, Notfallpläne, Übungen)
ZIE	Ziele	Ziele-Modul (Unternehmensziele,
		Informationssicherheitsziele, KPIs)
LNK	Links	Links-Modul (custom.py) für externe System-Links
PRK	Prozesslandkarte	Prozesslandkarte (Führungs-, Kern- und Unterstützungsprozesse)
NWZ	Netzwerkzonen	Netzwerkzonen-Karte (Netzsegmentierung nach Vertrauensstufen)
DSC	Datenschutz	Datenschutzmodul (Verarbeitungsverzeichnis, Schwellwertanalyse, DSFA, AVV-Block der Lieferanten)

Drei Module ohne eigene Berechtigungsspalte:
Die Zugriffsmatrix und das Changemanagement teilen sich das Recht assets, die Managementbewertung das Recht audit. Damit ist für beide Module keine Änderung am Benutzermodell nötig – wer Assets pflegt, pflegt auch deren Soll-Berechtigungen; wer interne Audits sieht, sieht auch die Leitungsbewertung.

Ziele: Seit v4.00 steht das Modul als Menüpunkt in der Gruppe Governance & Compliance und hängt allein am Recht ziele – auch Administratoren und Masteruser brauchen es. Zuvor lag es im Adminbereich und stand ihnen ohne dieses Recht offen. Bei der ersten Anfrage nach dem Update überträgt ziele_modul.ziele_recht_sicherstellen() ihnen und dem ältesten Benutzer einmalig die Stufe 2. Der Merker ziele_recht_uebernommen in isms_settings verhindert, dass ein später entzogenes Recht zurückkommt.

Prozesslandkarte und Netzwerkzonen: Seit v4.00 Menüpunkte in der Gruppe Organisation & Inventar mit je einem eigenen Recht (prozesslandkarte, netzwerkzonen), das auch Administratoren und Masteruser brauchen. isms_helpers.kartenrechte_sicherstellen() legt die Spalten selbst an und überträgt bisherigen Berechtigten einmalig die Stufe 2 (Merker kartenrechte_uebernommen). Fehlt eine Spalte, etwa nach dem Einspielen einer älteren Sicherung, zieht isms._benutzer_abfragen() sie beim Laden des Benutzers nach; darf der Datenbankbenutzer kein ALTER, gelten beide Rechte als nicht vergeben, und die Benutzerverwaltung schlägt fehl, bis die Spalten von Hand angelegt sind.

KI-Nutzung: Seit v4.00 ein Ja/Nein-Recht (ki) unter den System-Berechtigungen, Vorgabe aus – auch für vorhandene Benutzer. Ohne das Recht sind alle KI-Assistenten gesperrt; Administratoren brauchen es nicht. Einzelheiten im Abschnitt Benutzerrecht „KI-Nutzung“.

Datenschutz: Seit v4.01 ein Modulrecht (datenschutz) mit Kein Zugriff / Lesen / Bearbeiten, Vorgabe kein Zugriff – auch für vorhandene Benutzer, ohne Übernahme. Der Erstbenutzer einer neuen Installation bekommt Bearbeiten. Einzelheiten im Kapitel Datenschutz.

© 2012–2026 WolfInfoSec GmbH

2.3 Admin-Menüfunktionen

Menüpunkt	URL	Funktion
Alle Benutzer	/admin/all_users	Benutzerübersicht mit Berechtigungs-Badges; Konten aktiv/inaktiv schalten
Benutzer anlegen	/admin/create_user	Neuen Benutzer mit Berechtigungen erstellen
Benutzer bearbeiten	/admin/change_user	Benutzerdaten und Berechtigungen ändern
Benutzer importieren	/admin/import_users	Benutzerstammdaten aus einer Datei übernehmen
Rollen	/admin/rollen	Rollen anlegen und Benutzern zuweisen; auch für Masteruser
Aktivitäten	/admin/activities	Änderungsprotokoll aller Benutzeraktionen
Abteilungen & Teams	/admin/abteilungen	Abteilungen und Teams mit Leitung und Über-/Unterordnung pflegen
Unternehmen	/admin/unternehmen	Stammdaten der eigenen Organisation (Kontext, Größenklasse); auch für Masteruser
Pläne & Register	/plaene	Ablage für Netzwerkpläne, Prozessdarstellungen und das Assetregister
Auswirkungstabelle	/admin/auswirkung	Risikoskala bearbeiten (5 Schweregradstufen)
Bedrohungen	/admin/bedrohungen	Bedrohungskatalog pflegen (Art, Name, Beschreibung)
Schwachstellen	/admin/schwachstellen	Schwachstellenkatalog verwalten
Asset-Kategorien	/admin/asset_kategorien	OBASHI-Kategorien des Assetmanagements pflegen
Risiko-Kategorien	/admin/risiko_kategorien	Kategorienliste des Risikomoduls pflegen
Kategorien	/admin/kategorien	Kategorien von Incidents, Changemanagement, Rechtsregister und Kontakten, Bänder von Prozesslandkarte und Netzwerkzonen; Reiter zu den übrigen Katalogen
Lieferantentypen	/admin/vendor_typen	Typenliste des Lieferantenmoduls pflegen; Häkchen Cloud/SaaS (vendor_typ.cloud)
Risiko-Parameter	/admin/risiko_parameter	Vendor-Gewichte und Risikostufen-Schwellenwerte
Stammdaten	/admin/stammdaten	Verwaltung weiterer Dropdown-Werte (Status, Typen)
Logo hochladen	/admin/logo_upload	Firmenlogo (PNG) ersetzen
Word-Vorlage	/admin/vorlage_upload	DOCX-Exportvorlage hochladen
PDF-Briefpapier	/admin/dok_briefpapier	Kopf-/Fußzeilen-Vorlage (PDF) für den PDF-Export
Lizenz	/admin/lizenz_upload	Lizenzdatei (*.lic) hochladen und aktivieren
API-Keys	/admin/api_keys	REST-API-Keys anlegen, aktivieren, deaktivieren, löschen
Ticketsystem	/admin/ticketsystem	URL-Vorlage, über die die Incident-Nummer ins Ticketsystem verlinkt
Adminhandbuch	/admin/handbuch	Diese Administratordokumentation als PDF herunterladen
About	/about	Version, Lizenz und Lizenzbestimmungen; seit v4.00 eine Kachel statt des Logos oben rechts

© 2012–2026 WolfInfoSec GmbH

Lizenz	/admin/lizenz_upload	Lizenzdatei (*.lic) hochladen und aktivieren
Updates	/admin/updates	Eingebundene JavaScript-Bibliotheken (TinyMCE, bpmn-js) prüfen und aktualisieren; seit v4.00 ohne WolfISMS-Software-Update (im Container-Modus gesperrt)
Backup & Restore	/admin/backup	Sicherung erstellen oder Backup wiederherstellen
E-Mail / SMTP	/admin/smtp	Ausgehenden Mailserver konfigurieren und Testmail versenden
Single Sign-On	/admin/sso	Microsoft- und Google-Anmeldung einrichten, Domain-Allowlist, Discovery-Test
Framework-Import	/admin/normen_import	Framework-Kapitel aus CSV/JSON importieren; Direktinstallation aus dem Online-Framework-Katalog
Lieferantenfragebogen	/admin/ vendor_fragebogen	Fragenkatalog nach ISO/IEC 27001 und NIS2 pflegen
Daten SQL	/admin/daten_sql	Export/Import der Stammdaten von Dokumenten, Risiken und Maßnahmen – einzeln oder als Komplettpaket
Module	/admin/module	Fachmodule ein- und ausschalten, Reihenfolge innerhalb der Menügruppen
Cron-Zeiten	/admin/cron_zeiten	Uhrzeit und Rhythmus der Hintergrund-Jobs festlegen
2FA / TOTP	/admin/totp	Systemweites 2FA (totp_required) ein-/ausschalten
KI-Einstellungen	/admin/ki_einstellungen	KI-Anbieter, Modell, API-Key oder Entra ID (Azure), Prompt-Zusatz; in der freien Version gesperrt
Design & Farben	/admin/design	Farbschema (Petrol, Klassisch, Schwarz / Grau / Weiß), Dunkelmodus als Vorgabe, Farbe der Kopfleiste, Infokacheln ein/aus

Rollen an zwei Stellen:
Rollen werden unter /admin/rollen und im Rollenkatalog der Zugriffsmatrix (/zugriffsmatrix/rollen) gepflegt – es ist dieselbe Liste, siehe Kapitel Rollen und Lesepflicht. Anlegen, Umbenennen und Löschen bleiben Administratoren und Masterusern vorbehalten.

Gliederung der Administrationsseite

Seit v3.91 zeigt die Übersicht /admin ihre Kacheln nicht mehr als eine lange Reihe, sondern in sechs Gruppen mit Überschrift. Alle Kacheln einer Gruppe sind gleich breit.

Gruppe	Kacheln
Benutzer & Zugang	Alle Benutzer, Benutzer anlegen, Rollen, 2FA / TOTP, SSO, Aktivitäten
Organisation & ISMS-Kontext	Unternehmen, Abteilungen & Teams, Pläne & Register

© 2012–2026 WolfInfoSec GmbH

Gruppe	Kacheln
Risiko & Kataloge	Risiko-Parameter, Auswirkung, Kategorien, Bedrohungen, Schwachstellen, Lieferantenfragebogen, Framework-Import
Erscheinungsbild & Vorlagen	Design & Farben, Tooltips, Logo, Word-Vorlage, PDF-Briefpapier
Schnittstellen & KI	E-Mail / SMTP, Ticketsystem, API-Keys, KI-Einstellungen, KI-Verbrauch
System & Betrieb	Module, Lizenz, Backup & Restore, Daten SQL, Cron-Zeiten, Updates, Adminhandbuch, About

- **Sichtbarkeit**: Die Rechteprüfung ist unverändert. Eine Gruppe ohne sichtbare Kachel entfällt samt Überschrift – Updates nicht im Containerbetrieb, der Einrichtungsassistent erst, wenn seine Kachel freigeschaltet ist (dann unter Organisation)
- **Masteruser**: Die Master-Konsole eines reinen Masterusers nutzt dieselben Gruppen und zeigt nur seine Kacheln: Rollen, die Organisationsseiten, Risiko-Parameter und KI-Verbrauch
- **Neue Kacheln**: admin() sammelt die Kacheln unter einem Schlüssel und gibt sie über _kacheln_gruppiert() nach admin_modul.ADMIN_GRUPPEN aus. Eine Kachel ohne Eintrag dort erscheint in einer zusätzlichen Gruppe „Weitere“ – sie geht nicht verloren, gehört aber nachgetragen
- **Rücksprung:** Seit v4.00 steht über jeder Admin-Seite „‹ Administration“ (context_processor inject_admin_zurueck in isms.py, CSS .adm-zurueck) – für Administratoren und Masteruser unter /admin/* und bei den Modulen ohne Eintrag in der Seitenleiste (Organigramm, Pläne). Eigene Zurück-Knöpfe nach /admin sind entfallen; links zeigen Admin-Seiten das normale Hauptmenü

2.4 Abteilungen und Teams

Das frühere Freitextfeld „Abteilung“ ist durch die Stammdatentabelle Abteilungen ersetzt. Eine Zeile ohne parent_id ist eine Abteilung, eine Zeile mit parent_id ein Team bzw. eine Unterabteilung. Jede Abteilung kann eine Leitung (Benutzer) haben.

- **Pflege**: unter /admin/abteilungen – Anlegen, Umbenennen, Leitung zuweisen, übergeordnete Abteilung wählen
- **Grafische Ansicht**: die Schaltfläche „Grafische Ansicht" öffnet das Organigramm (/organigramm) – siehe Kapitel Unternehmen und Strukturkarten
- **Excel-Export**: /admin/abteilungen/export schreibt eine Mappe mit ID, Name, übergeordneter Abteilung, Leitung samt E-Mail, Beschreibung, Reihenfolge und Farbe; die beiden letzten Spalten (Mitarbeiter, Verwendung) sind reine Information
- **Excel-Import**: /admin/abteilungen/import liest dieselbe Mappe wieder ein. Zugeordnet wird über die Spaltenbeschriftung, nicht über die Position; eine Zeile findet ihren Bestand über die ID, sonst über den Namen. Die übergeordnete Abteilung wird über den Namen aufgelöst – auch gegen Abteilungen, die derselbe Import erst anlegt –, die Leitung über die E-Mail-Adresse, ersatzweise über „Nachname, Vorname". Der Import löscht nie, rät nie und schreibt erst nach der Vorschau mit gesetztem Häkchen. Kreise werden auf dem Baum geprüft, wie er nach dem Import aussähe
- **Verwendung**: in users, Assets, Massnahmen, Risiko und kontakte_mitarbeiter
- **Abwärtskompatibilität**: jedes Modul schreibt sowohl die neue Spalte abteilung_id als auch die bisherige Textspalte, damit bestehende Berichte und Filter unverändert weiterlaufen
- **Automatische Migration**: ensure_abteilungen_tabelle() legt die Tabelle beim ersten Aufruf an und übernimmt die vorhandenen Freitextnamen; ensure_abteilung_id_spalte() ergänzt die ID-Spalte je Tabelle und füllt sie über Namensgleichheit

© 2012–2026 WolfInfoSec GmbH

2.5 Passwort vergessen (Selbstbedienung)

Auf der Anmeldeseite steht ein Link „Passwort vergessen". Der Benutzer gibt seine E-Mail-Adresse ein und erhält einen Link, über den er selbst ein neues Passwort setzt – der Administrator muss nicht eingreifen. Verwendet wird derselbe Token-Mechanismus wie bei der Willkommensmail (Tabelle pw_reset_tokens, Route /set_password), nur mit deutlich kürzerer Gültigkeit.

Eigenschaft	Wert
Route	/forgot_password
Gültigkeit des Links	2 Stunden (Willkommensmail: 72 Stunden)
Mindestabstand je Konto	120 Sekunden zwischen zwei Mails
Rate-Limit	10 POST je Stunde und IP-Adresse
Voraussetzung	Ein aktiver SMTP-Versand (siehe /admin/smtp)
Protokollierung	Jeder Versuch landet in der Aktivitätentabelle, auch der erfolglose

Der Link auf der Anmeldeseite erscheint nur dort, wo auch das Formular mit E-Mail und Passwort steht. Bei reiner SSO-Anmeldung verwaltet der Identity Provider das Passwort; ein Reset-Link würde daneben ein lokales Passwort einrichten und damit die Vorgaben des Providers (MFA, Conditional Access) umgehen.

Die Seite antwortet immer gleich.
Ob die Adresse bekannt ist, ob das Konto deaktiviert oder ein SSO-Konto ist – der Anwender sieht stets dieselbe Bestätigung. Andernfalls liesse sich hier durchprobieren, welche Adressen ein Konto haben. Was tatsächlich geschah, steht im Aktivitätenprotokoll. Ist kein Mailversand eingerichtet, sagt die Seite das offen und verweist an den Administrator, statt einen Versand vorzutäuschen.

2.6 Kontosperre nach Fehlversuchen

Seit v3.99 zählt WolfISMS falsche Passwörter je Konto und sperrt die Anmeldung vorübergehend – nach dem Vorbild der 2FA-Code-Eingabe (Abschnitt Sperre nach Fehlversuchen im Kapitel Zwei-Faktor-Authentifizierung). Das IP-Limit der Anmeldeseite allein reicht nicht: es hilft nicht gegen verteilte Versuche und zählt hinter einem Reverse-Proxy nur dann je Client, wenn ISMS_PROXY_HOPS passt. Die Zählung liegt in der Datenbank und gilt damit für alle Worker-Prozesse gemeinsam.

© 2012–2026 WolfInfoSec GmbH

Eigenschaft	Wert
Fehlversuche bis zur Sperre	10 (LOGIN_MAX_FEHLVERSUCHE in isms.py)
Sperrdauer	15 Minuten (LOGIN_SPERRE_MINUTEN); der Zähler fällt beim Sperren auf 0, danach hat das Konto wieder den vollen Satz
Spalten	users.login_fehlversuche und users.login_gesperrt_bis – legt die Anwendung beim ersten Login nach dem Update selbst an
Während der Sperre	Das Passwort wird nicht geprüft, auch ein richtiges nicht; die Meldung nennt die Restzeit in Minuten
Aufheben	Von selbst nach Ablauf. Vorzeitig direkt in der Datenbank: UPDATE users SET login_fehlversuche=0, login_gesperrt_bis=NULL WHERE id=<USER_ID>;
Nicht betroffen	Anmeldungen über Single Sign-On und die 2FA-Code-Eingabe (eigene Sperre)
Protokoll	Aktivitätentabelle: „Fehlanmeldung (falsches Passwort, Konto gesperrt)“ beim Erreichen der Grenze, „Fehlanmeldung (Konto gesperrt)“ für jeden Versuch während der Sperre

Zwei bewusste Entscheidungen: Die Sperrmeldung erscheint nur für ein vorhandenes Konto und verrät damit nach zehn Versuchen, dass es existiert – ein Benutzer muss aber erfahren, warum sein richtiges Passwort nicht angenommen wird. Und alle Helfer scheitern offen: ein Datenbankfehler beim Zählen sperrt niemanden aus, sondern steht im Log.

2.7 Abwesenheit und Vertretung

Jeder Benutzer kann für einen Zeitraum eine Abwesenheit mit einem Vertreter eintragen – auf seiner Kontoseite über „Abwesenheit pflegen“ (/abwesenheit). Während des Zeitraums sieht der Vertreter die Aufgaben des Abwesenden unter „Meine Aufgaben“ und bekommt dessen Erinnerungsmails. Besitzer, Reviewer und Verantwortliche in den Datensätzen bleiben unverändert; nach dem Ende fällt alles von allein zurück.

Administratoren pflegen die Abwesenheit eines anderen Benutzers auf derselben Seite (/abwesenheit?user=<id>), erreichbar aus der Benutzerliste und der Benutzermaske. Es ist bewusst keine zweite Maske: die Prüfregeln stehen damit an einer einzigen Stelle (isms_helpers.vertretung_pruefen), die auch die REST-API benutzt (PUT /api/v1/users/<id>/vertretung).

Spalte in users	Bedeutung
abwesend_von / abwesend_bis	Zeitraum der Abwesenheit (DATE); beide Grenzen zählen mit
vertreter_id	Benutzer-ID des Vertreters

Die drei Spalten legt isms_helpers.ensure_vertretung_spalten() beim ersten Zugriff selbst an; ein Migrationsskript ist nicht nötig.

- **Kein Zeitraum ohne Vertreter**: Zulässig sind nur zwei Zustände – alle drei Felder leer oder eine vollständige Angabe. Ein Zeitraum ohne Vertreter wird abgewiesen, weil die Aufgaben sonst unbemerkt liegen blieben.

© 2012–2026 WolfInfoSec GmbH

- **Wer vertreten kann**: aktive Benutzer ohne Portal-Only-Kennzeichen (readonly), nicht der Abwesende selbst.
- **Ketten**: Ist der Vertreter selbst abwesend, gehen die Aufgaben an dessen Vertreter weiter (vertretung_empfaenger). Ein Ring wird erkannt und abgebrochen.
- **Erinnerungsmails**: aufgaben_mailer.py hängt die fälligen Aufgaben des Abwesenden an den Vertreter um. Offene Lesebestätigungen bleiben beim Abwesenden – sie kann niemand stellvertretend abgeben. Bekommt der Vertreter selbst keine Mails (no_mail), bleiben die Aufgaben beim Abwesenden und das Protokoll vermerkt es.
- **Protokoll**: Jede Änderung steht in der Aktivitätentabelle – mit dem bearbeiteten Benutzer und dem, der sie vorgenommen hat.

2.8 Ticketsystem-Verknüpfung (/admin/ticketsystem)

Wird ein Sicherheitsvorfall zugleich in einem Ticketsystem bearbeitet, lässt sich die Nummer in der Incident-Liste direkt dorthin verlinken. Hinterlegt wird eine URL-Vorlage, etwa https://ticket.example.com/browse/<ticketnr>; der Platzhalter wird durch die Nummer des Vorfalls ersetzt (admin_ticketsystem.py).

- **Speicherort**: isms_settings, Schlüssel ticket_url_template, höchstens 255 Zeichen.
- **Platzhalter**: <ticketnr> ist die dokumentierte Schreibweise; {ticketnr} und andere Groß-/Kleinschreibung werden ebenfalls erkannt. Fehlt er, wird die Nummer an die Vorlage angehängt.
- **Nur http und https**: Eine Vorlage mit anderem Schema wird nicht verlinkt – eine gespeicherte javascript:-Adresse landet nie als Link in der Liste.
- **Ohne Vorlage**: führt die Nummer wie bisher in die Detailansicht des Vorfalls.

2.9 Mehrere Besitzer (objekt_besitzer)

Seit v3.89 können Risiken, Assets, Dokumente, Maßnahmen, Projekte, Controls, Lieferanten, Incidents, Schwachstellen und Changes mehrere Besitzer haben. Im Formular steht statt des Auswahlfelds eine Mehrfachauswahl mit Suchfeld; die gewählten Benutzer stehen oben. Die Besitzer-Spalte der Fachtabelle (Besitzer, bei Incidents Owner, bei Lieferanten besitzer) bleibt dabei der Hauptbesitzer – Sortierung, REST-API, Importe und die Review-Vorbelegung arbeiten unverändert mit ihr. Alle weiteren Besitzer stehen in einer gemeinsamen Tabelle.

Spalte	Typ	Bedeutung
tabelle	VARCHAR(64)	Name der Fachtabelle: Risiko, Assets, Dokumente, Massnahmen, Projekte, Normen, vendor, Incidents, swm_schwachstellen, chm_changes
objekt_id	INT UNSIGNED	ID des Datensatzes in dieser Tabelle
user_id	INT UNSIGNED	weiterer Besitzer (users.id)

Primärschlüssel ist (tabelle, objekt_id, user_id), dazu ein Index auf user_id. Die Tabelle legt sich beim ersten Zugriff selbst an; ein Migrationsskript gibt es nicht.

- **Hauptbesitzer:** Bleibt es, solange er ausgewählt ist. Wird er abgewählt, rückt der erste gewählte Benutzer nach. Ohne Auswahl wird der Speichernde Besitzer – bei Controls bleibt ein bisher leerer Besitzer beim Bearbeiten leer.

© 2012–2026 WolfInfoSec GmbH

- **Rechte:** besitzer_check() prüft Hauptbesitzer und weitere Besitzer gleich. Ist nur ein weiterer Besitzer eingetragen, entfällt der Ersteller-Fallback.
- **Aufgaben und Erinnerungen:** /aufgaben, der Aufgaben-Mailer und die Change-Termine berücksichtigen jeden Besitzer; bei Changes ohne Umsetzer erscheint das Zeitfenster bei allen Besitzern.
- **Dokumente:** Die weiteren Besitzer hängen an der einzelnen Fassung und werden wie Anhänge und Lesepflicht in jede neue Version mitkopiert. Eine Ablehnung im Workflow geht an alle Besitzer.
- **Löschen:** Beim Löschen einer Schwachstelle, eines Changes oder einer Dokumentfassung werden ihre Einträge mit entfernt.
- **REST-API:** Liste und Einzelabruf liefern weitere_besitzer mit, der Filter ?besitzer= trifft Haupt- und weitere Besitzer; gesetzt wird per POST und PATCH (außer Incidents: nur POST).

Datensicherung und SQL-Export
Die Datensicherung erfasst objekt_besitzer wie jede Tabelle. Der bereichsweise SQL-Export (Kapitel Daten SQL-Export/Import) überträgt dagegen nur den Hauptbesitzer – weitere Besitzer sind nach einem Import neu zu setzen.

© 2012–2026 WolfInfoSec GmbH

3. Unternehmen und Strukturkarten

Vier Seiten beschreiben die Organisation selbst statt einzelner Datensätze: die Unternehmensangaben und die drei grafischen Sichten Organigramm, Prozesslandkarte und Netzwerkzonen. Unternehmensangaben und Organigramm stehen ausschließlich Administratoren und Masterusern offen und werden aus dem Adminbereich heraus aufgerufen. Prozesslandkarte und Netzwerkzonen sind seit v4.00 Menüpunkte in der Gruppe Organisation & Inventar und hängen je an einem eigenen Recht (siehe Kapitel Benutzerverwaltung). In isms_helpers.MODULE_REGISTRY stehen alle drei Karten, damit sie sich unter /admin/module abschalten lassen und modul_gate() ihre Adressen dann sperrt.

3.1 Unternehmensangaben (/admin/unternehmen)

Die Kachel „Unternehmen" führt zu den Stammdaten der Organisation, für die das ISMS geführt wird (admin_unternehmen.py). Sie beschreiben den Kontext der Organisation nach ISO/IEC 27001 Kapitel 4; Mitarbeiterzahl und Umsatzklasse bestimmen zusätzlich die Größenklasse nach NIS2.

Feld	Inhalt
Unternehmensname	Pflichtfeld; das Organigramm zeichnet damit die Wurzel des Baums
Adresse	mehrzeilig, je Adresszeile eine Zeile
Rechtsform	Textfeld mit Vorschlagsliste (GmbH, AG, e. V. …); ausländische Rechtsformen lassen sich frei eintragen
Mitarbeiterzahl	Vollzeitäquivalente als ganze Zahl
Umsatzklasse	vier Stufen nach der KMU-Definition der EU (Empfehlung 2003/361/EG)
Standorte / Länder	mehrzeilig, z. B. „München (DE)"
Branche	Freitext
NACE-Code	NACE Rev. 2 bzw. deutsche WZ-Unterklasse, etwa „J", „62" oder „62.01.0"; ein Freitext wird beanstandet

Die einzige Systemeinstellung, die auch Masteruser pflegen dürfen.
Alle übrigen Kacheln des Adminbereichs sind Administratoren vorbehalten. Die Angaben liegen in der Tabelle unternehmen mit genau einer Zeile (id=1), analog zu smtp_settings und sso_settings; sie legt sich beim Zugriff selbst an. Andere Module lesen die Zeile über admin_unternehmen.unternehmen_daten().

3.2 Organigramm (/organigramm)

Das Organigramm ist die grafische Sicht auf die Tabelle Abteilungen (organigramm_modul.py) – ein Kasten je Abteilung bzw. Team, eine Verbindung je parent_id. Es gibt keine zweite Datenhaltung: gezeichnet wird, was /admin/abteilungen tabellarisch pflegt. Der Weg dorthin führt über die Schaltfläche „Grafische Ansicht" in der Abteilungsverwaltung, zurück geht es über „Tabellenansicht".

- **Im Kasten**: Leitung (Abteilungen.leitung_id), die zugeordneten Benutzer (users.abteilung_id) und die Anzahl der Assets, Risiken und Maßnahmen der Abteilung

© 2012–2026 WolfInfoSec GmbH

- **Wurzel**: der Unternehmensname aus den Unternehmensangaben. Sie ist kein Datensatz, sondern die Klammer um die oberste Ebene – und das Ablageziel, um eine Abteilung wieder nach oben zu holen
- **Umhängen**: ein Kasten wird auf einen anderen gezogen und schreibt damit parent_id. Zyklen werden auf beiden Seiten verhindert, im Browser und im Server
- **Bedienung**: Ziehen verschiebt das Blatt, das Mausrad zoomt, ein Klick öffnet das Seitenpanel mit Name, Leitung, übergeordneter Abteilung, Beschreibung und Kopffarbe
- **Ausgabe**: PNG, SVG und Drucken (A4 quer eingepasst)

Datenbank: keine neue Tabelle. Abteilungen bekommt bei Bedarf per Lazy ALTER TABLE die drei optionalen Spalten beschreibung, sortierung und farbe (isms_helpers.ensure_abteilung_zusatzspalten). Kein anderes Modul liest sie.

3.3 Prozesslandkarte (/prozesslandkarte)

Die Prozesslandkarte zeigt die Prozesse in Bändern – vorgegeben sind die drei klassischen Führungs-, Kern- und Unterstützungsprozesse, anpassbar unter /admin/kategorien (prozesslandkarte_modul.py). Ein Kasten ist ein Prozess, ein Band eine Prozesskategorie. Ein Prozess mit parent_id bekommt keinen eigenen Kasten, sondern steht als Zeile im Kasten seines Hauptprozesses. Aufgerufen wird die Karte über den Menüpunkt in der Gruppe Organisation & Inventar; den Zugriff regelt das Recht prozesslandkarte (1 = Lesen, 2 = Bearbeiten).

- **Im Kasten**: Prozessverantwortlicher (users.id), zuständige Abteilung und die Anzahl der Teilprozesse
- **Prozessdokument**: ist ein Dokument verknüpft (Dokumente.id), führt aus dem Seitenpanel ein Link dorthin
- **Verschieben**: ein Kasten wird in ein anderes Band oder an eine andere Stelle des Bandes gezogen; der Server vergibt die Sortierwerte in Zehnerschritten neu

Datenbank: die selbstanlegende Tabelle prozesslandkarte (name, kategorie, parent_id, verantwortlich_id, abteilung_id, dokument_id, beschreibung, sortierung, farbe). Fremdschlüssel gibt es bewusst nicht – zeigt eine Spalte nach dem Löschen eines Benutzers, einer Abteilung oder eines Dokuments ins Leere, bleibt der Prozess stehen und die Zeile im Kasten entfällt.

3.4 Netzwerkzonen (/netzwerkzonen)

Die Netzwerkzonen zeigen die Netzsegmentierung in Bändern, die von oben nach unten vom ungeschützten zum am stärksten geschützten Netz führen (netzwerkzonen_modul.py), vorgegeben und unter /admin/kategorien anpassbar: Externe Netze, DMZ/Perimeter, Interne Netze, OT/Produktion und Hochsicherheit/Management. Ein Kasten ist eine Zone, ein Band eine Vertrauensstufe. Eine Zone mit parent_id ist kein eigener Kasten, sondern ein System bzw. Subnetz in einer Zone. Aufgerufen wird die Karte über den Menüpunkt in der Gruppe Organisation & Inventar; den Zugriff regelt das Recht netzwerkzonen (1 = Lesen, 2 = Bearbeiten).

- **Im Kasten**: Zonenverantwortlicher, zuständige Abteilung, VLAN und Adressbereich sowie die Anzahl der Systeme
- **Schutzbedarf**: normal, hoch oder sehr hoch – als Abzeichen am Kasten
- **Netzkonzept**: ist ein Dokument verknüpft, führt aus dem Seitenpanel ein Link dorthin
- **Verschieben**: wie in der Prozesslandkarte schreibt das Ziehen Vertrauensstufe und Reihenfolge

Datenbank: die selbstanlegende Tabelle netzwerkzonen – zusätzlich zu den Spalten der Prozesslandkarte mit vlan, ip_bereich und schutzbedarf.

© 2012–2026 WolfInfoSec GmbH

Bänder: Beide Karten lesen ihre Bänder seit v3.92 aus dem Kategorien-Katalog. Die Spalte kategorie speichert den Schlüssel des Bandes (etwa kern oder dmz); Bezeichnung, Farbe und Reihenfolge kommen aus stamm_kategorie

3.5 Pläne & Register (/plaene)

Neben den gezeichneten Karten gibt es eine Ablage für Pläne, die außerhalb von WolfISMS entstehen (plaene_modul.py). Wie das Organigramm steht sie nur Administratoren und Masterusern offen und wird über die Kachel „Pläne & Register“ im Adminbereich aufgerufen; einen Menüpunkt gibt es nicht. Unter /admin/module lässt sie sich abschalten.

Ablage	Dateitypen	Grenze
Netzwerkpläne (art = netzplan)	PNG, PDF, draw.io (.drawio, .xml)	32 MB
Prozesse (art = prozess)	PNG, PDF, draw.io (.drawio, .xml)	32 MB
Assetregister (art = assetregister)	CSV, XLS, XLSX, ODS	16 MB

- **Prüfung des Inhalts**: Nicht die Endung zählt, sondern die Signatur der Datei – eine als PNG umbenannte ausführbare Datei wird abgewiesen. draw.io-Dateien müssen ein <mxfile>- oder <mxGraphModel>-Element enthalten.
- **Anzeige**: Ein PNG erscheint direkt, ein PDF öffnet in einem neuen Tab. Bewusst ohne <iframe>: X-Frame-Options DENY und object-src none ließen ein eingebettetes PDF leer. draw.io-Dateien lassen sich nicht darstellen (der Betrachter wäre fremdes JavaScript) – die Detailseite zeigt die Seitennamen und bietet den Download an.
- **Assetregister**: wird derzeit nur abgelegt. Die Spalte importiert_am ist für einen späteren Import in die Assets vorgesehen und bleibt bis dahin leer.
- **Datenbank und Dateien**: Tabelle plaene_dateien (selbstanlegend); die Dateien liegen mit UUID-Präfix unter <upload_dir>/plaene/<art>/ und sind damit Teil der Datensicherung der Uploads.
- **Schreibschutz**: Bei abgelaufener Lizenz bleiben die Pläne sichtbar, nur Hochladen, Bearbeiten und Löschen sind gesperrt.

3.6 Kategorien der Module (/admin/kategorien)

Seit v3.92 stehen die Kategorien der Module, die bisher fest im Code standen, in einer gemeinsamen Tabelle stamm_kategorie (kategorien_katalog.py). Gepflegt werden sie unter /admin/kategorien (admin_kategorien.py) – eine Seite mit einem Reiter je Bereich, erreichbar über die Kachel „Kategorien“ in der Gruppe Risiko & Kataloge und nur für Administratoren.

Bereich	Gespeichert in	Art
Incidents	Incidents.Kategorie	Wertliste; „ohne“ ist fest vorgegeben
Change-Management	chm_changes.kategorie	Wertliste
Rechtsregister	rechtsregister.kategorie	Wertliste mit Farbe
Kontakte – Dienstleister	kontakte_dienstleister.kategorie	Wertliste; bisher Freitext, Startwerte aus dem Bestand
Netzwerkzonen – Bänder	netzwerkzonen.kategorie	Bänder mit Schlüssel und Farbe, Reihenfolge von außen nach innen
Prozesslandkarte – Bänder	prozesslandkarte.kategorie	Bänder mit Schlüssel und Farbe, Reihenfolge von oben nach unten

© 2012–2026 WolfInfoSec GmbH

- **Wertlisten**: Der Datensatz speichert den deutschen Namen. Umbenennen zieht alle Datensätze per UPDATE mit; der englische Name ist nur die Anzeige in der englischen Oberfläche
- **Bänder**: Netzwerkzonen und Prozesslandkarte speichern einen festen Schlüssel (z. B. kern, dmz). Bezeichnung, englischer Name, Farbe und Reihenfolge sind frei; Umbenennen ändert nur die Beschriftung. Vom letzten Band muss eines bleiben
- **Löschen**: wird abgelehnt, solange ein Datensatz die Kategorie verwendet
- **Erstbefüllung**: Beim ersten Zugriff wird ein Bereich aus den bisherigen Code-Vorgaben und den bereits verwendeten Werten befüllt. Dass das geschehen ist, merkt sich stamm_kategorie_bereich – gelöschte Vorgaben kommen deshalb nicht zurück
- **Andere Kataloge**: Asset-, Risiko-, Bedrohungs- und Schwachstellenkategorien haben eigene Tabellen und Masken (/admin/asset_kategorien, /admin/risiko_kategorien, /admin/bedrohungen, /admin/schwachstellen); die Seite verlinkt sie nur als Reiter
- **REST-API**: POST /api/v1/incidents prüft die Kategorie gegen denselben Katalog – eine hier angelegte Kategorie ist sofort auch über die API zulässig

Spalte	Bedeutung
bereich	Schlüssel des Bereichs aus kategorien_katalog.BEREICHE
schluessel	Nur bei Bändern: der gespeicherte Schlüssel
name / name_en	Bezeichnung deutsch (eindeutig je Bereich) und englisch
farbe	Hex-Farbe (#RRGGBB) bei Bändern und im Rechtsregister
sortierung	Reihenfolge in Auswahllisten und Karten

Für Entwickler:
Die früheren Konstanten sind Funktionen: incidents.KATEGORIELISTE(), changemanagement_modul.KATEGORIEN(), netzwerkzonen_modul.KATEGORIEN() und prozesslandkarte_modul.KATEGORIEN(). Ein neuer Bereich ist ein Eintrag in kategorien_katalog.BEREICHE mit Tabelle, Spalte und Vorgaben.

© 2012–2026 WolfInfoSec GmbH

4. Rollen und Lesepflicht

Eine Rolle beschreibt, welche Funktion jemand im ISMS wahrnimmt – Informationssicherheitsbeauftragter, Datenschutzbeauftragter, IT-Leitung, Auditor. Sie ist kein Recht: was ein Benutzer sehen und ändern darf, steht weiterhin allein in den Berechtigungsspalten der Tabelle users (Kapitel Benutzerverwaltung). Die Rolle beantwortet die andere Frage – wer welche Aufgabe innehat.

Rollen sind unternehmensweite Stammdaten in der Tabelle Rollen und werden an drei Stellen gebraucht: als Spalten der Zugriffsmatrix, als Merkmal im Personalmodul (Personal.Rollen) und – seit v3.71 – als Zuordnung zu Benutzerkonten. Auf dieser Zuordnung setzt die Lesepflicht-Matrix des Policy-Portals auf.

4.1 Rollenverwaltung (/admin/rollen)

Die Kachel „Rollen" im Adminbereich führt zur Rollenverwaltung (admin_modul.admin_rollen). Sie steht Administratoren und Masterusern offen. Die Liste zeigt je Rolle das Kürzel, unter dem sie in der Zugriffsmatrix steht, ob sie dort eine Spalte hat, und wer sie innehat.

Funktion	Verhalten
Anlegen	Name ist Pflicht und wird über Pruefung gegen die Spaltenbreite geprüft. Ein bereits vergebener Name wird abgewiesen – verglichen wird ohne Rücksicht auf Groß- und Kleinschreibung, genau wie im Rollenkatalog
Umbenennen	dieselbe Dublettenprüfung; der Katalogeintrag der Zugriffsmatrix wandert mit
Mitglieder	die Benutzer der Rolle werden angehakt (isms_helpers.set_rolle_users). Das ist die Gegenrichtung zur Mehrfachauswahl im Benutzerformular – dieselbe Tabelle, nur von der anderen Seite her gepflegt
In der Matrix aktiv	schaltet das Aktiv-Kennzeichen des Rollenkatalogs um. Es entscheidet allein darüber, ob die Rolle eine Spalte in der Zugriffsmatrix bekommt, nicht darüber, wer sie innehat
Löschen	nur, wenn die Rolle nirgends mehr verwendet wird (siehe unten)

Vor dem Löschen prüft die Maske vier Fundstellen: die Benutzer der Rolle (user_rollen), die belegten Zellen der Zugriffsmatrix (zugriff_matrix mit einer Stufe außer „n" und „–"), die Personaleinträge, die sie in Personal.Rollen führen, und die Dokumente, die sie in der Lesepflicht-Matrix nennen (dok_lesepflicht). Wird sie noch verwendet, meldet die Maske, wo – gelöscht wird nichts. Wer die Rolle nur aus der Zugriffsmatrix nehmen will, setzt sie dort auf „nicht aktiv", statt sie zu löschen.

© 2012–2026 WolfInfoSec GmbH

Eine frische Installation startet mit fünf Rollen.
ensure_rollen_tabellen() legt beim ersten Zugriff Informationssicherheitsbeauftragter, Datenschutzbeauftragter, Geschäftsführung / Vorstand, IT-Leitung und Auditor an – aber nur, wenn die Tabelle dabei neu entsteht. Eine bestehende Installation bringt ihre eigene Liste mit und wird nicht angefasst; eine bewusst leergeräumte Liste füllt sich nicht wieder auf.

4.2 Zuordnung zu Benutzern (user_rollen)

Bis v3.70 lag die Rolle eines Benutzers in der Spalte users.rolle – dort passt genau eine Rollen-ID hinein. Wer zugleich ISB und Datenschutzbeauftragter ist, ließ sich damit nicht abbilden. Die Zuordnung liegt deshalb jetzt in der n:m-Tabelle user_rollen.

- **Benutzerformular**: /admin/create_user und /admin/change_user zeigen eine Mehrfachauswahl aller Rollen. Rollen ohne Spalte in der Zugriffsmatrix sind als solche gekennzeichnet, bleiben aber wählbar
- **Benutzerliste**: /admin/all_users führt eine Spalte „Rollen"; der Excel-Export der Benutzerliste ebenfalls
- **Profilseite**: jeder Benutzer sieht unter „Meine Rollen", welche Rollen ihm zugeordnet sind
- **Altbestandsspalte**: users.rolle bleibt bestehen und wird als Spiegel der alphabetisch ersten Rolle fortgeschrieben (Dual-Write wie bei Abteilung/abteilung_id). Alle Stellen, die diese Spalte nur lesen, laufen unverändert weiter
- **Übernahme der Altdaten**: legt ensure_rollen_tabellen() die Tabelle user_rollen neu an, übernimmt sie einmalig die bisherige Einzelrolle aus users.rolle. Übernommen wird nur, was auf eine vorhandene Rolle zeigt; verwaiste Verweise und Platzhalter wie „keine" fallen weg. Der Backfill hängt am Neuanlegen der Tabelle und nicht daran, ob sie leer ist – sonst kämen gelöschte Zuordnungen beim nächsten Seitenaufruf zurück

4.3 Rollenkatalog der Zugriffsmatrix

Die Zugriffsmatrix führt zu jeder Rolle einen Katalogeintrag mit einem Kürzel (R01…Rnn) und einem Aktiv-Kennzeichen. Es ist dieselbe Rollenliste wie unter /admin/rollen, nur mit den Zusatzangaben, die die Matrix braucht. Damit sie dieselbe bleibt, fasst die Adminmaske die Katalog- und Matrixtabellen nicht selbst an, sondern ruft dafür drei Funktionen in zugriffsmatrix_modul auf: eine dort angelegte Rolle bekommt sofort ihren Katalogeintrag, eine gelöschte nimmt Katalogeintrag und Matrixzellen mit, und das Aktiv-Kennzeichen ist in beiden Masken sichtbar. Die drei arbeiten unabhängig davon, ob das Modul Zugriffsmatrix unter /admin/module eingeschaltet ist – die Tabellen sollen auch dann zusammenpassen.

Lizenzvoraussetzung: Die Zugriffsmatrix gehört zu Advanced (ADV, mit und ohne „+“) und zu Enterprise (ENT); in den übrigen ist auch ihr Rollenkatalog gesperrt. Rollen anlegen, umbenennen und löschen bleibt davon unberührt – das geschieht unter /admin/rollen und steht in jeder Edition zur Verfügung. Wo eine Maske auf den Rollenkatalog verweist (etwa die Lesepflicht-Matrix des Portals), führt der Verweis in den übrigen Editionen dorthin.

4.4 Lesepflicht-Matrix (/portal/lesematrix)

Bisher galt im Policy-Portal: jedes freigegebene Dokument eines Typs mit Lesepflicht musste von jedem Benutzer gelesen und bestätigt werden. Wer nur die Besucherordnung braucht, bekam damit auch die Kryptorichtlinie in seine Leseliste. Seit v3.71 entscheidet eine Matrix aus Dokument

© 2012–2026 WolfInfoSec GmbH

und Rolle, wer was lesen muss (Tabelle dok_lesepflicht). Gepflegt wird sie vom Masteruser unter /portal/lesematrix; erreichbar ist sie über die Schaltfläche „Lesepflicht-Matrix" im Portal.

- **Zeilen**: die freigegebenen Dokumente (Status 6) der Typen, für die eine Lesebestätigung vorgesehen ist – die Positivliste LESE_TYPEN in portal_modul.py
- **Spalten**: die Rollen aus der Tabelle Rollen, mit der Zahl ihrer Benutzer. Ein Umschalter je Spalte setzt oder löscht die ganze Spalte
- **Häkchen**: wer diese Rolle hat, muss das Dokument lesen und bestätigen
- **Leere Zeile**: niemand muss es lesen. Das Dokument bleibt im Portal sichtbar und lässt sich freiwillig bestätigen, zählt aber nicht in Fortschritt, Lesestatus und Erinnerungen
- **Protokollierung**: jede geänderte Zeile geht mit der Zahl der zugeordneten Rollen in die Aktivitäten

Die Matrix ist deny by default – und es gibt bewusst keinen Backfill.
Ein frisch freigegebenes Dokument hat keine Zeile und damit keine Lesepflicht; für Bestandsdokumente gilt beim Update dasselbe. Der alte Zustand „alle müssen alles lesen" wird also nicht übernommen, sondern einmal bewusst gesetzt. Nach dem Update auf v3.71 ist deshalb zuerst die Matrix zu füllen – sonst weist WolfISMS niemandem mehr eine Lesepflicht zu und verschickt keine Erinnerungen mehr.

Damit das nicht unbemerkt bleibt, meldet sich das Dokumentenmodul: wird eine Richtlinie freigegeben, für die keine Rolle eingetragen ist, weist ein Hinweis darauf hin, dass sie niemand lesen muss und keine Mail ausgelöst wird. Masteruser bekommen dazu den Weg in die Matrix genannt. Wird ein Dokument gelöscht, verschwindet auch seine Zeile aus der Matrix.

Die Matrix wirkt an fünf Stellen:

Stelle	Wirkung
Policy-Portal	der Fortschrittsbalken zählt nur die Pflichtlektüre. Dokumente ohne Lesepflicht tragen die Kennzeichnung „Freiwillig" und stehen hinter der Pflichtlektüre
Aufgaben	offene Lesebestätigungen erscheinen nur noch für Dokumente, die der Benutzer lesen muss
Erinnerungsmailer	gemahnt wird ausschließlich die Pflichtlektüre; ein Benutzer ohne Rolle bekommt keine Mahnung
Freigabemail	die Mail bei der Freigabe geht an die Benutzer der zugeordneten Rollen statt an alle. Ist keine Rolle zugeordnet, geht keine Mail hinaus
Personalmodul	Lesestatus und Excel-Export zählen die Pflichtlektüre je Mitarbeiter. Die Matrix im Export kennt drei Zustände: gelesen, offen, keine Lesepflicht

4.5 Datenbank

Alle drei Tabellen legen die Helfer in isms_helpers bei Bedarf selbst an (CREATE TABLE IF NOT EXISTS beim ersten Zugriff) – ein Migrationsskript ist nicht nötig.

Tabelle	Aufbau und Zweck
Rollen	id, rolle – die Rollenliste selbst. Gibt es seit jeher; neu ist nur der Startbestand bei einer frischen Installation
user_rollen	user_id, rolle_id als zusammengesetzter Primärschlüssel, dazu ein Index auf

© 2012–2026 WolfInfoSec GmbH

Tabelle	Aufbau und Zweck
	rolle_id. Ordnet Benutzer und Rollen einander zu (n:m)
dok_lesepflicht	dok_id, rolle_id als zusammengesetzter Primärschlüssel, dazu ein Index auf rolle_id. Legt fest, welche Rolle welches Dokument lesen muss

Beide Auswertungen verbinden über Rollen.
Die Abfragen der Lesepflicht schließen die Tabelle Rollen mit ein. Eine Zeile, deren Rolle es nicht mehr gibt, fällt damit von selbst heraus und kann keine Lesepflicht mehr auslösen.

© 2012–2026 WolfInfoSec GmbH

5. Datensicherung (Backup)

Das Backup-Modul erstellt vollständige Systemsicherungen (Datenbank + hochgeladene Dateien) als tar.gz-Archiv und ermöglicht die Wiederherstellung über die Weboberfläche.

5.1 Backup-Inhalt

Archivpfad	Inhalt
db/isms_db.sql	Datenbank-Dump (mariadb-dump bzw. mysqldump --single-transaction --routines --triggers), unkomprimiert im Archiv abgelegt; Host, Port und TLS kommen aus der Datenbankanbindung
uploads/	1:1-Kopie des Upload-Verzeichnisses (config.upload_dir) mit allen hochgeladenen Dateien
Nicht enthalten	Anwendungscode (/var/www/isms/), Konfigurationsdateien (config.py, /etc/wolfisms/secret.env, isms.conf) und die Lizenzdatei werden NICHT gesichert – diese separat sichern (z. B. Versionskontrolle, Konfigurationsmanagement)
Archivname	isms_backup_YYYYMMDD_HHMMSS.tar.gz

5.2 Backup erstellen

- Admin-Bereich → "Backup & Restore" → Schaltfläche "Backup erstellen"
- Alternativ: Direktlink /backup/download für sofortigen Download

Das Archiv wird in /tmp erstellt und nach dem Download automatisch gelöscht.
Für automatisierte Backups: eigener Cronjob mit curl/wget gegen /backup/download oder direkt mysqldump + tar auf dem Server. Die unter /admin/cron_zeiten konfigurierbaren Zeitpläne steuern nur die Hintergrund-Jobs für Aufgaben-Erinnerungen und die Risikoscore-Historie – nicht das Backup.

5.3 Backup wiederherstellen (UI)

- Admin-Bereich → "Backup & Restore" → Backup-Datei wählen (isms_backup_*.tar.gz, oder ein reiner .sql/.sql.gz-Dump eines Altbackups)
- Kombiniertes Backup (.tar.gz): stellt sowohl die Datenbank (db/isms_db.sql) als auch die hochgeladenen Dateien (uploads/ → config.upload_dir) wieder her; das bisherige Upload-Verzeichnis wird vorher automatisch als Zeitstempel-Kopie gesichert (*.bak_YYYYMMDD_HHMMSS)
- Reiner Datenbank-Dump (.sql/.sql.gz): stellt nur die Datenbank wieder her, hochgeladene Dateien bleiben unverändert
- Ein Apache-Neustart ist für den Restore selbst nicht erforderlich; nach einer Wiederherstellung empfiehlt sich dennoch systemctl restart apache2, um laufende Sessions und Caches zu bereinigen

Endpunkt / Verhalten	Details
POST /backup/restore	Multipart-Form, Feld file: entweder komplettes Backup (.tar.gz/.tgz) oder reiner Datenbank-Dump (.sql/.sql.gz/.gz/.gzip); das Format wird anhand von Dateiendung und Magic-Bytes automatisch erkannt
HTTP 200	Wird auch bei teilweisem Fehlschlag zurückgegeben – maßgeblich ist

© 2012–2026 WolfInfoSec GmbH

Endpunkt / Verhalten	Details
	das Feld status ("ok"/"error") im JSON-Body, nicht der HTTP-Status
HTTP 403	Nicht als Administrator angemeldet
HTTP 400	Keine Datei ausgewählt oder nicht unterstütztes Dateiformat

5.4 Zusätzliche Prüfungen beim Restore

Der Wiederherstellungsvorgang wurde gegen unvollständige und manipulierte Archive gehärtet:

- **Tabellen-Scan vor dem Einspielen**: der Dump wird vorab auf die enthaltenen Tabellen untersucht; ein Dump ohne erkennbare Tabellen wird abgelehnt
- **Pre-Drop**: die Zieltabellen werden vor dem Import kontrolliert entfernt, damit keine Reste eines älteren Schemas zurückbleiben
- **Benutzer-Momentaufnahme**: die vorhandenen Benutzerkonten werden vor dem Import gesichert, damit nach einem fehlgeschlagenen Restore weiterhin eine Anmeldung möglich ist
- **Prüfung nach dem Import**: nach dem Einspielen wird geprüft, ob die erwarteten Tabellen und Daten tatsächlich vorhanden sind; das Ergebnis erscheint im Restore-Protokoll
- **Path-Traversal- und Symlink-Schutz**: _safe_extract_tar() prüft jeden Archiveintrag; präparierte Archive werden mit einem Fehler abgelehnt
- **Schema-Merker:** Nach einem erfolgreichen Restore setzt die Anwendung die Merker der Lazy-ALTER-Helfer zurück (isms_helpers.einmal_alle_zuruecksetzen), damit nachgerüstete Spalten bei einem Dump mit älterem Schema wieder angelegt werden. Das gilt nur für den ausführenden Prozess – bei mehreren Workern rät das Restore-Protokoll zum Neustart

Achtung: Restore überschreibt vorhandene Dateien und die gesamte Datenbank unwiderruflich. Vor dem Restore immer ein aktuelles Backup des Ist-Zustands erstellen! Der Endpunkt ist wie alle Admin-Routen nur für current_user.admin zugänglich.

5.5 Manueller Restore (Kommandozeile)

```
tar -xzf isms_backup_*.tar.gz -C /tmp/restore/
# Datenbank aus dem Dump im Archiv wiederherstellen
mysql -u isms -p isms < /tmp/restore/db/isms_db.sql
# Hochgeladene Dateien wiederherstellen (Zielpfad = config.upload_dir)
cp -r /tmp/restore/uploads/. /var/www/isms/wolfisms/uploads/
chown -R www-data:www-data /var/www/isms/wolfisms/uploads/
# Anwendungscode, config.py, secret.env und Lizenzdatei sind NICHT Teil dieses
# Archivs und muessen ggf. separat wiederhergestellt werden
systemctl restart apache2
```

© 2012–2026 WolfInfoSec GmbH

6. Lizenzverwaltung

WolfISMS prüft beim Start eine kryptografisch signierte Lizenzdatei. Der Flask Secret Key wird direkt aus der Lizenz-ID abgeleitet – eine gültige Lizenz ist daher für den Produktivbetrieb zwingend erforderlich. Ohne Lizenz läuft das System im Demo-Modus mit maximal einem aktiven Benutzer.

Komponente	Pfad / Beschreibung
Prüfbinary	/var/www/isms/bin/license_check
Öffentlicher Schlüssel	/var/www/isms/license/license_public.pem
Lizenzdatei	/var/www/isms/license/customer.lic
Lizenzfelder	Customer, Address, Tier, Expires, LicenseID
Secret-Key	Zufällig erzeugt und dauerhaft: ISMS_SECRET_KEY aus der Umgebung oder, wenn leer, die beim ersten Start angelegte Datei .flask_secret_key im Session-Verzeichnis (0600; isms_helpers.app_geheimnis()). Nicht aus der Lizenz abgeleitet – ein Lizenzwechsel lässt Sitzungen gültig
Demo-Modus	Keine Lizenz → max. 1 aktiver Benutzer, Warnmeldung im Log
Upload über UI	/admin/lizenz_upload – neue .lic-Datei hochladen und sofort aktivieren

Nach einem Lizenztausch muss Apache neu gestartet werden: systemctl restart apache2
Die Lizenzprüfung ist gecacht (lru_cache) – ohne Neustart bleibt die alte Lizenz aktiv. Ein Lizenztausch ändert den Secret Key und invalidiert alle bestehenden Benutzersessions.

6.1 Verhalten bei abgelaufener Lizenz

Läuft die Lizenz ab, bleibt das System lesbar, aber schreibgeschützt. Die Sichtbarkeit der Module ändert sich nicht – nur Schreiboperationen werden gesperrt (isms_helpers.eff_readonly()).

- **Normale Benutzer**: arbeiten nur noch lesend; Speichern- und Bearbeiten-Schaltflächen sind gesperrt
- **Administratoren und Masteruser**: behalten Schreibzugriff, um eine neue Lizenz einzuspielen und das System zu administrieren (isms.has_master_privileges())
- **Sicherheitsnetz**: ist kein Benutzer als Master gekennzeichnet, gilt der älteste Benutzer der Datenbank als solcher – damit sperrt sich niemand dauerhaft aus

6.2 Editionen und Funktionsumfang

Das Feld Tier der customer.lic steuert drei Dinge: wie viele schreibende Benutzer angelegt werden dürfen, welche Fachmodule zur Verfügung stehen und ob Single Sign-On und REST-API freigeschaltet sind. Es gibt zwei Editionen, Essentials und Advanced, jeweils für bis zu 5, 10 oder 25 schreibende Benutzer, dazu Enterprise für mehr als 25 Benutzer oder mehrere Installationen im Konzern. Die Nutzerzahl steht im Kürzel (ADV10 = Advanced für bis zu zehn Benutzer); ein angehängtes „+“ schaltet das Zusatzpaket SSO & REST-API frei (ADV10+). Ohne gültige Lizenzdatei gilt die Installation als DEMO, die freie Version.

© 2012–2026 WolfInfoSec GmbH

Edition (Tier)	Schreibende Benutzer	Fachmodule	SSO, REST-API	KI-Funktionen
DEMO (Free)	max. 1	Kernmodule	gesperrt	gesperrt
Essentials ESS5, ESS10, ESS25	max. 5, 10 bzw. 25	Kernmodule	gesperrt	verfügbar
Essentials Plus ESS5+, ESS10+, ESS25+	max. 5, 10 bzw. 25	Kernmodule	verfügbar	verfügbar
Advanced ADV5, ADV10, ADV25	max. 5, 10 bzw. 25	alle	gesperrt	verfügbar
Advanced Plus ADV5+, ADV10+, ADV25+	max. 5, 10 bzw. 25	alle	verfügbar	verfügbar
Enterprise ENT	unbegrenzt	alle	verfügbar	verfügbar

Gezählt werden nur schreibende Benutzer, also aktive Konten ohne das Kennzeichen readonly (Portal Only). Nur lesende Benutzer und damit sämtliche Portalnutzer zählen in keiner Edition mit; ihre Zahl ist überall unbegrenzt. Wie und wo die Grenze geprüft wird, steht im Abschnitt Nutzergrenze der Editionen. In der freien Version kommt zusätzlich eine Mengengrenze für Assets, Risiken, Dokumente und Lieferanten hinzu – siehe den folgenden Abschnitt.

Zu den Kernmodulen und damit in jede Edition gehören Dokumente samt Dokumentenlenkung, Policy Portal, Assets, Risiken, Maßnahmen, Changemanagement, Aufgaben, Lieferanten, Incidents, Controls samt SoA und Datenschutz, dazu die Strukturkarten (Organigramm, Prozesslandkarte, Netzwerkzonen, Pläne & Register) und die Links. Die freie Version hat denselben Modulumfang wie Essentials. Die KI-Assistenten sind in jeder bezahlten Edition enthalten.

Dreizehn Fachmodule gehören zu Advanced und damit auch zu Enterprise:

- **Ziele:** /ziele – Unternehmens- und Sicherheitsziele mit KPIs
- **Personal:** /personal samt On-/Offboarding und Schulungen
- **Kontakte:** /kontakte – Adressbuch und Notfallkontakte
- **Berichte:** /berichtsuebersicht – die Einzelberichte bleiben in ihren Modulen erreichbar
- **Projekte:** /projekte – teilt sich das Recht mass
- **Operational Plan:** /op_plan – Fristenkalender für wiederkehrende Pflichtaufgaben, nur für Masteruser
- **Audit:** /audit – interne Audits, Prüflisten, Abweichungen
- **Managementbewertung:** /managementreview – teilt sich das Recht audit
- **Schwachstellen:** /schwachstellen samt KI-Abgleich gegen bekannte CVE und den Endpunkten /api/v1/schwachstellen
- **Rechtsregister:** /rechtsregister – teilt sich das Recht norm
- **Playbooks:** /playbooks samt der Anhang- und Link-Routen /play_upload, /play_attachment und /play_attachment_delete
- **BCM:** /bcm – BIA je Asset, Notfallpläne, Übungen
- **Zugriffsmatrix:** /zugriffsmatrix samt Rollenkatalog der Matrix; Rollen selbst bleiben unter /admin/rollen in jeder Edition pflegbar

© 2012–2026 WolfInfoSec GmbH

In Free und Essentials fehlen diese Menüpunkte, und der direkte Aufruf einer ihrer URLs endet auf einer Hinweisseite. Die Maßnahmenansicht blendet die Verknüpfung zur Managementbewertung aus, die Erinnerungsmails führen keine Operational-Plan-Termine auf, und die Anhang- und Link-Routen der Playbooks und BCM-Pläne in isms.py antworten mit 403 (isms._require_module_access mit der Zuordnung _FLAG_MODUL). Erfasste Daten bleiben erhalten und sind nach einem Lizenzwechsel unverändert da.

Das Zusatzpaket „+“ umfasst zwei Funktionen:

Funktion	Betroffene Bereiche	Verhalten ohne „+“
REST-API	Alle Endpunktgruppen unter /api/v1/ und die Verwaltung der API-Keys unter /admin/api_keys (Kapitel REST-API)	Jeder Aufruf endet mit 403 und einem JSON-Fehlerobjekt, noch bevor der API-Key geprüft wird; das Feld tier_required nennt alle erlaubten Stufen. Vergebene Schlüssel bleiben gespeichert
Single Sign-On	Anmeldung über Microsoft Entra ID und Google, die Maske /admin/sso (Kapitel Single Sign-On)	Schalter und Zugangsdaten werden beim Lesen neutralisiert: die Anmeldeseite zeigt keine SSO-Schaltfläche, die Rückwege fallen auf die Passwortanmeldung zurück. Die Werte in sso_settings bleiben erhalten

Das „+“ ist eine Auswahl, keine Schwelle.
Advanced enthält alle Fachmodule, SSO und REST-API aber nur mit „+“ – und Essentials Plus hat SSO und REST-API, aber nicht die Advanced-Module. Eine Rangprüfung kann das nicht ausdrücken. Im Code steht dafür die Aufzählung isms_helpers.PLUS_TIERS (alle Stufen mit „+“ und ENT), aus der sich API_TIERS und admin_sso.SSO_TIERS bedienen.

Was nie an der Edition hängt.
Die Kernmodule stehen in jeder Edition zur Verfügung – damit lässt sich das ISMS betreiben. Für die Zertifizierung und den Nachweis gegenüber einem Auditor – Audit, Managementbewertung, BCM, Rechtsregister, Ziele und Kennzahlen – ist Advanced vorgesehen.

Die Stufe liest isms_helpers.lizenz_tier(). Innerhalb eines Requests hält flask.g das Ergebnis fest, außerhalb (Cron-Skripte) wird es 60 Sekunden zwischengespeichert – jede Prüfung startet sonst das license_check-Binary. Geprüft wird auf zwei Arten: tier_ok(min_tier) vergleicht die Stufe mit einer Mindeststufe aus der Rangfolge LIZENZ_TIER_RANK (DEMO, Essentials, Essentials Plus, Advanced, Advanced Plus, Enterprise – die Nutzergröße spielt dabei keine Rolle), tier_ok_an(anforderung) prüft gegen eine Aufzählung erlaubter Stufen. tier_stufen() bringt beide Schreibweisen auf ein Tupel, tier_namen() formuliert daraus den Text für Hinweisseite und Fehlerobjekt und fasst dabei die drei Nutzergrößen zusammen („Advanced (ADV5/10/25) oder höher“); tier_beispiel() schlägt auf der Hinweisseite eine passende Stufe ohne Rückschritt vor. Das Feld tier in isms_helpers.MODULE_REGISTRY nimmt beides auf: die Mindeststufe von Advanced bei Projekten, Operational Plan, Audit, Managementbewertung, Schwachstellen, Rechtsregister, Playbooks und BCM, die Aufzählungen NICHT_ESS_TIERS bei Ziele, Personal, Kontakte und Berichte und ZUGRIFF_TIERS bei der Zugriffsmatrix. Die Nutzergröße steht in USER_LIMIT_TIER.

© 2012–2026 WolfInfoSec GmbH

Gesperrt wird an drei Sorten von Gates: isms_helpers.modul_gate() – der before_request-Hook isms.enforce_module() – prüft zentral für jedes Blueprint eines Moduls zuerst die Edition und danach den Schalter des Administrators; isms_helpers.tier_gate() dient als zweite Schranke im Blueprint (Zugriffsmatrix, /admin/sso) beziehungsweise direkt in einer einzelnen Route (/admin/api_keys) und rendert die Hinweisseite; isms_helpers.api_tier_gate() sitzt vor den REST-Blueprints, antwortet als JSON und prüft unabhängig von der Anmeldung, weil die API nur den X-API-Key kennt. Für Routen außerhalb der Modul-Blueprints prüft isms._require_module_access() über modul_aktiv(). Die KI-Funktionen gehen ihren eigenen Weg über admin_ki_modul.ki_lizenz_ok() – siehe Kapitel KI-Einstellungen.

6.3 Mengengrenzen der freien Version

Ohne gültige Lizenzdatei läuft die Installation als DEMO. Seit v3.75 ist sie nicht nur in der Benutzerzahl begrenzt, sondern auch in der Datenmenge: je zehn Assets, Risiken, Dokumente und Lieferanten. In jeder bezahlten Edition gibt es keine Obergrenze.

Bereich	Tabelle	Grenze in DEMO	Gezählt wird
Assets	Assets	10	Zeilen der Tabelle
Risiken	Risiko	10	Zeilen der Tabelle
Dokumente	Dokumente	10	verschiedene Dokumentennummern
Lieferanten	vendor	10	Zeilen der Tabelle

Gezählt wird der Bestand in der Datenbank, nicht die Zahl der Neuanlagen: ein gelöschtes Objekt gibt seinen Platz wieder frei. Bei Dokumenten zählt die Dokumentennummer und nicht die Zeile – eine weitere Version desselben Dokuments legt zwar eine neue Zeile in Dokumente an, verbraucht aber nichts vom Kontingent.

Geprüft wird an jeder Stelle, die ein neues Objekt anlegt:

- **Masken**: die Maske für einen neuen Eintrag lässt sich gar nicht erst öffnen, und das Speichern eines neuen Datensatzes wird abgewiesen (assets_modul.py, risiken.py, dokumente_modul.py, lieferanten.py)
- **Import**: der Assets-Import legt nur so viele neue Zeilen an, wie das Kontingent noch hergibt, und meldet die übersprungenen im Ergebnis
- **REST-API**: POST auf /api/v1/assets, /api/v1/risiken und /api/v1/lieferanten antwortet mit HTTP 403 und der Meldung im Feld error
- **Erzeugte Dokumente**: Risikobehandlungsplan, Statement of Applicability und NIS2-Betroffenheitsprüfung legen Nachweisdokumente an und zählen deshalb mit; eine neue Version eines vorhandenen Dokuments bleibt frei

Bestehende Daten bleiben unangetastet.
Die Grenze sperrt ausschließlich das Anlegen. Ansehen, Ändern, Exportieren und Löschen bleiben in jeder Edition möglich – auch dann, wenn eine Installation nach dem Ablauf oder Tausch der Lizenz über der Grenze liegt. Wer wieder anlegen will, löscht einen vorhandenen Eintrag oder spielt eine bezahlte Lizenz ein.

Grenzen und Prüfung stehen in isms_helpers.py: FREE_LIMIT_ANZAHL gibt die Zahl vor, LIMIT_BEREICHE ordnet jedem Bereich Tabelle und Zählspalte zu, limit_erreicht() prüft und limit_meldung() liefert den Text in beiden Sprachen. Schlägt die Zählung fehl – etwa weil die

© 2012–2026 WolfInfoSec GmbH

Datenbank gerade nicht erreichbar ist –, gilt die Grenze als nicht erreicht: eine kaputte Prüfung darf die Anwendung nicht blockieren.

Wie weit eine Installation ist, zeigt /admin/lizenz_upload: in der freien Version steht unter der Lizenzangabe für jeden der vier Bereiche der Stand (etwa 7 / 10), ein erschöpfter Bereich rot hervorgehoben. In den bezahlten Editionen entfällt die Anzeige, weil es nichts zu begrenzen gibt.

6.4 Nutzergrenze der Editionen

Free lässt einen schreibenden Benutzer zu, Essentials und Advanced so viele, wie das Kürzel nennt – 5, 10 oder 25; Enterprise ist unbegrenzt. Das „+“ hebt also nicht die Benutzerzahl, sondern schaltet SSO und REST-API frei: ADV10 und ADV10+ lassen beide zehn schreibende Benutzer zu. Gezählt wird in der Tabelle users mit aktiv=1 und readonly=0. Ein Konto mit dem Kennzeichen „Portal Only" und ein deaktiviertes Konto zählen also nicht mit – ein Unternehmen kann in jeder Edition beliebig viele Mitarbeiter am Policy Portal teilnehmen lassen.

Geprüft wird an den drei Stellen, die die Zahl erhöhen:

Weg	Prüfung
/admin/create_user	Ein neuer Benutzer ohne „Portal Only" wird abgewiesen, wenn die
	Grenze erreicht ist
/admin/change_user	Nur, wenn der Benutzer bisher nicht als schreibender Benutzer zählte und es jetzt werden soll. Das Bearbeiten eines bereits schreibenden Benutzers läuft immer durch
/admin/toggle_user_active	Beim Reaktivieren eines Kontos ohne „Portal Only". Das Deaktivieren ist nie betroffen

Nicht geprüft werden der Benutzerimport unter /admin/import_users und die Selbstanlage beim ersten SSO-Login: beide legen ausschließlich readonly-Benutzer an und erhöhen die Zahl damit nicht.

Gesperrt wird nur der Zuwachs.
Liegt eine Installation nach einem Lizenzwechsel über der Grenze, bleibt sie vollständig bedienbar: alle vorhandenen Benutzer melden sich weiter an und lassen sich weiter bearbeiten. Abgewiesen wird erst der nächste zusätzliche schreibende Benutzer. Das ist dieselbe Regel wie bei den Mengengrenzen der freien Version.

Die Zahlen stehen an einer einzigen Stelle: isms_helpers.USER_LIMIT_TIER. Dazu gehören user_limit() für die Grenze der aktuellen Edition, user_anzahl() für den Bestand, user_zaehlt() für die Frage, ob ein bestimmter Benutzer heute schon mitzählt, user_limit_erreicht() für die Entscheidung und user_limit_meldung() für den Text in der Maske. isms.check_lizenz() liest dieselben Werte und dient als Gesamtschau. Schlägt die Zählung fehl – etwa weil die Datenbank nicht erreichbar ist –, gilt die Grenze als nicht erreicht: eine kaputte Prüfung darf niemanden aussperren.

© 2012–2026 WolfInfoSec GmbH

7. Modulverwaltung

Unter /admin/module legt der Administrator fest, welche Fachmodule die Installation anbietet und in welcher Reihenfolge sie innerhalb ihrer Menügruppe stehen (admin_modulverwaltung.py). Die Seite ist Administratoren vorbehalten und über die Kachel „Module" im Adminbereich erreichbar.

7.1 Registry und Speicherung

Alle Module stehen mit Menübild, Ziel-URL, Beschriftung (deutsch und englisch), Modulrecht, geforderter Edition und den zugehörigen Blueprints in isms_helpers.MODULE_REGISTRY – das ist die einzige Stelle, an der ein Modul eingetragen wird. Das Hauptmenü, die URL-Sperre und diese Verwaltungsseite bedienen sich daraus.

Die Einstellungen des Administrators liegen in der Tabelle isms_module (modul, aktiv, sortierung). Sie legt sich beim ersten Aufruf selbst an; ein Migrationsskript gibt es wie überall im Projekt nicht. Die Sortierwerte werden in Zehnerschritten vergeben, damit sich zwischen zwei Positionen notfalls noch etwas einsortieren lässt.

Nicht jedes Modul der Registry hat einen Menüpunkt: Organigramm sowie Pläne & Register werden ausschließlich aus dem Adminbereich heraus aufgerufen und stehen nur wegen der URL-Sperre und dieser Verwaltungsseite in der Registry. Für alle übrigen gilt: Ein Menüpunkt erscheint, wenn drei Bedingungen zugleich erfüllt sind:

- **Schalter**: der Administrator hat das Modul aktiviert gelassen (Feld aktiv)
- **Edition**: die Lizenzstufe reicht aus (Feld tier). Das Feld nimmt eine Mindeststufe oder eine Aufzählung erlaubter Stufen auf – daran hängen die dreizehn Advanced-Module (siehe Kapitel Lizenzverwaltung, Abschnitt „Editionen und Funktionsumfang")
- **Recht**: der angemeldete Benutzer besitzt das Modulrecht (Feld recht)

Eine kaputte Verwaltung darf das Menü nicht leeren.
Fehlt die Tabelle isms_module, ist sie leer oder kennt sie ein Modul noch nicht, gilt das Modul als aktiv und behält seine Position aus der Registry. Ein neu ausgeliefertes Modul erscheint deshalb ohne Zutun des Administrators.

Seit v3.90 steht auch die Berichtsübersicht (Schlüssel berichte) mit recht None in der Registry. Ihr Menüpunkt erscheint für alle schreibenden Benutzer; welche Karten sie zeigt, entscheidet die Seite je Modul nach denselben drei Bedingungen – siehe Kapitel Berichte und Meldewesen.

SoA: Seit v3.93 hat die Erklärung zur Anwendbarkeit einen eigenen Eintrag (Schlüssel soa, Ziel /berichte?berdetail=2, Recht norm, ohne eigenen Blueprint – die Route liegt in bp_normen). Er lässt sich unter /admin/module wie jedes Modul abschalten; die Seite prüft das selbst und antwortet dann mit 404. menu() zeigt den Punkt nur, wenn iso27001_installiert() einen nicht gelöschten Eintrag findet, dessen Framework mit „ISO/IEC 27001:“ beginnt (eine zusätzliche Abfrage je Seitenaufruf, zwischengespeichert); die SoA-Seite selbst wertet weiterhin ISO/IEC 27001:2022 aus

Mehrere Fassungen: seit v4.00 zeigt die SoA jede eingespielte Fassung von ISO/IEC 27001 (etwa 2022 und 2024), standardmäßig die neueste; ein Umschalter im Kopf wechselt, ?fassung= wählt direkt.

© 2012–2026 WolfInfoSec GmbH

7.2 Menügruppen

Seit v3.91 gliedert isms.menu() die Seitenleiste in Gruppen. Sie stehen mit Schlüssel, deutscher und englischer Überschrift und ihren Modulschlüsseln in isms_helpers.MENU_GRUPPEN; MODUL_GRUPPE bildet daraus Modul → Gruppe ab.

Gruppe	Module (Standardreihenfolge)
Mein Arbeitsbereich	Policy Portal, Aufgaben
Governance & Compliance	Controls, SoA, Rechtsregister, Datenschutz, Dokumente, Ziele
Organisation & Inventar	Assets, Zugriffsmatrix, Lieferanten, Personal, Prozesslandkarte, Netzwerkzonen
Risiko & Maßnahmen	Risiken, Maßnahmen, Schwachstellen
Betrieb & Resilienz	Changemanagement, Kontakte, Incidents, Playbooks, BCM
Planung & Überprüfung	Projekte, Operational Plan, Audit, Managementbewertung, Berichte
(ohne Überschrift)	Links – abgesetzt durch eine Linie

- **Reihenfolge**: modul_liste() sortiert zuerst nach der Gruppe, darin nach dem gespeicherten Sortierwert aus isms_module. Die Gruppen selbst lassen sich nicht umstellen; eine bestehende Einstellung bleibt innerhalb jeder Gruppe erhalten
- **Leere Gruppen**: Eine Gruppe ohne sichtbaren Eintrag – abgeschaltet, nicht lizenziert oder ohne Modulrecht – entfällt samt Überschrift. Bleibt nur eine Gruppe übrig, etwa beim readonly-Benutzer mit dem Policy Portal, zeichnet das Menü ganz ohne Überschriften
- **Aktueller Eintrag**: modul_von_blueprint() ermittelt das Modul der Anfrage über den Blueprint, ersatzweise über den Parameter active von menu(). Sein Eintrag ist hervorgehoben, seine Gruppe immer geöffnet
- **Zugeklappte Gruppen**: merkt sich der Browser im localStorage unter wolfisms_menu_zu, je Benutzer und Browser. Auf dem Server wird nichts gespeichert
- **Menü-Icons**: Seit v4.00 steht vor jedem Eintrag eine farbige Symbolkachel mit weißem Linien-Icon statt eines PNG-Bildes. isms.MENU_ICON ordnet Modul → Icon zu, die Farbe kommt aus der Menügruppe (CSS-Klasse .mi-<gruppe>), die Ausgabe übernimmt isms.menu_eintrag(). Der Grund der Leiste folgt dem Farbschema; die aktive Seite ist eine Pille in der Akzentfarbe, in der eingeklappten Leiste ein Ring um die Kachel
- **Kompakte Darstellung**: Ein Knopf oben in der Seitenleiste, links neben «, rückt die Einträge enger und verkleinert Symbolkacheln und Icons. Der Browser merkt sich die Wahl im localStorage wie das Einklappen – je Browser, nicht je Konto

Ein neues Modul gehört in genau eine Gruppe.
Fehlt sein Schlüssel in MENU_GRUPPEN, landet es in der letzten Gruppe ohne Überschrift bei Links. Außerdem braucht es einen Eintrag in isms.MENU_ICON; der Icon-Name muss in ui_icons._SVG stehen, sonst erscheint ein gefülltes Kästchen.

7.3 Bedienung

- **Ein- und Ausschalten**: ein Häkchen je Zeile; „Speichern" schreibt den Stand aller Zeilen in einem Zug
- **Reihenfolge**: Zeile ziehen (Drag & Drop) oder die Pfeiltasten der Zeile verwenden. Gespeichert wird die Reihenfolge der Liste, nicht die einzelne Bewegung. Sie wirkt nur innerhalb einer

© 2012–2026 WolfInfoSec GmbH

Menügruppe; die Gruppe steht in jeder Zeile vor der Ziel-URL, und eine über die Gruppengrenze gezogene Zeile springt nach dem Speichern in ihre Gruppe zurück
- **Zurücksetzen**: aktiviert alle Module und stellt die Reihenfolge der Registry wieder her
- **Gesperrte Zeilen**: Module oberhalb der aktuellen Lizenz-Edition stehen ausgegraut mit Schlosshinweis in der Liste. In Free und Essentials betrifft das die dreizehn Advanced-Module; diese Zeilen lassen sich weder einschalten noch aufrufen. Der Schlosshinweis nennt die Editionen, in denen das Modul enthalten ist
- **Pflichtmodule**: das Policy Portal trägt „immer aktiv" und lässt sich nicht abschalten – readonly-Benutzer werden dorthin umgeleitet, ein abgeschaltetes Portal wäre eine Endlosschleife. Auch ein direktes UPDATE in isms_module wird beim Lesen überstimmt

7.4 Was ein abgeschaltetes Modul bedeutet

- **Menü**: der Eintrag verschwindet für alle Benutzer, auch für Administratoren und Masteruser
- **Adressen**: isms_helpers.modul_gate() sperrt als before_request der App jede URL des Moduls über den Blueprint-Namen. Ein direkter Aufruf endet mit dem Status 403 und einer Hinweisseite
- **REST-API**: der zugehörige API-Bereich ist mitgesperrt und antwortet mit 403 und dem Fehlerobjekt {"success": false, "error": "… Dieses Modul wurde deaktiviert."} – der API-Key ist in Ordnung, das Modul nicht erreichbar
- **Daten**: die Tabellen werden weder gelöscht noch geleert. Nach dem Wiedereinschalten sind die Inhalte unverändert erreichbar
- **Wirkung**: die Einstellungen werden je Request in flask.g gehalten und sonst frisch gelesen – eine Änderung wirkt sofort, ein Apache-Neustart ist nicht nötig

Admin- und Anmeldebereich bleiben immer erreichbar.
Gesperrt werden ausschließlich die Blueprints der Fachmodule. Adminseiten, Anmeldung, Portal und die Aufgabenübersicht gehören zu keinem Modul der Registry und laufen deshalb immer durch – ein Administrator kann sich mit dieser Seite nicht selbst aussperren.

7.5 Seitenleiste der Detailansichten

Die Detailansichten der Module bauen ihre rechte Seitenleiste selbst als HTML zusammen. Seit v3.94 gelten dafür zwei gemeinsame Helfer in isms_helpers.py; neue Module sollten beide verwenden.

- **Aufbau**: oben Eigenschaften und Metadaten (Ersteller, erstellt, Änderer, geändert, gegebenenfalls Reviewer und Review-Datum), darunter je Bereich eine <h4>-Überschrift mit Anzahl und „Verwalten“ und die Einträge als Links. Die Trennlinie hängt an der Oberkante der nächsten Überschrift und schließt so den vorigen Abschnitt ab
- **sidebar_kuerzen(html, en)**: die fertige Leiste wird einmal durchgereicht. Abschnitte mit „Verwalten“-Link werden auf SIDEBAR_MAX = 5 Einträge gekürzt, dahinter folgt „mehr … (+N)“ auf dieselbe Adresse wie „Verwalten“. Abschnitte ohne diesen Link – Leserechte, Änderungsprotokoll, Versionen, Kommentare – bleiben vollständig. Eingesetzt in Dokumente, Incidents, Controls, Maßnahmen, Audit, Projekte, Risiken, Schwachstellen, Change Management, Assets und Lieferanten
- **rueck_abschnitte_html(quellen, spalte, ziel_ids, wp, en)**: Rückverweise. Verknüpfungen stehen als kommagetrennte ID-Liste nur beim Datensatz, der sie anlegt; der Helfer sucht in den Quelltabellen aus _RUECK_QUELLEN (change, vendor, schwachstellen, projekte, risiko, seit v3.95 auch audit für Risiken und bcm für Maßnahmen) nach den IDs des Ziels und gibt je Quelle

© 2012–2026 WolfInfoSec GmbH

einen Abschnitt aus – nur mit Einträgen und nur, wenn rueck_sichtbar() das Modul als aktiv und lesbar meldet. Eine neue Verknüpfungsart braucht einen Eintrag in _RUECK_QUELLEN

- **Aktionsleisten**: seit v3.95 ohne Verknüpfungs-Schaltflächen. Sie führten auf dieselben Verknüpfungsansichten wie „Verwalten“ in der Seitenleiste und sind in Dokumente, Assets, Lieferanten, Risiken, Maßnahmen und Incidents entfallen; die Routen selbst bleiben. Kopftabellen wiederholen nicht mehr, was in der Überschrift steht, Besitzer, Verantwortliche und Bereich/Abteilung stehen in den Metadaten der Seitenleiste. Die Aktionen zum Lieferantenfragebogen stecken in einem <details>-Aufklappmenü (ven-menu)

Ziel	Rückverweise aus
Assets	Elternassets (Feld der Tochterassets), Changes, Lieferanten, Schwachstellen
Risiken, Incidents	Changes, Schwachstellen
Maßnahmen	Projekte, Schwachstellen, BCM-Übungen
Dokumente	Risiko.dokumente, Changes – über alle Versionen der Reihe; Dokument↔Dokument siehe Kapitel Dokumentenlenkung und Versionen
BPMN	Dokumente.bpmn, als Liste „Verknüpfte Dokumente“ unter dem Diagramm; je Reihe eine Zeile, abgelöste Versionen zählen nicht

Eine Migration ist nicht nötig: am Speicherort ändert sich nichts. Die Controls zeigen seit v3.94 Dokumente, Maßnahmen, Risiken und Assets nur noch in der Seitenleiste; die Tabellen unter der Beschreibung samt „Alle anzeigen“ und ihre Hilfsfunktionen sind entfallen. Risiken haben keine Kopftabelle mehr, Nummer und Name stehen in der Überschrift.

7.6 Oberfläche seit v4.00

Mit v4.00 hat die Oberfläche ein ruhigeres, einheitliches Erscheinungsbild bekommen. Die Module bauen ihr HTML weiter selbst als f-String; vereinheitlicht wird an wenigen zentralen Stellen, wie schon beim Farbschema:

© 2012–2026 WolfInfoSec GmbH

Baustein	Ort	Wirkung
Linien-Icons	ui_icons.py, after_request-Hook apply_ui_icons in isms.py (läuft vor dem Farbschema)	Ein Emoji am Anfang von Knöpfen, Reitern, Überschriften h1–h4 und Symbol-Divs wird zu <i class=wi-NAME> – einer CSS-Maske in der Schriftfarbe. Neues Emoji: Eintrag in _EMOJI, neue Form: _SVG
Knöpfe und Reiter	templates/template.html	Übersteuert die Modulfarben: neutral grau mit 8 px Rundung bzw. als Kapsel. Gefüllt in der Akzentfarbe #2E5F8A sind nur der Hauptknopf (Klasse btn-haupt, automatisch beim Plus- und beim Diskettensymbol) und Absendeknöpfe von POST-Formularen; Gefahr-Knöpfe sind rot. Inline-Farben mit !important gewinnen weiterhin und gehören nicht mehr an Knöpfe
Listen	isms_ui.py: liste_kopf, liste_suche, liste_kacheln, liste_segment, liste_filter, liste_sortierung, liste_chips, liste_zeile, liste_rahmen, liste_zeile_form	Alle Modul- und Admin-Listen. Die Suche filtert im Browser (?search= setzt nur den Anfangswert), jedes Wort muss vorkommen; das Blättern ist entfallen. Nebenaktionen wie Export stehen im Drei-Punkte-Menü
Titelkachel	isms_ui.titel_kachel()	Haupt- und Detailansichten, SoA samt Einzelberichten und das Policy Portal zeigen ihre Überschrift in einer weißen Kachel
Fußzeile	templates/template.html	Rechts „WolfISMS Version <Version>“ (isms.Version)
Hell/Dunkel	theme_modul, /design/modus	Knopf neben der Sprachflagge, Wahl je Benutzer – siehe Kapitel Design und Farbschema

- **Bilder:** images/ enthält seit v4.00 nur noch elf Dateien – Flaggen, Login-Knöpfe, Lesebestätigung, About-Bild, Admin-Kachel und den Branding-Ersatz. Das Feld bild in MODULE_REGISTRY ist entfallen, hover_button() erwartet einen Icon-Namen. Neue Symbole gehören als Icon nach ui_icons._SVG, nicht als Bild nach images/
- **Neue Module:** die Liste aus den liste_*-Bausteinen bauen; Löschen in einer Zeile als kleines POST-Formular (Klasse wl-zeile-knopf), Pflegetabellen mit Eingabefeldern als liste_zeile_form. Eine Kachelreihe mit Kennzahlen gehört in theme_modul.INFOKACHEL_KLASSEN, damit sie sich abschalten lässt
- **Icon-Namen:** vor der Verwendung in ui_icons._SVG nachsehen – ein unbekannter Name ergibt ein gefülltes Kästchen

© 2012–2026 WolfInfoSec GmbH

8. REST-API

WolfISMS stellt eine REST-API für sechs Bereiche bereit: Incidents (Sicherheitsvorfälle), Maßnahmen, Assets, Risiken, Lieferanten und Schwachstellen. Sie ist für die maschinelle Anbindung gedacht – ein SIEM meldet Vorfälle, ein Ticketsystem holt offene Maßnahmen ab, eine CMDB gleicht den Assetbestand ab, ein Einkaufssystem meldet Lieferanten, ein Schwachstellenscanner liefert seine Funde ein, ein Reporting-Werkzeug liest Risiken. Authentifiziert wird über einen API-Key im HTTP-Header, nicht über CSRF-Tokens.

Lizenzvoraussetzung: Die REST-API gehört zum Zusatzpaket „+“ – also zu Essentials Plus und Advanced Plus, erkennbar am „+“ im Kürzel – und zu Enterprise (ENT). In allen übrigen Editionen sind die Endpunkte und die Verwaltung der API-Keys gesperrt; jeder Aufruf endet mit 403 und einem JSON-Fehlerobjekt, noch bevor der API-Key geprüft wird. Das Feld tier_required im Fehlerobjekt nennt seit v3.85 alle erlaubten Stufen als Liste. Bereits vergebene Schlüssel bleiben gespeichert und gelten wieder, sobald eine passende Lizenz eingespielt ist.

Dieses Kapitel beschreibt die administrative Seite: welche Endpunkte es gibt, wie sie abgesichert sind und wie Schlüssel verwaltet werden. Die vollständige Schnittstellenbeschreibung mit allen Feldern, Wertelisten, Filtern und Beispielen steht im eigenen Dokument „WolfISMS REST-API", das für die Entwickler der anbindenden Systeme gedacht ist.

8.1 Bereiche

Bereich	Basis-URL	Modul	Besonderheit
Incidents	/api/v1/incidents	api_incidents.py	Fehlende Spalten (Ticketnummer, sync/synctime, _en-Felder) legt das Modul beim Laden selbst an
Maßnahmen	/api/v1/massnahmen	api_massnahmen.py	Verknüpfungen zu Risiken und Assets sind nur lesbar
Assets	/api/v1/assets	api_assets.py	Zulässige Werte (Kategorie, Standort, Klassifizierung …) kommen aus der Datenbank und sind über /werte abfragbar
Risiken	/api/v1/risiken	api_risiken.py	Die Bewertung wird aus Auswirkung und Wahrscheinlichkeit berechnet und nicht übergeben. Seit v3.72 gehört das Freitextfeld chancen (Spalte Chancen) zu den schreib- und lesbaren Feldern; die Spalte legt das Modul beim Laden selbst an

© 2012–2026 WolfInfoSec GmbH

Bereich	Basis-URL	Modul	Besonderheit
Lieferanten	/api/v1/lieferanten	api_lieferanten.py	Kriterien werden als Worte übergeben; der Score wird daraus berechnet
Schwachstellen	/api/v1/schwachstellen	api_schwachstellen. py	Fehlt die Nummer beim Anlegen, vergibt die API sie; der Schweregrad wird aus dem CVSS-Score abgeleitet

8.2 Endpunkte

Alle sechs Bereiche sind gleich geschnitten. <bereich> steht für incidents, massnahmen, assets, risiken, lieferanten oder schwachstellen.

Methode	URL	Funktion
POST	/api/v1/<bereich>	Datensatz anlegen; die Nummer muss eindeutig sein. Nur bei den Schwachstellen darf sie fehlen – dann vergibt die API die nächste freie Nummer WEA-NNNNN
GET	/api/v1/<bereich>	Liste mit Filtern, Freitextsuche, Sortierung und Paginierung
GET	/api/v1/<bereich>/<nummer>	Einzelnen Datensatz abrufen
PATCH / PUT	/api/v1/<bereich>/<nummer>	Synchronisationsstand schreiben (sync und synctime); PUT verhält sich identisch zu PATCH
GET	/api/v1/<bereich>/werte	Zulässige Werte laut Stammdaten – nur für assets, risiken, lieferanten und schwachstellen

PATCH ändert nur wenige Felder.
Jedes andere Feld im Body wird mit HTTP 400 abgewiesen, statt still ignoriert zu werden – ein Tippfehler im aufrufenden System fällt damit sofort auf. Bei Assets, Risiken und Lieferanten bleibt es bei sync, synctime und weitere_besitzer, bei Incidents kommt status hinzu, bei Maßnahmen die Ticketnummer und weitere_besitzer. Nur die Schwachstellen nehmen zusätzlich status, score, besitzer und weitere_besitzer entgegen: der Behandlungsstand wandert dort oft aus dem meldenden System zurück. Alles Übrige wird in der Anwendung gepflegt.

8.3 Authentifizierung

Jeder Aufruf muss den API-Key im HTTP-Header X-API-Key übermitteln. In der Datenbank liegt nur der SHA-256-Hash; den Klartext zeigt die Anwendung genau einmal bei der Erstellung an.

Eigenschaft	Details
Header-Name	X-API-Key
Key-Format	64 Hex-Zeichen (32 Bytes, secrets.token_hex(32))

© 2012–2026 WolfInfoSec GmbH

Eigenschaft	Details
Speicherung	Nur SHA-256-Hash in Tabelle api_keys – Klartext nicht rekonstruierbar
Verwaltung	Über Admin-UI: /admin/api_keys (anlegen, aktivieren, deaktivieren, löschen)
Sperren	aktiv=0 wirkt sofort – der schnellste Weg, eine Anbindung stillzulegen
CSRF	Kein CSRF-Token erforderlich – die API-Blueprints sind CSRF-exempt
Rechte	Ein Key hat keine Benutzerrolle: er darf in allen sechs Bereichen lesen und anlegen

8.4 Kein Rate-Limit auf der API

Bis 3.51 galten für die Incidents-API 30 Anfragen je Stunde für POST und 120 für GET. Diese Grenzen sind entfallen: ein Sync-Client holt und quittiert Datensätze im Sekundentakt und liefe sonst dauerhaft gegen HTTP 429.

```
# isms.py – die API-Blueprints werden aus der Zaehlung genommen
limiter.exempt(bp_api_incidents)
limiter.exempt(bp_api_massnahmen)
limiter.exempt(bp_api_assets)
limiter.exempt(bp_api_risiken)
limiter.exempt(bp_api_lieferanten)
limiter.exempt(bp_api_schwachstellen)
```

limiter.exempt() statt eines hohen limit()-Wertes: sonst griffen zusätzlich die globalen Vorgaben von 2000 Anfragen je Tag und 500 je Stunde. Der Zugang hängt damit allein am API-Key – ein Grund mehr, je anbindendem System einen eigenen Schlüssel zu vergeben und nicht mehr benötigte Schlüssel zu löschen.

8.5 Synchronisationsstand

Jeder der sechs Bereiche führt die Spalten sync und synctime. Damit merkt sich nicht das anbindende System, was es schon übertragen hat, sondern WolfISMS – ein Client, der neu aufgesetzt wird, verschickt deshalb keine Altlasten erneut.

Wert	Bedeutung
sync = null	Dieser Datensatz ist nicht für die Synchronisation vorgesehen
sync = 0	Noch nicht synchronisiert – dies ist die Arbeitsmenge des Clients
sync = 1	Synchronisiert
synctime	Zeitpunkt der letzten Synchronisation; wird beim Setzen von sync=1 automatisch auf jetzt gesetzt, wenn er nicht mitgegeben wird

```
# 1. Offene Datensaetze holen
curl -s "https://host/isms/api/v1/massnahmen?sync=0&limit=100" \
     -H "X-API-Key: $KEY"

# 2. Nach dem Uebertragen quittieren
curl -X PATCH "https://host/isms/api/v1/massnahmen/MAS-0042" \
```

© 2012–2026 WolfInfoSec GmbH

```
    -H "X-API-Key: $KEY" -H "Content-Type: application/json" \
    -d '{"sync": 1}'
```

8.6 API-Key verwalten

1. Admin-Bereich öffnen: /admin → "API-Keys"

2. Beschreibung eingeben (z. B. "SIEM-Integration") und "API-Key generieren" klicken

3. Den angezeigten Klartext-Key sofort kopieren – er wird nur einmalig angezeigt

4. Keys können jederzeit deaktiviert (gesperrt) oder endgültig gelöscht werden

Sicherheitshinweis: API-Keys gewähren lesenden und schreibenden Zugriff auf alle fünf API-Bereiche. Verwenden Sie für jede externe Anwendung einen eigenen Key und deaktivieren Sie nicht mehr benötigte Keys umgehend.

8.7 Testen ohne REST-Konsole

Die REST-Konsole unter /admin/rest_console ist mit v4.00 entfallen, ebenso die SQL-Konsole unter /admin/sql_console – beide samt Code (admin_rest_console.py, admin_sql_console.py) und Kachel. Zum Testen der Endpunkte dient ein gewöhnlicher HTTP-Client wie curl; die REST-API-Dokumentation enthält zu jedem Bereich Beispielaufrufe. Abfragen direkt gegen die Datenbank laufen über die Werkzeuge des Datenbankservers, nicht mehr über die Anwendung.

© 2012–2026 WolfInfoSec GmbH

9. Daten SQL-Export/Import

Unter /admin/daten_sql steht ein Werkzeug zum Export und Import von Stammdaten zwischen WolfISMS-Instanzen zur Verfügung (admin_daten_sql.py), z. B. um Vorlagen von einer Test- auf eine Produktivinstanz zu übertragen. Das Modul löst das frühere admin_dokumente_sql.py ab und deckt nun drei Bereiche sowie ein Komplettpaket ab.

9.1 Bereiche

Bereich	Tabelle	Verhalten beim Import
Dokumente	Dokumente	Frischer Entwurf: Version/Subversion = 0, Status = Entwurf, Ersteller = importierender Benutzer, Erstelldatum = Importdatum; Prüf- und Freigabefelder werden geleert. Auch die Versionsreihe fängt neu an: stamm_id und vorgaenger_id der Quellinstanz werden verworfen und über die Dokumentennummer neu aufgebaut
Risiken	Risiko	Die Bewertung (Status, Wahrscheinlichkeit/Auswirkung vor und nach Maßnahmen, Behandlung, Risikoklasse, Score) wird 1:1 übernommen. Zurückgesetzt werden Zuständigkeits- und Zustandsfelder sowie die Verknüpfungsspalten (massnahmen, assets, bedrohungen, schwachstellen, dokumente, normen, lieferanten), da sie Fremd-IDs enthalten, die auf der Zielinstanz auf andere Objekte zeigen würden
Maßnahmen	Massnahmen	Inhalt (Text, Priorität, Aufwand, Kosten, Enddatum) wird übernommen, der Umsetzungsfortschritt der Quellinstanz dagegen zurückgesetzt: status = offen, erledigt = 0, archiviert = 0, Ticketnummer geleert
Alle (Komplettpake t)	alle drei	Exportiert alle drei Tabellen als eine SQL-Datei und importiert sie bereichsweise, jeder Block mit den Regeln seines Bereichs

9.2 Weitere Eigenschaften

- **Dateiformate**: Export als .sql zum Download; Import von .sql oder gepackt als .tar.gz/.tgz
- **Importpaket NIS2-Standardrisiken:** Seit v4.00 liegt import/nis2_standardrisiken.sql bei – 70 unbewertete Risiken (deutsch und englisch) zu Art. 20, 21 und 23 NIS2 bzw. §§ 30, 32 und 38 BSIG. Einspielen unter Daten SQL › Risiken › SQL-Datei importieren. Die Risiken kommen mit Status „offen“ und Behandlung „nicht bewertet“; Besitzer, Ersteller und Review übernimmt der importierende Benutzer
- **Neue IDs**: der Import fügt ausschließlich an – bestehende Datensätze bleiben unverändert
- **Doppelte Nummern**: existiert eine Nummer aus der Importdatei bereits, wird die nächste freie Nummer im gleichen Schema vergeben (z. B. RSK-0042 → RSK-0087) und im Protokoll ausgewiesen
- **Abteilungszuordnung**: bei Risiken und Maßnahmen wird die Abteilung über den Klartextnamen auf die lokale Abteilungen-Tabelle gemappt; findet sich kein passender Name, bleiben Abteilung und abteilung_id leer
- **Gemischte Dateien**: da die Blöcke nach Tabellen getrennt bleiben, lässt sich ein Komplettpaket auch auf einem Einzel-Tab einspielen (dann greift er nur „seine“ Tabelle heraus) und umgekehrt
- **Nicht enthalten**: Anhänge, Historien (dok_historie, Risiko_historie, Score-Historie) und Verknüpfungen werden nicht mit exportiert oder importiert – nur die Stammdaten selbst
- **Altrouten**: /admin/dokumente_sql und /admin/risiken_sql leiten weiterhin auf die neuen Bereiche um

© 2012–2026 WolfInfoSec GmbH

Sicherheitshinweis: Zugriff ausschließlich für current_user.admin (wie der Framework-Import). Da Besitzer-, Prüfer- und Freigeberfelder Benutzer-IDs referenzieren, die auf der Zielinstanz ggf. anderen Personen zugeordnet sind, sollten importierte Datensätze vor der Freigabe geprüft werden.

© 2012–2026 WolfInfoSec GmbH

10. Zwei-Faktor-Authentifizierung (2FA / TOTP)

WolfISMS unterstützt zeitbasierte Einmalpasswörter nach RFC 6238 (TOTP) als zweiten Authentifizierungsfaktor. Kompatible Apps sind unter anderem Google Authenticator und Microsoft Authenticator. Das Modul ist in totp_modul.py implementiert.

10.1 Datenbank-Migration

Vor dem ersten Einsatz müssen folgende SQL-Migrationen einmalig ausgeführt werden (in der aktuellen isms.sql bereits enthalten):

```
-- Spalten in der users-Tabelle
ALTER TABLE users ADD COLUMN totp_secret VARCHAR(64) DEFAULT NULL;
ALTER TABLE users ADD COLUMN totp_active TINYINT(1) NOT NULL DEFAULT 0;
```

```
-- totp_required-Flag (in Stammdaten oder isms_settings)
ALTER TABLE Stammdaten ADD COLUMN IF NOT EXISTS totp_required TINYINT(1) NOT NULL
DEFAULT 0;
```

```
-- Alternativ: isms_settings-Tabelle
CREATE TABLE IF NOT EXISTS isms_settings (
  `key` VARCHAR(64) PRIMARY KEY,
  `value` VARCHAR(255) NOT NULL DEFAULT ''
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

10.2 Routen und Ablauf

Route	Funktion
/2fa/setup	Benutzer richtet 2FA ein: QR-Code anzeigen, TOTP-Secret generieren (pyotp.random_base32), Verifikation mit dem ersten Code
/2fa/verify	TOTP-Code-Eingabe nach dem Login (wird nach der Passwortprüfung aufgerufen, wenn totp_active=1); gültiger Code = vollständiger Login
/2fa/disable	Benutzer deaktiviert 2FA für den eigenen Account (setzt totp_active=0, totp_secret=NULL)
/admin/totp	Administrator-Schalter: systemweites totp_required aktivieren/deaktivieren. Bei totp_required=1 werden Benutzer ohne aktives 2FA beim nächsten Login automatisch zu /2fa/setup umgeleitet

10.3 Sicherheitshinweise

- TOTP-Secrets werden in der Datenbank im Klartext gespeichert – Datenbankzugriff entsprechend absichern
- Der QR-Code wird als SVG inline im Browser gerendert – kein Server-Roundtrip der Rohdaten
- Pip-Voraussetzungen: pyotp und qrcode (bereits in Schritt 1 der Installation enthalten)
- Bei verlorener Authenticator-App setzt der Administrator 2FA für das Konto zurück; das löscht auch Fehlversuchszähler und Sperre. Ersatzweise direkt in der Datenbank: UPDATE users SET totp_secret=NULL, totp_active=0 WHERE id=<USER_ID>;

© 2012–2026 WolfInfoSec GmbH

10.4 Sperre nach Fehlversuchen

Das Raten eines sechsstelligen Codes war bis 3.51 unbegrenzt möglich. Seit 3.52 zählt WolfISMS die Fehlversuche je Konto in der Datenbank und sperrt die Code-Eingabe vorübergehend. Die Zählung liegt bewusst in der Datenbank und nicht im Prozessspeicher: bei mehreren WSGI-Workern gälte sie sonst je Prozess und wäre damit leicht zu umgehen.

Eigenschaft	Wert
Fehlversuche bis zur Sperre	5
Sperrdauer	15 Minuten
Spalten	users.totp_fehlversuche und users.totp_gesperrt_bis – legt das Modul beim Laden selbst an
Zusätzlich je IP-Adresse	60 POST je 15 Minuten; nur POST wird gezählt, damit das blosse Aufrufen der Seite kein Kontingent verbraucht
Aufheben	Der 2FA-Reset durch den Administrator löscht Zähler und Sperre mit

Ebenfalls seit 3.52 lässt sich eine erzwungene 2FA-Einrichtung (totp_required=1) tatsächlich abschliessen: /2fa/setup ermittelt das Konto aus der laufenden Anmeldung und meldet den Benutzer an, sobald der erste gültige Code eintrifft. Zuvor lief die Weiterleitung gegen @login_required und endete wieder auf der Anmeldeseite.

© 2012–2026 WolfInfoSec GmbH

11. Single Sign-On (Microsoft / Google)

WolfISMS kann die Anmeldung optional an Microsoft Entra ID (Azure AD) oder Google delegieren. Die Umsetzung erfolgt über OpenID Connect (authlib). Beide Verfahren lassen sich unabhängig voneinander aktivieren; die klassische Anmeldung mit E-Mail und Passwort bleibt daneben bestehen.

Lizenzvoraussetzung: Single Sign-On gehört zum Zusatzpaket „+“ (Essentials Plus, Advanced Plus) und zu Enterprise (ENT). In den übrigen Editionen ist die Maske /admin/sso gesperrt, und admin_sso.sso_config() neutralisiert Schalter und Zugangsdaten: die Anmeldeseite zeigt keine SSO-Schaltfläche, und die Rückwege /auth/microsoft und /auth/google fallen auf die Passwortanmeldung zurück. Eine bereits eingerichtete Anmeldung läuft nach einem Lizenzwechsel also nicht weiter; die Werte in sso_settings bleiben erhalten.

11.1 Aktivierung

Die Zugangsdaten stehen entweder in der Umgebung (.env / secret.env) oder in der Tabelle sso_settings, die unter /admin/sso gepflegt wird. Welche Quelle gilt, entscheidet ein einzelner Schalter am Kopf der Maske; admin_sso.sso_config() liefert die daraus gültigen Werte an alle Anmeldewege.

Schalter „Werte aus dieser Maske verwenden"	Wirkung
aus (Voreinstellung)	Es gelten die Werte aus .env / secret.env – unverändertes bisheriges Verhalten. Das Formular wird gespeichert, aber nicht benutzt.
ein	Die Zeile aus sso_settings überschreibt die Umgebung vollständig, einschließlich der Ein/Aus-Schalter MS_AUTH und GOOGLE_AUTH.

Der Schalter ist bewusst ein einzelner.
Nur so lässt sich ein in der Umgebung eingeschaltetes SSO über die Oberfläche auch wieder abschalten. Ein feldweises „leer = Umgebung" könnte das nicht abbilden und ließe nie erkennen, welcher Wert gerade gilt.

11.2 Einrichtung über die Oberfläche (/admin/sso)

Schritt	Vorgehen
1. Anwendung registrieren	Im Entra-Portal bzw. in der Google Cloud Console eine Anwendung anlegen. Die einzutragenden Redirect-URIs zeigt die Maske im Abschnitt „Redirect-URIs für die Anbieter-Konsolen" – sie leiten sich aus WEBPATH ab
2. Zugangsdaten eintragen	Microsoft: Directory-(Tenant-)ID, Application-(Client-)ID und Client-Secret. Google: Client-ID, Client-Secret und optional eine abweichende Redirect-URI
3. Domain-Allowlist setzen	Zugelassene E-Mail-Domains kommagetrennt, z. B. example.de, example.com. Leer heißt: niemand darf

© 2012–2026 WolfInfoSec GmbH

Schritt	Vorgehen
	sich per SSO anmelden
4. Verfahren einschalten	Häkchen „Microsoft-Anmeldung einschalten" und/oder „Google-Anmeldung einschalten", dazu den Schalter am Kopf der Maske
5. Prüfen	„Speichern & Discovery prüfen" holt das OIDC-Discovery-Dokument des Anbieters und meldet den Issuer oder den Fehler. Ein Neustart ist nicht nötig – der OAuth-Client wird beim Anmeldevorgang mit den dann gültigen Werten gebaut

- **Secrets**: werden seit v3.99 AES-256-GCM-verschlüsselt abgelegt – derselbe Weg wie beim KI-API-Key (Kapitel KI-Einstellungen), der Schlüssel stammt aus dem Secret-Key der Anwendung – und in der Maske nur maskiert angezeigt. Ein bis dahin Base64-kodierter Bestand wird beim ersten Lesen der Tabelle umgestellt. Ein leeres Feld behält das gespeicherte Secret, ein Häkchen „löschen" entfernt es
- **Tenant-ID**: GUID, verifizierte Domain oder common / organizations / consumers. Der Wert landet in der Discovery-URL und wird deshalb streng geprüft
- **Wirksame Werte**: am Fuß der Seite steht, welche Werte gerade gelten und aus welcher Quelle sie stammen
- **Unvollständig eingerichtet**: fehlt Tenant- oder Client-ID, führt /auth/microsoft bzw. /auth/google nicht ins Leere, sondern mit einer Meldung zurück auf den Notfallzugang

11.3 Einrichtung über die Umgebung

Schritt	Vorgehen
1. Anwendung registrieren	Im Entra-Portal bzw. in der Google Cloud Console eine Anwendung anlegen und die Callback-URL hinterlegen (https://host/auth/microsoft/callback bzw. https://host/isms/auth/google/callback)
2. Zugangsdaten eintragen	MS_TENANT_ID/MS_CLIENT_ID/MS_CLIENT_SECRET bzw. GOOGLE_CLIENT_ID/GOOGLE_CLIENT_SECRET in .env hinterlegen
3. Domain-Allowlist setzen	SSO_ALLOWED_DOMAINS mit den zugelassenen E-Mail-Domains füllen, z. B. SSO_ALLOWED_DOMAINS=example.de,example.com
4. Verfahren einschalten	MS_AUTH=1 und/oder GOOGLE_AUTH=1 setzen
5. Neu starten	sudo systemctl restart apache2

11.4 Sicherheitsverhalten

- **Domain-Allowlist**: Anmeldungen werden nur akzeptiert, wenn die E-Mail-Domain in SSO_ALLOWED_DOMAINS steht. Eine leere Allowlist lässt niemanden zu – das ist der sichere Default und keine Fehlkonfiguration
- **E-Mail-Verifizierung**: bei Google muss die E-Mail-Adresse laut Anbieter verifiziert sein (email_verified). Microsoft ist bereits über die Tenant-ID eingeschränkt

© 2012–2026 WolfInfoSec GmbH

- **Automatisches Anlegen**: ohne SSO_AUTO_CREATE=1 können sich nur bereits angelegte Benutzer über SSO anmelden. Mit aktivem Auto-Create werden unbekannte Benutzer angelegt – dann sollte die Allowlist besonders eng gefasst sein
- **Deaktivierte Konten**: Benutzer mit aktiv=0 können sich auch über SSO nicht anmelden
- **OIDC-Nonce**: für jeden Anmeldevorgang wird ein Nonce erzeugt und gegen das ID-Token geprüft

11.5 Automatische Weiterleitung zu Microsoft

Ist MS_AUTH=1 gesetzt, leitet /login von sich aus zu Entra weiter, statt zuerst das Passwortformular anzuzeigen. Der Sprung ist bewusst zurückhaltend gebaut, damit die Anmeldeseite weder in einer Schleife pendelt noch Meldungen verschluckt.

- **Einmal je Sitzung**: das Session-Merkmal sso_auto_versucht verhindert, dass ein fehlgeschlagener SSO-Versuch sofort erneut ausgelöst wird. Nach einer erfolgreichen Anmeldung wird es zurückgesetzt
- **Nach dem Abmelden kein Sprung**: der Logout setzt sso_kein_auto. Ohne diese Sperre würde die noch offene Microsoft-Sitzung den Benutzer kommentarlos wieder anmelden
- **Anstehende Meldungen haben Vorrang**: liegen Flash-Meldungen an (SSO-Fehler, Rate-Limit, falsches Passwort), bleibt die Seite stehen, damit sie gelesen werden können
- **Notfallzugang (Break-Glass)**: der Link „Anmeldung mit E-Mail und Passwort (Notfallzugang)“ ruft /login?breakglass=1 auf und zeigt das Passwortformular – etwa wenn Entra nicht erreichbar ist oder ein Konto noch kein SSO nutzt. Ein Link führt von dort zurück zur Microsoft-Anmeldung
- **Google in dieser Ansicht ausgeblendet**: solange automatisch zu Microsoft gesprungen wird, erscheint der Google-Button erst auf der Break-Glass-Seite

Vor dem Einschalten prüfen:
Mindestens ein Administrator muss sich über den Notfallzugang anmelden können. Ist MS_AUTH=1 gesetzt und Entra nicht erreichbar, führt der Weg ausschließlich über /login?breakglass=1.

© 2012–2026 WolfInfoSec GmbH

12. KI-Einstellungen

WolfISMS ermöglicht eine optionale KI-Unterstützung an mehreren Stellen: die KI-Assistenten der Fachmodule (seit v4.00 je eine eigene Seite: Risiken, Assets, Lieferanten, Dokumente, Maßnahmen, Schwachstellen, Changes, Incidents, Projekte, Controls, Rechtsregister, Datenschutz, Audits und Playbooks), die KI-Übersetzung von Dokumenten und die Aufbereitung eines Word-Imports (Übersicht im Abschnitt KI-Assistenten im Überblick, die Bedienung steht in der Benutzerdokumentation). Alle nutzen dieselbe Anbieter-Konfiguration. Sie wird ausschließlich über /admin/ki_einstellungen gepflegt, ist nur für Administratoren zugänglich und setzt eine bezahlte Lizenz voraus (ab Essentials).

12.1 Lizenzvoraussetzung: jede bezahlte Edition

Die KI-Funktionen stehen in jeder bezahlten Edition bereit – Essentials, Advanced und Enterprise, mit und ohne „+“; nur die freie Version (DEMO) sieht einen Hinweis. admin_ki_modul.ki_lizenz_ok() prüft das über isms_helpers.tier_ok() gegen die Rangstufe von Essentials – analog zum Framework-Import. Ohne gültige customer.lic gilt die Installation als DEMO und damit als gesperrt. Das Ergebnis wird je Request in flask.g gehalten, weil jede Prüfung das externe license_check-Binary startet und eine Seite mehrfach nachfragt.

Die Sperre greift an vier Stellen, nicht nur in der Oberfläche:

- **get_ki_config()** meldet licensed=False und erzwingt enabled=False
- **/admin/ki_einstellungen** zeigt eine Hinweisseite statt des Formulars und speichert nicht
- **ki_bearbeiten und ki_speichern** weisen Anfragen ab; ebenso die Routen aller KI-Assistenten
- **_call_ki_api()** bricht als letzte Absicherung ab, falls doch ein Aufruf durchkommt

Konfiguration bleibt erhalten:
Die in isms_settings hinterlegten Werte werden von der Sperre nicht angetastet und greifen wieder, sobald eine ausreichende Lizenz eingespielt ist. Die Admin-Kachel kennzeichnet die Funktion als gesperrt, statt sie zu verbergen.

12.2 Benutzerrecht „KI-Nutzung“

Seit v4.00 braucht jeder Benutzer neben Lizenz und Einrichtung das Recht „KI-Nutzung“ (Spalte users.ki, 0/1), um die KI-Assistenten zu verwenden. Vergeben wird es in der Benutzerverwaltung unter System-Berechtigungen (/admin/create_user, /admin/change_user); die Rechte-Matrix und der Excel-Export führen es als „KI“. Administratoren dürfen die KI immer – wie bei den Modulrechten. Die Vorgabe ist aus, auch für Benutzer, die es vor dem Update gab: isms_helpers.kartenrechte_sicherstellen() legt die Spalte mit 0 an, das Recht vergibt der Administrator ausdrücklich.

- **Knöpfe:** admin_ki_modul.ki_zugang_vorhanden() liefert ohne das Recht False – damit fehlen alle KI-Knöpfe und KI-Dialoge, auch der Knopf „KI-Assistent“ der Fachmodule.
- **Routen:** ein before_request-Hook in isms.py sperrt jede Adresse mit einem Pfadsegment ki_… (etwa /assets/ki_assistent, /dokumente/ki_bearbeiten). Seiten leiten mit Meldung auf die Startseite, Abfragen der KI-Dialoge erhalten JSON mit Fehlermeldung. /admin/… bleibt außen vor – KI-Einstellungen und Verbrauchsauswertung sind Verwaltung.
- **Aufruf:** _call_ki_api() lehnt als letzte Absicherung ab; ein Hintergrundauftrag (ki_job_starten) wird beim Start geprüft und endet ohne Recht sofort mit der Meldung.

© 2012–2026 WolfInfoSec GmbH

12.3 Datenbank-Voraussetzung (isms_settings)

Die KI-Konfiguration und das totp_required-Flag werden in der Tabelle isms_settings gespeichert (Key-Value-Store). Falls die Tabelle noch nicht existiert:

```
CREATE TABLE IF NOT EXISTS isms_settings (
  `key`   VARCHAR(64) PRIMARY KEY,
  `value` VARCHAR(255) NOT NULL DEFAULT ''
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;
```

12.4 Konfigurationsparameter

isms_settings-Schlüssel	Bedeutung / Werte
ki_enabled	„1“ = KI aktiv, „0“ = deaktiviert (Default)
ki_provider	KI-Anbieter: anthropic | openai | mistral | google | nele | azure_openai | copilot | custom (Default: anthropic). azure_openai und copilot (Azure AI Foundry) werden über denselben Aufrufpfad bedient. nele (seit v3.93, nele.ai der GAL Digital GmbH, Hosting in der EU) ruft POST https://api.aieva.io/api:v1/chat-completion-sync mit Bearer-Token auf; der Key kommt aus chat.nele.ai. Modell-IDs vergibt nele.ai je Konto, sie werden von Hand eingetragen. Die eingebaute Websuche wird als Internetrecherche angeboten. nele.ai rechnet in Credits ab und meldet keine Token – die Verbrauchsauswertung zählt nur Anfragen, ohne Kosten –, und eine am Längenlimit abgeschnittene Antwort wird nicht automatisch fortgesetzt
ki_model	Modell-ID des jeweiligen Anbieters
ki_api_key	API-Schlüssel des KI-Anbieters – seit v3.99 AES-256-GCM-verschlüsselt in isms_settings (Präfix enc1:; der Schlüssel dazu wird per HKDF aus dem Secret-Key der Anwendung abgeleitet). Kein Hash, da der Klartext für die API-Aufrufe benötigt wird. Ein noch Base64-kodierter Altbestand wird beim ersten Lesen umgestellt. Im Admin-UI nur maskiert dargestellt
ki_api_base_url	Nur für Azure OpenAI, Azure AI Foundry (Copilot) und Custom: Basis-URL des Endpunkts. Endet sie auf /openai/v1, wird die neue, OpenAI-kompatible Azure-Oberfläche verwendet: das Deployment steht dann als Modellname im Body und es wird keine API-Version gesendet
ki_api_version	Nur für Azure OpenAI / AI Foundry mit klassischer Endpunkt-URL, z. B. 2024-10-21. Azure führt laufend neue Versionen ein und stellt alte ab – maßgeblich ist die Ziel-URI des Deployments
ki_azure_auth	„key“ (Vorgabe) = Anmeldung mit dem api-key-Header; „entra“ = Token von Microsoft Entra ID statt API-Key. Nur für azure_openai und copilot, seit v4.00
ki_entra_tenant	Verzeichnis-ID (Tenant ID) des Entra-Mandanten
ki_entra_client_id	Anwendungs-ID (Client ID) der App-Registrierung
ki_entra_secret	Geheimer Clientschlüssel der App-Registrierung – AES-256-GCM-verschlüsselt wie ki_api_key. Nach dem Speichern liest die Maske den Wert zur Probe zurück und meldet „gespeichert, aber nicht lesbar“ getrennt von „leer“

© 2012–2026 WolfInfoSec GmbH

isms_settings-Schlüssel	Bedeutung / Werte
ki_prompt_addendum	Optionaler zusätzlicher Prompt-Text, der an alle KI-Anfragen angehängt wird (z. B. unternehmensspezifische Vorgaben)

12.5 Anmeldung per Entra ID (Azure)

Seit v4.00 melden sich Azure OpenAI und Microsoft Copilot (Azure AI Foundry) auf Wunsch über Microsoft Entra ID statt mit einem API-Key an. Nötig ist das bei Ressourcen, an denen die schlüsselbasierte Anmeldung abgeschaltet ist – oft per Azure Policy (disableLocalAuth). Der Verbindungstest meldet dann „Key based authentication is disabled for this resource“ mit HTTP 403.

- **Azure:** im Entra Admin Center eine App-Registrierung mit geheimem Clientschlüssel anlegen und ihr an der KI-Ressource unter „Zugriffssteuerung (IAM)“ die Rolle „Cognitive Services OpenAI User“ zuweisen
- **WolfISMS:** unter /admin/ki_einstellungen beim Anbieter Azure OpenAI oder Copilot „Authentifizierung: Entra ID“ wählen und Tenant-ID, Client-ID und Client-Secret eintragen; das Feld API-Key entfällt dann. Basis-URL, Modell bzw. Deployment und API-Version bleiben wie gehabt
- **Ablauf:** admin_ki_modul._entra_token() holt per Client-Credentials ein Token für https://cognitiveservices.azure.com/.default und hält es im Prozess bis kurz vor Ablauf vor. Gesendet wird nur Authorization: Bearer, kein api-key-Header
- **KI eingerichtet?:** die Module prüfen seit v4.00 ki_zugang_vorhanden(cfg) statt eines gesetzten API-Keys – sonst fehlten die KI-Knöpfe im Entra-Modus. Neue KI-Funktionen müssen das ebenso tun
- **Secret nicht lesbar:** zeigt die Maske „gespeichert, aber nicht lesbar“, konnte der Prozess den Wert nicht entschlüsseln – meist, weil ISMS_SECRET_KEY aus mehreren Quellen kommt (envvars des Apache, /etc/wolfisms/secret.env, .flask_secret_key). Seit v4.00 hält jeder Prozess den Secret-Key ab dem Start fest; das Secret danach einmal neu eintragen. Dasselbe gilt für die Client-Secrets des Single Sign-On

12.6 KI-Assistenten im Überblick

Alle Assistenten laufen über dieselbe Anbindung (_call_ki_api) und dasselbe Lizenz-Gate. Sie sind nur sichtbar, wenn die KI eingeschaltet ist, ein Zugang hinterlegt ist (API-Key oder Entra ID) und der Benutzer die Hürde der Ansicht nimmt, die der Assistent ersetzt; nicht im Nur-Lese-Modus.

Assistent	Ort	Recht	Schreibt
KI-Assistent Risiken	eigene Seite /risiko/ki_assistent, Knopf in Liste und Detailansicht	risk ≥ 2, kein Nur-Lese-Zugang	Risiken samt Bewertung und Verknüpfungen; fehlende Maßnahmen und Lieferanten
KI-Assistent Assets	eigene Seite /assets/ki_assistent, Knopf in Liste und Detailansicht	assets ≥ 2, kein Nur-Lese-Zugang	Assets samt Schutzbedarf (Begründung in bem_*) und Verknüpfungen; fehlende Lieferanten, Tochterassets und Risiken
KI-Assistent Lieferanten	eigene Seite /vendor/ki_assistent, Knopf in Liste und Detailansicht	vendor ≥ 2, kein Nur-Lese-Zugang	Lieferanten, Bewertung und Verknüpfungen; verschickt und wertet den Lieferantenfragebogen aus

© 2012–2026 WolfInfoSec GmbH

Assistent	Ort	Recht	Schreibt
KI-Assistent Dokumente	eigene Seite /dokumente/ki_assistent, Knopf in Liste und Detailansicht	dok ≥ 2, kein Nur-Lese-Zugang	neue Dokumente und Fassungen, Verknüpfungen, englische Fassung, Übergabe in die Prüfung; freigeben nie
KI-Assistent Maßnahmen	eigene Seite /massnahmen/ki_assistent, Knopf in Liste und Detailansicht	mass ≥ 2, kein Nur-Lese-Zugang	Maßnahmen, Stand und Nachweis, Verknüpfungen zu allen Bereichen
KI-Assistent Schwachstellen	eigene Seite /schwachstellen/ki_assistent, Knopf in Liste und Detailansicht	schwachstellen ≥ 2, kein Nur-Lese-Zugang	Schwachstellen (frei, CVE, Advisory, Scan- oder Pentestbericht), Bewertung, Behebungsnachweis, Verknüpfungen; ersetzt den Schwachstellen-Abgleich
KI-Assistent Changes	eigene Seite /change/ki_assistent, Knopf in Liste und Detailansicht	assets ≥ 2, kein Nur-Lese-Zugang	Changes und Statusschritte nach denselben Regeln wie /change/status; stimmt nie ab
KI-Assistent Incidents	eigene Seite /incidents/ki_assistent, Knopf in Liste und Detailansicht	incidents ≥ 2, kein Nur-Lese-Zugang	Vorfälle, Klassifizierung, Bewertung der Meldepflicht, erfasste Meldungen, Ursache und Lehren; meldet nie selbst; ersetzt die KI-Analyse
KI-Assistent Projekte	eigene Seite /projekte/ki_assistent, Knopf in Liste und Detailansicht	mass ≥ 2, kein Nur-Lese-Zugang	Projekte samt Steckbrief, Abschnitten, Rollen, Gates und Verknüpfungen
KI-Assistent Controls	eigene Seite /normen/ki_assistent, Knopf in Liste und Detailansicht	norm ≥ 2, kein Nur-Lese-Zugang	Status, Umsetzung, Nachweise und verwandte Controls – nur Bestätigtes; schlägt nie einen Ausschluss vor
KI-Assistent Rechtsregister	eigene Seite /rechtsregister/ki_assistent, Knopf in Liste und Detailansicht	norm ≥ 2 oder Admin, kein Nur-Lese-Zugang	Einträge (immer „Zu prüfen“), Umsetzung, Nachweise, Prüfdaten; nie „Nicht geltend“ oder „Konform“
KI-Assistent Audits	eigene Seite /audit/ki_assistent, Knopf in Liste und Detailansicht	audit ≥ 2, kein Nur-Lese-Zugang	Audits mit Prüfkatalog und Prüffragen, Bewertungsvorschläge, Abweichungen, Auditbericht; Prüffragen in aud_pruefpunkte.beschreibung
KI-Assistent Playbooks	eigene Seite /playbooks/ki_assistent, Knopf in Liste und Detailansicht	play ≥ 2 oder Admin, kein Nur-Lese-Zugang	Playbooks samt Abschnitten, Ablaufschritten, Rollen, Tests und Verknüpfungen
KI-Assistent Datenschutz	eigene Seite /datenschutz/ki_assistent, Knopf in Liste und Detailansicht	datenschutz ≥ 2 oder Admin, kein Nur-Lese-Zugang	Verarbeitungen als Entwurf, Vorschlag zur Schwellwertanalyse, DSFA-Entwurf, TOM-Anlage als Word-Datei; nie „Freigegeben“, keine Restwerte und keine Stellungnahme des DSB
KI-	Bearbeitungsformular	Bearbeitungsrec	nichts – das Ergebnis steht im

Assistent	Ort	Recht	Schreibt
Aufbereitung beim Word-Import	eines Dokuments, Haken im Kasten „Aus Word übernehmen“	ht	Editor, gespeichert wird mit „Speichern“
KI-Übersetzung	Dokument, Detailansicht, nur englische Oberfläche	dok ≥ 1 und Besitzer; nicht freigegeben, in Prüfung oder abgelöst	Dokumente.text_en, auf Wunsch Dokumentenname_en – als neue Version
GAP-Assistent	Audit der Art gap: Prüfliste und Detailansicht	audit ≥ 2	führt durch die Prüfliste; Bewertung, Feststellung und Nachweis je Punkt

- **Haftungshinweis**: Jeder KI-Dialog zeigt unter dem Titel den Kasten aus admin_ki_modul.ki_gewaehr_html() – KI-generierte Angaben sind vor der Verwendung zu prüfen, für Richtigkeit und Vollständigkeit wird keine Gewähr übernommen. Ein neuer Assistent muss ihn mitbringen
- **Antwort gegen den Bestand prüfen**: Vorgeschlagene IDs werden gegen die echten Kataloge abgeglichen, erfundene und bereits verknüpfte fallen heraus. Die KI-Risikoanalyse übernimmt Kategorien, Bedrohungen und Schwachstellen nur aus den Katalogen, höchstens drei je Risiko
- **Frühere Dialoge:** Der Risiko-Assistent im Dialogfenster sowie „KI-Risikoanalyse“ und „KI-Schutzbedarf“ am Asset sind seit v4.00 in den beiden KI-Assistenten aufgegangen; ihre Knöpfe sind entfallen. Die Routen in risiko_ki_modul.py und asset_ki_modul.py bleiben bestehen, die Assistenten nutzen deren Kern – die folgenden Regeln gelten dort weiter
- **Neue Risiken aus der KI-Risikoanalyse**: entstehen wie im Einrichtungsassistenten mit Status „offen“, Behandlung „nicht bewertet“, der Abteilung des Assets und der Ersteinschätzung als wahrvm/auswvm (Werte nach Maßnahmen 0), dazu eine Zeile in Risiko_historie. Die Mengengrenze der Edition gilt, und vorhandene Risiken, die der Benutzer nach der Besitzerregel nicht ändern darf, werden übersprungen und gemeldet
- **KI-Schutzbedarf**: bietet die Stufen der Tabelle kritikalität ohne „kritisch“ an – dieselben wie die Bearbeitungsmaske. Geschrieben werden nur die bestätigten Schutzziele, die Begründung seit v3.95 in die Bemerkungsspalte bem_<schutzziel> (auch wenn die Stufe gleich bleibt); Wert und Vererbung fasst er nicht an, die Kritikalität rechnet kritik_nachziehen() nach. Ist das Asset von der Vererbung ausgenommen, geht die Begründung mit an das Modell
- **Protokoll**: Verknüpfen, Anlegen und Ändern stehen im Aktivitätenprotokoll mit dem Vermerk „per KI“ bzw. „Schutzbedarf per KI“
- **KI-Verknüpfungen (dokument_ki_modul.py):** seit v3.94, seit v4.00 ohne eigenen Knopf – die Aufgabe Verknüpfungen des Dokumenten-Assistenten (dokument_ki_verknuepfung.py) tritt an ihre Stelle; Routen POST /dokumente/ki_verknuepfen, …_stand und …_uebernehmen. Der Aufruf läuft als Hintergrundauftrag (ki_jobs, Bereich dokumente_link), weil Dokumententext und vier Kataloge die 60 Sekunden von Apache überschreiten können. Je Katalog gehen höchstens 150 Einträge bzw. 22.000 Zeichen mit, nach Stichwortnähe vorsortiert; höchstens zehn Vorschläge je Bereich. Geschrieben wird an dieselben Stellen wie die Verknüpfungsansichten /dokumente?link=…; Dokumente, die von der Gegenseite schon verweisen, gelten als verknüpft. Der Blueprint hängt am Registry-Eintrag dokumente
- **Risiko-Texte:** seit v3.95 hat der Risiko-Assistent den Bereich texte (erster in BEREICHE, nur mit risk ≥ 2): Vorschläge für Beschreibung, Business Impact und Chancen (TEXTFELDER). Die Übernahme escaped die Texte, wandelt sie in HTML, prüft sie mit Pruefung und schreibt in die Spalten der Anzeigesprache (bei EN die _en-Spalten) – Besitzerregel wie beim Bearbeiten
- **KI-Übersetzung (dokument_uebersetzung_modul.py):** seit v3.95; Routen POST /dokumente/ki_uebersetzen, …_stand und …_uebernehmen, Blueprint dokument_uebersetzung

© 2012–2026 WolfInfoSec GmbH

am Registry-Eintrag dokumente. Das Modell bekommt nie HTML: zerlegen() teilt den deutschen Text an jedem Block-Tag (Überschrift, Absatz, Listenpunkt, Zelle, br, img) in Abschnitte, Inline-Tags (strong, em, a, span …) werden zu Platzhaltern <x1>…</x1>, Inhalt von pre bleibt unübersetzt, gleiche Abschnitte gehen nur einmal mit. zusammensetzen() setzt die Übersetzung in das unveränderte Original-HTML zurück – Tags, Attribute und eingebettete Bilder stammen nie aus der Antwort. Stimmen die Platzhalter eines Abschnitts nicht, entfällt dort nur die Inline-Formatierung; fehlt ein Abschnitt, bleibt er deutsch; beides meldet der Dialog. Blöcke zu 6.000 Zeichen bzw. 120 Abschnitten laufen als Hintergrundauftrag mit Fortschritt (Bereich dokumente_uebersetzung); eine unbrauchbare Antwort wird halbiert und erneut angefragt. Gesperrt wie die KI-Bearbeitung, gespeichert wie ki_speichern: sanitize_tinymce_html, Längenprüfung gegen text_en, neue Version über _version_anlegen()

12.7 Beratergespräche: KI-Assistenten für Risiken und Assets

Seit v4.00 haben Risiken und Assets je einen KI-Assistenten auf einer eigenen Seite: ein Gespräch, in dem die KI Einträge gemeinsam mit dem Anwender anlegt, prüft und verknüpft. Der Knopf „KI-Assistent“ steht im Kopf der Liste und in der Detailansicht; aus der Detailansicht beginnt das Gespräch mit diesem Eintrag.

Eigenschaft	Risiken	Assets
Modul, Route	risiko_ki_assistent.py, GET/POST /risiko/ki_assistent[?risiko=<id>]; Aufgaben in risiko_ki_neu.py, risiko_ki_verknuepfung.py, risiko_ki_stapel.py	asset_ki_assistent.py, GET/POST /assets/ki_assistent[?asset=<id>]; Aufgaben in asset_ki_neu.py, asset_ki_verknuepfung.py, asset_ki_stapel.py
Aufgaben	Neues Risiko anlegen (frei oder ausgehend von Asset, Bedrohung, Schwachstelle oder Incident), mehrere Risiken aus Liste oder Upload, Lücken finden, ein Risiko prüfen, eine Kategorie prüfen, Gesamtbestand prüfen	Neues Asset anlegen, mehrere Assets aus Liste oder Upload, ein Asset prüfen, eine OBASHI-Ebene prüfen, Gesamtbestand prüfen
Recht	risk ≥ 2	assets ≥ 2
Dialogstand	Tabelle ki_dialog, Bereich risiko	Tabelle ki_dialog, Bereich asset

- **Aufbau:** Chat links mit einer Antwortkarte darunter, rechts der bearbeitbare Entwurf bzw. eine Vorschlagstabelle; der gemeinsame Unterbau liegt in ki_dialog_bausteine.py. GET zeichnet die Seite, POST (fetch aus dem Browser) führt eine Aktion aus und liefert die neuen Seitenteile als JSON. Längere Aufrufe laufen als Auftrag im Hintergrund (ki_jobs)
- **Schreiben:** nichts ohne Bestätigung. Entwurf und Verknüpfungen werden je Bereich vorgeschlagen und begründet; erst „Übernehmen“ schreibt. Was in einem anderen Modul angelegt wird – Maßnahmen, Lieferanten, Risiken, Tochterassets –, prüft dessen Recht
- **Listen:** hochgeladen werden Excel, CSV, PDF, Word oder ein Bildschirmfoto; die KI liest die Einträge aus, der Assistent gleicht sie mit dem Bestand ab und geht sie nacheinander durch
- **Prüfen im Stapel:** Kategorie, Ebene und Gesamtbestand beginnen mit den Einträgen mit dem größten Handlungsbedarf – bei Risiken etwa über dem Risikoappetit ohne Behandlung oder ohne Begründung akzeptiert, bei Assets ohne Verknüpfungen, Schutzbedarf, Besitzer oder Zeitziele
- **Gesprächsstand:** eine Zeile je Benutzer und Assistent; wer nach einer Pause zurückkommt, kann dort weitermachen

© 2012–2026 WolfInfoSec GmbH

12.8 KI-Assistenten der übrigen Fachmodule

Seit v4.00 hat jedes Fachmodul einen KI-Assistenten nach dem Muster von Risiken und Assets: dieselbe Seite (Chat, Antwortkarte, Entwurf rechts), derselbe Unterbau (ki_dialog_bausteine.py), dieselbe Regel „nichts ohne Bestätigung“. Route ist jeweils GET/POST <modul>/ki_assistent, aus der Detailansicht mit der ID des Eintrags; das Recht ist das Schreibrecht des Moduls (siehe Übersicht der Assistenten). Der Gesprächsstand steht in ki_dialog, eine Zeile je Benutzer und Bereich.

Modul	Datei	Aufgaben	Bereich ki_dialog
Lieferanten	lieferant_ki_assistent.py (Aufgaben in lieferant_ki_neu, _verknuepfung, _fragebogen, _stapel)	anlegen (frei, aus Asset oder Unterlage), mehrere aus Liste/Upload, Lückenanalyse, prüfen (Revision), Typ/Tier, Gesamtbestand, Nachweise auswerten, Fragebogen verschicken und auswerten (auch CAIQ, VDA ISA/TISAX)	vendor
Dokumente	dokument_ki_assistent.py (dokument_ki_neu, _verknuepfung, _stapel)	erstellen, aus Unterlagen übernehmen, überarbeiten, prüfen, zusammenführen, übersetzen, in die Prüfung geben, verknüpfen, Lückenanalyse (Pflichtdokumente, Controls ohne Dokument), Typ, Gesamtbestand	dokument
Maßnahmen	massnahme_ki_assistent.py (massnahme_ki_neu, _verknuepfung, _stapel)	anlegen (frei oder aus Risiko, Control, Abweichung, Incident, Schwachstelle, Lieferant, Rechtsregister), mehrere aus Bericht oder Maßnahmenplan, Lückenanalyse, prüfen, Statusrunde, Nachweis, Gesamtbestand, Umsetzungsreihenfolge	massnahme
Schwachstellen	schwachstelle_ki_assistent.py (schwachstelle_ki_neu, _verknuepfung, _stapel)	erfassen (frei, CVE, Advisory), Scan-/Pentestbericht übernehmen, Asset-Versionen gegen bekannte CVEs prüfen, prüfen, Triage, Statusrunde, Behebung belegen, Gesamtbestand, Lagebericht	schwachstelle
Changes	change_ki_assistent.py (change_ki_neu, _verknuepfung, _stapel)	entwerfen, prüfen, Statusschritt begleiten, Gremium vorschlagen, Nachbetrachtung, Statusrunde, Kollisionsprüfung im Änderungskalender, Lagebericht	change
Incidents	incident_ki_assistent.py (incident_ki_neu, _melden, _verknuepfung, _stapel)	erfassen, prüfen, zusammenführen, Meldepflicht nach NIS-2, DSGVO und CRA bewerten, Meldung vorbereiten und erfassen, Ursache und Lehren, Statusrunde, Lagebericht	incident
Projekte	projekt_ki_assistent.py (projekt_ki_neu, _verknuepfung, _stapel)	anlegen (Idee oder Projektauftrag), ISMS-Check, Sicherheitsanforderungen ableiten, Risikobewertung, Gate-Review, Abschluss, Statusrunde, Lagebericht	projekt
Controls	normen_ki_assistent.p	Fragen beantworten, Control prüfen,	norm

© 2012–2026 WolfInfoSec GmbH

Modul	Datei	Aufgaben	Bereich ki_dialog
	y (normen_ki_neu, _stapel; Mapping normen_mapping.py)	Umsetzung beschreiben, Nachweis prüfen, Ausschlüsse auf Anhaltspunkte prüfen, Lückenanalyse, Mapping zwischen Frameworks, Stand verwandter Controls übertragen, Statusrunde, Lagebericht	
Rechtsregister	rechtsregister_ki_assistent.py (rechtsregister_ki_neu, _stapel)	Fragen, Einträge anlegen, fehlende Rechtsgrundlagen, prüfen, Umsetzung, Nachweis, Prüfrunde, Rechtsänderungen, Lagebericht	recht
Audits	audit_ki_assistent.py (audit_ki_neu, _pruefung, _stapel)	Audit planen samt Prüfkatalog und Prüffragen, Prüfpunkte bewerten, Abweichungen aufbereiten, Auditbericht, Auditprogramm, Nachverfolgung, Audit-Bereitschaft, Lagebericht	audit
Playbooks	playbook_ki_assistent.py (playbook_ki_neu, _verknuepfung, _stapel)	anlegen (Szenario schildern oder Dokument hochladen), prüfen, Übung (Tabletop) vorbereiten, Testergebnis erfassen, Nachbereitung nach einem Incident, Lückenanalyse, Statusrunde, Lagebericht	playbook
Datenschutz	datenschutz_ki_assistent.py (datenschutz_ki_neu)	Verarbeitung anlegen, als Auftragsverarbeiter erfassen, Lücken aus den Assets (nur auf Knopfdruck), Eintrag prüfen, Schwellwertanalyse, DSFA entwerfen, TOM-Anlage nach Art. 32	datenschutz

- **Grenzen:** Bewertungen, Status und Anwendbarkeit sind immer Vorschläge. Der Change-Assistent stimmt nie im Gremium ab, der Incident-Assistent meldet nie an eine Behörde, der Controls-Assistent schlägt nie einen Ausschluss vor (er meldet nur Anhaltspunkte, dass ein ausgeschlossenes Control doch anwendbar sein dürfte), der Audit-Assistent entscheidet keine Bewertung – „umgesetzt“ nur mit Text zur Wirksamkeitsprüfung
- **Keine Rechtsberatung:** der Rechtsregister-Assistent legt Einträge immer mit „Zu prüfen“ an und schlägt weder „Nicht geltend“ noch „Konform“ vor
- **Datenschutz:** auch der Datenschutz-Assistent berät nicht rechtlich – Rechtsgrundlage, Löschfrist und Ergebnis der Schwellwertanalyse bleiben Vorschläge, freigeben und abschließen tut der Mensch
- **Websuche:** die Aufgabe Rechtsänderungen recherchiert im Internet nur, wenn Anbieter und Modell es können (admin_ki_modul.web_suche_moeglich) – sonst arbeitet sie ohne Websuche und sagt das. Treffer stehen mit Quelle da
- **Übergaben:** neue Maßnahmen entstehen im Maßnahmen-Assistenten – aus Abweichungen, aus dem Rechtsregister und aus den übrigen Modulen springt der Assistent dorthin
- **Lizenz:** jede bezahlte Edition; die Routen prüfen ki_zugang_vorhanden() und das Lizenz-Gate selbst. Die Module, an denen die Assistenten hängen, behalten ihre Edition

© 2012–2026 WolfInfoSec GmbH

12.9 Länge der Antworten

Bis v3.71 rief WolfISMS jedes Modell mit fest 4.096 max_tokens auf. Für eine überarbeitete Richtlinie reicht das nicht – die Antwort brach nach rund 7.500 Zeichen mitten im Satz ab. Seit v3.72 gilt eine Staffelung in drei Stufen:

- **Grenze je Modell**: Voreinstellung sind 16.384 Tokens. Modelle mit kleinerer Obergrenze stehen in einer Tabelle im Modul – claude-3-5 und claude-3-7 mit 8.192, claude-3, gpt-4, gpt-4-turbo und gpt-3.5 mit 4.096
- **Rückfall bei Ablehnung**: lehnt der Anbieter den Wert mit HTTP 400 ab, wird die genannte Obergrenze aus der Fehlermeldung gelesen und der Aufruf wiederholt. Verlangt er max_completion_tokens statt max_tokens, wird auch das nachgezogen
- **Fortsetzen**: endet die Antwort trotzdem am Limit, holt WolfISMS sie in bis zu drei weiteren Runden zu Ende; der bisherige Text geht dabei als Vorgabe an das Modell zurück

Ein Aufruf samt Fortsetzungen hat ein Zeitbudget von 600 Sekunden; eine weitere Runde beginnt nur, wenn davon noch mindestens 30 Sekunden übrig sind. Der Rahmen dafür ist der Ingress, der eine Antwort 300 Sekunden offen lässt (k8s/09-ingress.yaml) – lieber der bis dahin geschriebene Text als eine tote Verbindung. Die Werte sind Konstanten in admin_ki_modul.py (KI_TOKENS_STD, KI_FORTSETZUNGEN, KI_GESAMTZEIT, KI_MINDESTZEIT) und stehen bewusst nicht in isms_settings.

Auch die Datenbank muss die längere Antwort fassen.
Dokumente.Text war schon immer mediumtext, die englische Fassung text_en dagegen TEXT – 65.535 Bytes, die MySQL außerhalb des Strict-Modus still abschneidet. Das Dokumentenmodul hebt die Spalte deshalb einmalig auf mediumtext an. Fehlen dem Datenbankbenutzer die ALTER-Rechte, bleibt es bei der kleineren Grenze: ein zu langer Text wird dann gemeldet und nicht stillschweigend gekürzt.

12.10 Verbrauch und Kosten

Über den unter KI-Einstellungen hinterlegten Arbeits-Key gibt kein Anbieter einen Kontostand heraus. Anthropic und OpenAI verlangen für ihre Usage- und Cost-API einen eigenen Admin-Schlüssel, der auch Benutzer verwalten und Schlüssel abschalten darf; Azure rechnet gar nicht über den Key ab, sondern über die Kostenanalyse des Abonnements. Ein solcher Schlüssel gehört nicht in ein ISMS. WolfISMS schreibt den Verbrauch deshalb selbst mit: jede Antwort führt ihre eigene Nutzung im Feld usage mit, und die wird je Aufruf aufsummiert und in der Tabelle ki_verbrauch abgelegt. Die Tabelle legt das Modul beim ersten Aufruf selbst an.

Erfasst wird an einer einzigen Stelle – in _call_ki_api() in admin_ki_modul.py, über die alle KI-Funktionen laufen. Jede Zeile hält Zeitpunkt, Benutzer, Modul, Anbieter, Modell, die Tokenzahlen und die Zahl der HTTP-Anfragen fest:

- **Aufruf und Anfrage**: eine Zeile entspricht einem Aufruf, etwa einer KI-Bearbeitung eines Dokuments. Ein Aufruf kann mehrere HTTP-Anfragen auslösen – die Fortsetzungsrunden am Ausgabelimit aus dem vorigen Abschnitt und die Pausen einer Websuche. Deshalb steht die Zahl der Anfragen als eigener Wert daneben
- **Modul**: dokumente, risiken, assets, schutzbedarf, incidents, schwachstellen, audit oder test – der Verbindungstest der KI-Einstellungen zählt mit
- **Abgebrochene Aufrufe**: auch ein Aufruf, der mit einem Fehler endet, hat bereits Token verbraucht. Gespeichert wird deshalb in jedem Fall, nicht nur nach einer erfolgreichen Antwort

© 2012–2026 WolfInfoSec GmbH

- **Zählweise**: Anthropic weist die Eingabe ohne die aus dem Cache gelesenen Token aus, die Chat-Completions-Oberfläche von OpenAI und Azure zählt sie mit. WolfISMS rechnet auf die Anthropic-Lesart um – sonst wären dieselben Kosten je Anbieter unterschiedlich gerechnet

Die Auswertung steht unter /admin/ki_verbrauch, erreichbar über die Kachel KI-Verbrauch der Administration und über die KI-Einstellungen. Die Seite ist Administratoren vorbehalten und wie alle KI-Funktionen in jeder bezahlten Edition freigeschaltet. Sie zeigt für den gewählten Monat oder den gesamten Zeitraum die Aufrufe, die Anfragen, die Token und die geschätzten Kosten, aufgeschlüsselt nach Modul, Benutzer und Modell, dazu den Verlauf der letzten zwölf Monate.

Preisliste

Aus Token werden erst mit einer Preisliste Kosten. Weil die Anbieter ihre Preise ändern, steht sie nicht im Programm fest, sondern wird auf derselben Seite gepflegt – je Modell ein Eingabe- und ein Ausgabepreis pro 1 Mio. Token, dazu das Währungskürzel. Die Anbieter listen in US-Dollar; wer in Euro rechnen möchte, trägt Euro-Preise ein und stellt das Kürzel um. Eine Kursumrechnung nimmt WolfISMS nicht vor.

- **Zwischengespeicherte Eingabe**: wird aus dem Eingabepreis abgeleitet, statt zwei weitere Preise je Modell pflegen zu lassen. Das Schreiben in den Cache zählt 1,25-fach, das Lesen daraus 0,10-fach (KI_CACHE_FAKTOR_NEU, KI_CACHE_FAKTOR_LES)
- **Namen greifen als Präfix**: verglichen wird der Anfang des Modellnamens, längere Einträge zuerst. Ein Eintrag claude-haiku-4-5 bepreist damit auch die datierte Variante claude-haiku-4-5-20251001
- **Auslieferungszustand**: hinterlegt sind nur die Modelle, deren Listenpreis belegt ist – die Claude-Modelle nach dem Stand 06/2026. Für OpenAI, Azure und ältere Claude-Modelle ist bewusst nichts eingetragen, statt zu schätzen

Aufrufe mit einem Modell ohne hinterlegten Preis zählen bei den Token mit und bei den Kosten nicht; ihre Zahl weist jede Tabelle gesondert aus. So wird aus einer Lücke in der Preisliste keine zu niedrige Summe. Ein Modell wird entfernt, indem sein Name geleert und gespeichert wird; die Schaltfläche Auslieferungswerte stellt die mitgelieferte Liste wieder her.

Aufbewahrung und Grenzen

Einträge älter als 730 Tage werden beim Schreiben entfernt (KI_VERBRAUCH_AUFHEBEN in admin_ki_modul.py). Die Seite hat zusätzlich eine Schaltfläche, die alle erfassten Daten löscht – die Erfassung läuft danach weiter, die Auswertung beginnt wieder bei null.

Eine Näherung, keine Abrechnung
Serverseitige Werkzeuge wie die Internetrecherche werden vom Anbieter zusätzlich berechnet und stecken in der gemeldeten Nutzung nicht vollständig. Für die Frage, welches Modul die Kosten verursacht, sind die Zahlen aussagekräftiger als das Dashboard des Anbieters; für die Rechnungsprüfung bleibt dessen Abrechnung maßgeblich.

12.11 Aufträge im Hintergrund

Die Überarbeitung eines längeren Dokuments dauert beim Anbieter Minuten. So lange lässt sich eine Anfrage nicht offen halten: Apache beendet einen Request nach 60 Sekunden, mod_wsgi und gunicorn haben eigene Grenzen, und davor sitzt in der Regel noch ein Proxy mit den seinen. Die Verbindung brach ab, und das Ergebnis war verloren.

© 2012–2026 WolfInfoSec GmbH

Deshalb antwortet die Route nicht mit dem Ergebnis, sondern mit einer Auftragsnummer: Sie legt den Auftrag an, ein Thread arbeitet weiter, und der Browser fragt den Stand alle paar Sekunden ab. Jede einzelne Anfrage bleibt damit kurz.

Tabelle ki_jobs

Der Stand steht in der Datenbank und nicht im Prozessspeicher – bei mehreren Workern landete die Abfrage sonst beim falschen Prozess und fände den Auftrag nicht. Die Tabelle wird beim ersten Zugriff angelegt; Migrationsskripte gibt es wie im übrigen Projekt nicht.

Spalte	Bedeutung
id	Auftragsnummer, 32-stellige Hexzahl aus uuid4. Nur dieses Format wird angenommen
bereich	Woher der Auftrag kommt: dokumente, assets, schutzbedarf oder audit
nutzer	Benutzer-ID des Auftraggebers
status	laeuft, fertig oder fehler
ergebnis	Die Antwort der Arbeitsfunktion als JSON (MEDIUMTEXT)
angelegt / geaendert	Zeitpunkte für Aufräumen und Zeitüberschreitung

- **KI_JOB_MAXDAUER (1800 Sekunden)**: So lange darf ein Auftrag auf „laeuft“ stehen. Danach gilt er als abgebrochen – etwa, weil der Worker zwischendurch neu gestartet wurde. Der Anwender bekommt die Meldung, die Bearbeitung erneut zu starten.
- **KI_JOB_AUFHEBEN (86400 Sekunden)**: Beim Anlegen des nächsten Auftrags werden ältere Zeilen gelöscht. Ohne das wüchse die Tabelle unbegrenzt; ein eigener Cron-Lauf ist dafür nicht nötig.
- **Nur der eigene Auftrag**: Die Standsabfrage vergleicht die Benutzer-ID der Zeile mit dem angemeldeten Benutzer und gibt einen fremden Auftrag nicht heraus – die Auftragsnummer allein genügt nicht.

Der Thread lebt im Worker-Prozess.
Ein Neustart von Apache während eines Laufs verwirft ihn; der Auftrag bleibt auf „laeuft“ stehen, bis KI_JOB_MAXDAUER abgelaufen ist, und wird dann als unterbrochen gemeldet. Verbrauch und Kosten werden erst nach dem Aufruf verbucht – ein verworfener Lauf kann beim Anbieter dennoch angefallen sein.

Über diesen Weg laufen die KI-Bearbeitung eines Dokuments, die Aufbereitung eines Word-Imports, die Aufrufe aller KI-Assistenten (ki_dialog_bausteine.job_starten), die KI-Verknüpfungen und die KI-Übersetzung eines Dokuments und der Gap-Assessment-Assistent – dieser mit Fortschrittsmeldung (ki_job_starten(…, mit_fortschritt=True)), die ki_job_stand() als fortschritt an den Browser reicht. Seit v4.00 gibt es keinen KI-Aufruf mehr, der direkt an Apache vorbei auf die Antwort wartet.

12.12 Sicherheitshinweise

- Der API-Key liegt seit v3.99 AES-256-GCM-verschlüsselt in isms_settings. Der Schlüssel dazu wird aus dem Secret-Key der Anwendung abgeleitet (ISMS_SECRET_KEY bzw. .flask_secret_key im Session-Verzeichnis) und liegt damit außerhalb der Datenbank: ein Datenbankabzug oder ein Backup geben den Key nicht mehr preis. Bis v3.98 war er nur Base64-kodiert; solche Werte stellt

© 2012–2026 WolfInfoSec GmbH

die Anwendung beim ersten Lesen um. Der Datenbankbenutzer sollte trotzdem möglichst eingeschränkte Rechte haben

- Im Admin-UI wird der API-Key nur maskiert angezeigt (erste und letzte vier Zeichen); der vollständige Key lässt sich nicht mehr auslesen. Wird der Secret-Key der Anwendung ausgetauscht (anderes ISMS_SECRET_KEY, gelöschtes Session-Verzeichnis), ist der gespeicherte Key nicht mehr entschlüsselbar: die Maske zeigt dann keinen Key und das Log eine Warnung – den Key unter /admin/ki_einstellungen neu eintragen
- Dokumenteninhalte werden bei KI-Anfragen an den externen Anbieter übermittelt – Datenschutz- und Vertraulichkeitsanforderungen prüfen, bevor die Funktion aktiviert wird
- **Fehlermeldungen**: schlägt ein Aufruf fehl, wird die Meldung des Anbieters im Wortlaut angezeigt statt eines allgemeinen Fehlers. Ein leerer Modellname wird bereits vor dem Aufruf abgewiesen, weil die Anbieter ihn nur mit einem nichtssagenden HTTP 400 quittieren

© 2012–2026 WolfInfoSec GmbH

13. Design und Farbschema

Unter /admin/design lässt sich das Farbschema der Oberfläche einstellen (theme_modul.py). Der Menüpunkt ist Administratoren vorbehalten.

13.1 Funktionsweise

Die Oberfläche wird nicht über Stylesheets gestaltet: die Module bauen ihr HTML als f-String zusammen und tragen die Farben als Hex-Literale ein – rund 2.500 Stück in über 40 Dateien. Dieselben Literale stecken teilweise in Excel- und PDF-Exporten, wo sie nicht als CSS, sondern als Bibliotheksparameter gebraucht werden. Ein Suchen und Ersetzen quer durch die Module wäre daher weder vollständig noch gefahrlos.

Stattdessen rechnet ein after_request-Hook in isms.py jede ausgelieferte HTML-Antwort auf die eingestellte Palette um (theme_modul.response_einfaerben):

- **Umrechnung**: der Farbton wird auf den Grundton des Themes gezogen, die Sättigung gedämpft, die Helligkeit bleibt erhalten. Kontraste, Hell-Dunkel-Verhältnisse und damit die Lesbarkeit bleiben so, wie sie entworfen wurden – nur in einer Farbfamilie
- **Nur HTML**: der Hook fasst ausschließlich text/html an. Exporte und Downloads (xlsx, pdf, docx) bleiben unverändert
- **Vorlage angepasst**: in templates/template.html steht die Hover-Farbe der Links seit v3.51 als #0000ff statt des Schlüsselworts blue, damit auch sie von der Umrechnung erfasst wird
- **Feste Farben:** Seit v3.94 rechnet recolor_html() nichts um, was zwischen den Markern FARBE_FEST_AN/FARBE_FEST_AUS (HTML-Kommentare) bzw. im <style> zwischen FARBE_FEST_CSS_AN/FARBE_FEST_CSS_AUS (CSS-Kommentare /*farbe-fest*/ … /*/farbe-fest*/) steht. Ein HTML-Kommentar im Stylesheet wäre ungültiges CSS und risse die folgende Regel mit. Geschützt sind Risikomatrix samt Legende, die Kacheln der Risikostufen und die Score-Tachos auf der Startseite sowie in risiken.py und lieferanten.py – grün bis rot ist dort die Aussage. Neue Elemente mit fester Farbbedeutung gehören ebenso eingeschlossen

13.2 Verfügbare Farbschemata

Schlüssel	Bezeichnung	Charakter
petrol	Petrol / Blaugrün	Kühles Blaugrün, etwas freundlicher als reines Blau. Standard
klassisch	Klassisch (bunt)	Originalfarben von WolfISMS; die Umrechnung entfällt vollständig
graustufen	Schwarz / Grau / Weiß	Seit v4.00. Schlicht ohne Farbe: Sättigung 0, Signalfarben werden zu festen Helligkeitsstufen, kräftige Farben dunkeln Richtung Schwarz; Kopfleiste als Vorgabe #1c1c1e

Die früheren Schemata Blau/Grau, Graphit, Marine und Schiefer sowie die eigene Feineinstellung mit Reglern für Grundton und Sättigung sind entfallen; die Auswahl zeigt zu jedem Schema Streifen mit den umgerechneten Signalfarben. Die eigene Hintergrundfarbe des Hauptmenüs gibt es seit v4.00 nicht mehr – die Seitenleiste folgt dem Schema von selbst.

13.3 Hell und Dunkel, Kopfleiste und Infokacheln (isms_settings)

Die Einstellungen liegen als Schlüssel-Wert-Paare in derselben Tabelle isms_settings wie die KI-Konfiguration. Fehlt die Tabelle, wird sie beim ersten Aufruf automatisch angelegt.

© 2012–2026 WolfInfoSec GmbH

isms_settings-Schlüssel	Werte	Bedeutung
theme_name	petrol | klassisch | graustufen	Aktives Schema; Vorgabe petrol
theme_dunkel	1 | 0	Dunkelmodus als Vorgabe für alle, die nicht selbst gewählt haben
modus_u<id>	dunkel | hell	Eigene Wahl eines Benutzers über den Knopf neben der Sprachflagge (POST /design/modus); ohne Anmeldung gilt die Sitzung
theme_kopf_bg	#rrggbb oder leer	Hintergrund der Kopfleiste; leer = Vorgabe des Schemas (#0c519b, bei Schwarz / Grau / Weiß #1c1c1e). Die Farbe wird nicht umgerechnet, auf hellem Grund wird die Schrift dunkel
theme_infokacheln	1 | 0	Statistik- und Kategoriekacheln über den Modullisten zeigen; Vorgabe 0 = ausgeblendet
theme_menu_bg	–	Früher die eigene Menüfarbe; seit v4.00 ohne Wirkung, Speichern leert den Wert

- **Dunkelmodus:** rechnet jede Farbe nach ihrer Rolle um – Text, Rand oder Fläche, erkannt an der CSS-Eigenschaft vor der Farbe –, damit Menü und dunkle Knöpfe dunkel bleiben. Bereiche zwischen den FARBE_FEST-Markern bleiben auch dort unverändert. Neue Farben deshalb als Hex mit dem Eigenschaftsnamen davor schreiben, sonst gelten sie als Fläche
- **Infokacheln:** ausgeblendet wird beim Ausliefern per CSS über die Klassen in theme_modul.INFOKACHEL_KLASSEN (18 Module). Dashboard, Administration, Historie und PDF-Berichte bleiben unberührt, ebenso die Filterkacheln der Listen
- **Zurücksetzen:** stellt Petrol ohne Dunkelvorgabe, die Kopfleiste des Schemas und ausgeblendete Infokacheln wieder her; die Hell/Dunkel-Wahl der Benutzer bleibt. Im Nur-Lese-Modus wird nichts gespeichert. Bei mehreren Serverprozessen greift eine Änderung nach spätestens 20 Sekunden überall

13.4 Tooltips

Seit v3.97 tragen Formularfelder und Tabellenspalten aller Module ein ⓘ-Symbol mit einem kurzen Erklärtext. Er erscheint, wenn die Maus darüber steht, das Symbol per Tabulator den Fokus bekommt oder es angetippt wird; Esc schließt ihn. Die Texte erklären Fachbegriffe und Auswirkungen eines Feldes – etwa Brutto- und Restrisiko, RTO, RPO und MAZ, Meldefristen oder die Begründung in der SoA – und liegen deutsch und englisch vor. Buttons zeigen ihren Text ohne ⓘ direkt, wenn die Maus kurz darauf ruht – bei Symbolknöpfen und Aktionen wie Review oder Vererbung O→I ebenso wie bei Speichern oder Zurück. Die Module selbst sind dafür nicht geändert worden.

- **Auslieferung:** Ein Kontextprozessor in isms.py gibt jeder Seite über tooltips_modul.tooltips_js() nur die Texte mit, die zu ihr gehören. Das Skript in templates/template.html sucht die Felder über ihr name-Attribut und die Spalten über ihre sichtbare Überschrift.
- **Nur speichernde Formulare:** An Feldern erscheint das Symbol nur in Formularen mit method="post". Such- und Filterleisten bleiben frei.
- **Platz des Symbols:** Es hängt an der Beschriftung des Feldes – label for, Formgruppe, Tabellenzelle davor oder Überschrift darüber. Findet das Skript keine Beschriftung, steht es hinter dem Feld. Je Formular und Feldname gibt es ein Symbol.

© 2012–2026 WolfInfoSec GmbH

- **Buttons:** Zugeordnet über die sichtbare Beschriftung ohne Symbole davor und dahinter – ein vorangestelltes Symbol fällt weg –, bei reinen Symbolknöpfen über das Symbol (z. B. ✏ für Bearbeiten, ✕ für Entfernen). Der Text erscheint nach einer knappen halben Sekunde Hover, bei Tastaturfokus oder bei langem Drücken auf Touch-Geräten – ein langes Drücken löst den Button nicht aus. Ein vorhandenes title-Attribut wird im selben Stil gezeigt, ein Katalogtext geht vor. Erfasst werden nur Buttons im Inhaltsbereich, auch solche, die ein Skript später einfügt (KI-Assistenten).
- **Seitenleiste:** Jeder Menüeintrag und jede Menügruppe hat einen kurzen Text, der rechts neben dem Eintrag erscheint. Zugeordnet wird über den Modulschlüssel aus MODULE_REGISTRY (data-modul, gesetzt von isms.menu()) bzw. gruppe:<Schlüssel> aus MENU_GRUPPEN. In der eingeklappten Leiste steht der Modulname davor; der Tooltip ersetzt dort die bisherige Browser-Blase des Symbols.

Die Vorgabetexte stehen in tooltips_katalog.py auf drei Ebenen; die speziellere geht vor:

Ebene	Schlüssel	Gilt für
Allgemein	*	jede Seite, z. B. besitzer, abteilung_id, link
Modul	Blueprint-Name, z. B. risiken; Seiten ohne Blueprint: isms	alle Seiten des Moduls
Seite	Endpoint, z. B. bcm.bcm_bia_edit	nur diese Seite – wenn ein Feldname im selben Modul Verschiedenes meint

Ein Schlüssel der Form name@anderes gilt nur in Formularen, die auch ein Feld anderes enthalten, und geht dem schlichten name vor. So bekommt das Link-Formular (bezeichnung@link) einen anderen Text als die Maske einer neuen Maßnahme (bezeichnung), und ein Kommentarfeld namens text bleibt ohne Symbol.

Ein- und Ausschalten

- **Kachel:** Die Kachel „Tooltips“ unter Administration zeigt den Zustand („an“ bzw. „aus“) und schaltet mit einem Klick für alle Benutzer um (isms_settings, Schlüssel tooltips_aktiv, Vorgabe an).
- **Wirkung:** Jeder Prozess hält den Zustand und die Texte eine Minute vor. Danach gilt die Änderung für alle, ein Neustart ist nicht nötig.

Für neue Felder gehört der Text in tooltips_katalog.py, nicht ins Modul. Meint ein Feldname im selben Modul auf verschiedenen Masken Verschiedenes, hilft eine Seiten-Ebene oder ein Schlüssel name@anderes.

© 2012–2026 WolfInfoSec GmbH

14. Lieferantenfragebogen

Der Lieferantenfragebogen erhebt bei einem Lieferanten schriftlich, wie er Informationssicherheit umsetzt. Er besteht aus drei Teilen: einem Fragenkatalog, den der Administrator pflegt, dem Versand an den Lieferanten und der Auswertung, die am Lieferantendatensatz hängt. Implementiert ist er in lieferanten_fragebogen.py.

Die Bedienung – Versenden, Nachfassen, Antworten lesen – steht in der Benutzerdokumentation. Hier stehen die administrativen Voraussetzungen.

14.1 Fragenkatalog

Der Katalog wird unter /admin/vendor_fragebogen gepflegt. Beim ersten Aufruf spielt WolfISMS einmalig eine Vorlage nach ISO/IEC 27001:2022 und NIS2 (Art. 20, 21 und 23) ein: 49 Fragen in 12 Kategorien, jeweils mit Framework-Bezug und deutscher wie englischer Fassung.

Feld	Bedeutung
Reihenfolge	Sortierung im Fragebogen; die Vorlage vergibt Zehnerschritte, damit sich Fragen dazwischen einfügen lassen
Kategorie	Abschnittsüberschrift im Formular (deutsch und englisch)
Frage	Fragetext (deutsch und englisch)
Framework-Bezug	Freitext, z. B. „A.5.19" oder „NIS2 Art. 21"
Antworttyp	ja_nein (Ja / Teilweise / Nein / N/A), text, langtext oder datum
Pflicht	Pflichtfragen müssen im Formular beantwortet werden
Aktiv	Nur aktive Fragen kommen in neu versendete Fragebögen; bereits versendete bleiben unverändert

Die Vorlage wird genau einmal eingespielt.
Ist der Katalog leer und war er es immer (AUTO_INCREMENT steht auf 1), spielt WolfISMS die Vorlage ein. Löscht ein Administrator später alle Fragen, bleibt der Katalog leer – ein bewusst geleerter Katalog füllt sich nicht von selbst wieder. Über die Schaltfläche im Katalog lässt sich die Vorlage jederzeit erneut einspielen; der vorhandene Katalog wird dabei ersetzt.

14.2 Versand

Versendet wird aus der Detailansicht des Lieferanten. Der Lieferant erhält eine E-Mail mit einem Link auf ein öffentliches Formular (/lieferantenfragebogen) – ohne Anmeldung, geschützt durch einen Einmal-Token in der URL.

Voraussetzung	Details
SMTP	Ein eingerichteter Mailversand (/admin/smtp); ohne ihn steht nur der Weg über PDF oder EML zur Verfügung
ISMS_BASE_URL	Bestimmt die Adresse im Link. Ohne diese Umgebungsvariable entsteht ein relativer Pfad, den der Empfänger nicht aufrufen kann – dieselbe Variable nutzt der Aufgaben-Mailer
Externe Domain	Optional. Ist das ISMS für Lieferanten unter einer anderen Adresse erreichbar als für die eigenen Mitarbeiter, wird sie unter /admin/vendor_fragebogen

© 2012–2026 WolfInfoSec GmbH

Voraussetzung	Details
	hinterlegt (isms_settings-Schlüssel vendor_fragebogen_domain) und geht ISMS_BASE_URL vor. Ohne Pfadangabe hängt WolfISMS den Webpfad der Installation an
Lieferantendaten	E-Mail-Adresse und Ansprechpartner müssen am Lieferanten hinterlegt sein; fehlt eines von beidem, wird der Versand mit Begründung abgelehnt
Gültigkeit	30 Tage ab Versand (LINK_GUELTIGKEIT_TAGE). Danach zeigt das Formular den Bogen als abgelaufen; ein neuer Versand erzeugt einen neuen Token

Neben dem Link gibt es zwei Wege ohne Formular: den Fragebogen als PDF verschicken oder als EML-Datei erzeugen und aus dem eigenen Mailprogramm senden. Antworten, die auf diesen Wegen zurückkommen, trägt ein Mitarbeiter manuell ein; der Bogen trägt dann die Quelle „manuell" statt „link" und läuft nicht ab.

14.3 Datenbank

Die drei Tabellen legt das Modul beim ersten Laden selbst an – ein manuelles Schema-Update ist nicht nötig. Bestandsinstallationen werden über ALTER TABLE um die Felder der manuellen Erfassung ergänzt.

Tabelle	Inhalt
vendor_frage_katalog	Der Fragenkatalog: Reihenfolge, Kategorie, Frage, Framework-Bezug, Antworttyp, Pflicht, Aktiv – je deutsch und englisch
vendor_fragebogen	Ein versendeter oder erfasster Bogen: Lieferant, Token, Empfänger, Sprache, Status, Ablauf, Quelle, Anhang
vendor_fragebogen_antwort	Die Antworten. Frage, Framework-Bezug und Antworttyp werden mitkopiert, damit ein alter Bogen lesbar bleibt, wenn der Katalog sich später ändert

14.4 Betrieb und Sicherheit

- **Öffentliche Route:** /lieferantenfragebogen ist bewusst ohne Anmeldung erreichbar. Zugang gibt allein der Token; ohne gültigen Token zeigt die Seite nichts an.
- **Kein Rateraum:** Der Token umfasst 64 Zeichen aus secrets und ist je Bogen einmalig (UNIQUE-Index).
- **Ablauf:** Abgelaufene Token nehmen keine Antworten mehr entgegen. Der Datensatz bleibt bestehen – wer nicht geantwortet hat, ist ein Ergebnis, das in die Lieferantenbewertung gehört.
- **Rückwirkung auf die Bewertung:** Mit dem Eingang der Antworten setzt WolfISMS das Feld „Fragebogen" des Lieferanten auf „Ja" und rechnet den Risikoscore neu.
- **Löschen:** Ein Bogen lässt sich samt Antworten löschen; das Aktivitätenlog behält den Vorgang.

© 2012–2026 WolfInfoSec GmbH

15. Lieferanten: Import und Cloud-Exit-Strategie

Seit v3.93 lassen sich Lieferanten aus Excel oder CSV einlesen, und Lieferanten von Cloud- und SaaS-Typen bekommen eine Cloud-Exit-Strategie. Beides steckt in lieferanten.py; die Bedienung beschreibt die Benutzerdokumentation.

15.1 Import (/vendor/import)

- **Zugang**: users.vendor ≥ 2 und kein Nur-Lese-Modus; fremde Lieferanten ändert nur, wer nach der Besitzerregel darf
- **Dateien**: .xlsx oder .csv (Komma oder Semikolon), MAX_IMPORT_BYTES = 8 MB. Die Spaltenköpfe werden in beiden Sprachen erkannt, die Bewertungskriterien unter ihrer Beschriftung aus Risiko-Parameter (dazu die Vorgabetexte, falls die Mappe vor einer Umbenennung entstand)
- **Zuordnung**: bekannte Lieferantennummer → dieser Lieferant; ohne Nummer → der Lieferant mit genau diesem Namen; sonst Neuanlage, wenn die Option gesetzt ist. Nichts wird geraten: unklare Zeilen werden mit Grund übersprungen
- **Zwei Durchgänge**: erst Vorschau mit „alter Wert → neuer Wert“, geschrieben wird erst beim erneuten Hochladen mit „Übernehmen“
- **Gelesen, nicht übernommen**: Risiko, Risikoscore und Cloud-Exit-Strategie. Risiko und Score berechnet der Import mit derselben Funktion wie die Maske neu und schreibt die Score-Historie fort
- **Protokoll und Grenzen**: jede Neuanlage und Änderung steht im Aktivitätsprotokoll; die Mengengrenze der freien Version gilt

Der Excel-Export hat dafür zusätzliche Spalten bekommen: E-Mail, die sieben Bewertungskriterien als Klartext der Kurzbewertung und die Cloud-Exit-Strategie als Klartext („(fehlt)“ bei Cloud-Lieferanten ohne Strategie). Ein Export lässt sich unverändert wieder einlesen und ergibt „Keine Unterschiede zum Bestand“. Auswertungen, die die bisherige Spaltenzahl voraussetzen, sind anzupassen.

15.2 Cloud-Exit-Strategie

Spalte	Bedeutung
vendor_typ.cloud	TINYINT; 1 = Typ gilt als Cloud/SaaS. Gepflegt per Häkchen unter /admin/vendor_typen. Beim ersten Anlegen der Spalte werden Typen mit „Cloud“ oder „SaaS“ im Namen vorbelegt – in der Auslieferung „Cloud Anbieter“, nicht aber „Softwareanbieter“
vendor.cloud_exit	MEDIUMTEXT; die Strategie als TinyMCE-HTML, beim Speichern bereinigt; ein leerer Editor gilt als leer

Beide Spalten legt die Anwendung beim ersten Aufruf selbst an – über die Oberfläche wie über die REST-API. Die Liste zeigt die Zahl der Cloud-Lieferanten ohne Strategie samt Filter, der Excel-Export eine eigene Spalte, die REST-API die Felder cloud, cloud_exit und cloud_exit_fehlt (siehe REST-API-Dokumentation, Abschnitt Lieferanten).

15.3 Beschreibung und Detailansicht

Seit v4.00 hat jeder Lieferant ein Freitextfeld Beschreibung (Spalte vendor.beschreibung), gepflegt mit demselben Editor wie die Cloud-Exit-Strategie. Die Anwendung legt die Spalte beim ersten Aufruf selbst an. Das Feld steht in Bearbeiten-Maske und Detailansicht, im Excel-Export und in der REST-API (GET, Liste, POST, PATCH; Abschnitt Lieferanten der REST-API-Dokumentation).

© 2012–2026 WolfInfoSec GmbH

Der Block „Risikobewertung“ mit den sieben Bewertungskriterien erscheint seit v4.00 nur noch in der Bearbeiten-Maske; die Detailansicht zeigt Beschreibung und Cloud-Exit-Strategie ohne ihn.

© 2012–2026 WolfInfoSec GmbH

16. Auditmanagement

Das Audit-Modul (audit.py) führt Audits und ihre Abweichungen. Die Bedienung – Audit anlegen, Feststellungen erfassen, Maßnahmen ableiten – steht in der Benutzerdokumentation. Hier stehen die Umstellung von den früheren Fragenkatalogen, das Datenmodell, die normbezogene Prüfliste und die Ablage der Anhänge.

16.1 Umstellung von den Fragenkatalogen

Bis v3.71 war das Modul ein Normen-Selbstcheck: Kapitel-Checklisten zu ISO 27001, NIS-2, Cyber Essentials, VdS 10000, DSGVO und TISAX auf den Tabellen Audit, AuditNIS2, AuditCE, AuditVdS, AuditDSGVO und AuditTISAX. Mit v3.72 ist daraus ein Auditmanagement geworden. Die Checklisten sind ersatzlos entfallen – mit ihnen die Berichts- und Exportrouten je Katalog und die Audit-Diagramme.

Der Erfüllungsgrad je Framework wird nicht mehr an zwei Stellen gepflegt: Er steht im Controls-Modul, je Anforderung mit Anwendbarkeit, Umsetzungsstatus und Begründung, verdichtet in Kapitelübersicht und Konformitätsbericht.

Die alten Tabellen bleiben stehen.
Weder die Anwendung noch ein Migrationsskript fasst Audit, AuditNIS2, AuditCE, AuditVdS, AuditDSGVO oder AuditTISAX an. Die früheren Bewertungen bleiben vollständig in der Datenbank, werden aber von keiner Route mehr gelesen. Wer sie noch braucht, holt sie direkt aus der Datenbank oder aus einer Datensicherung – gelöscht wird nichts von selbst, und wer die Tabellen löscht, tut es bewusst.

16.2 Datenmodell

Vier Tabellen, die _ensure_tables() beim ersten Aufruf des Moduls per CREATE TABLE IF NOT EXISTS anlegt – kein Migrationsskript nötig.

Tabelle	Aufbau und Zweck
aud_audits	nummer (AUD-00001, fortlaufend), titel, beschreibung, datum_von / datum_bis, status (geplant | fest | durchgeführt), art (intern | extern | gap), auditor, norm, dazu Ersteller/erstellt und Aenderer/geaendert
aud_abweichungen	audit_id, nummer (ABW-00001), titel, beschreibung, typ (Empfehlung | Nebenabweichung | Hauptabweichung), status (offen | in Arbeit | umgesetzt | nicht umgesetzt), verantwortlich (users.id), faellig, norm, norm_punkt, massnahmen, risiken (seit v3.95)
aud_anhaenge	audit_id, original_name, stored_name, rel_path, mime_type, file_size, beschreibung, ersteller, erstellt – derselbe Spaltenschnitt wie mass_anhaenge, damit die Helfer aus isms.py unverändert greifen
aud_pruefpunkte	audit_id, norm_id (Verweis auf Normen.id), norm, kapitel, nummer, titel – die vier letzten als Kopie aus Normen –, bewertung (offen | erfüllt | weitgehend erfüllt | teilweise erfüllt | nicht erfüllt | nicht anwendbar), feststellung, nachweis, beschreibung (Prüffragen, von der KI oder von Hand), abweichung_id, sortierung

© 2012–2026 WolfInfoSec GmbH

Maßnahmen werden verknüpft, nicht kopiert.
Die Spalte massnahmen führt die IDs aus der Tabelle Massnahmen als Komma-Liste – dasselbe Muster wie im Schwachstellen- und im Incident-Modul. /audit/mass_neu legt eine Maßnahme an und verknüpft sie sofort; dafür braucht der Benutzer zusätzlich das Bearbeitungsrecht mass.
Risiken: Seit v3.95 führt die Spalte risiken die IDs verknüpfter Risiken. /audit/risiko_neu legt ein Risiko aus der Abweichung an (Audit-Bearbeitungsrecht und risk ≥ 2, Mengengrenze der Edition) und verknüpft es samt den Maßnahmen der Abweichung; eine neue Maßnahme an der Abweichung wandert in Risiko.massnahmen aller verknüpften Risiken, die der Benutzer nach der Besitzerregel ändern darf. Am Risiko erscheint die Abweichung über _RUECK_QUELLEN['audit'], der Excel-Export hat die Spalte „Risiken“.

16.3 Prüflisten und Gap-Assessments

Seit v3.76 kann ein Audit gegen ein Framework geführt werden: Die Anforderungen eines Frameworks aus der Tabelle Normen werden als Prüfpunkte in das Audit übernommen und dort einzeln bewertet. Ein Gap-Assessment ist deshalb kein eigener Objekttyp, sondern ein Audit der Art gap – dieselbe Prüfliste, dieselbe Auswertung, dieselben Abweichungen.

Die Prüfliste kopiert, sie verweist nicht.
Framework, Kapitel, Nummer und Titel werden beim Aufbau der Prüfliste in aud_pruefpunkte geschrieben. Ein Auditbericht muss den Wortlaut zeigen, gegen den damals geprüft wurde – auch wenn das Control später geändert oder gelöscht wird. Die Spalte norm_id bleibt zusätzlich stehen, damit die Prüfliste zum aktuellen Framework-Eintrag verlinken kann; der Verweis erscheint nur für Benutzer mit Leserecht auf Frameworks.

Die Tabelle Normen wird dabei ausschließlich gelesen. Ein Audit ändert nie den Umsetzungsstatus einer Anforderung: Der Umsetzungsstand des ISMS steht im Controls-Modul, das Prüfergebnis eines bestimmten Audits steht im Audit. Beides kann auseinanderlaufen, und genau das ist der Sinn eines Audits.

Bewertet wird fünfstufig:

Bewertung	Erfüllungsgrad	Bedeutung
erfüllt	100 %	Anforderung umgesetzt und nachgewiesen
weitgehend erfüllt	75 %	umgesetzt, mit kleineren Lücken im Nachweis oder in der Durchführung
teilweise erfüllt	50 %	im Ansatz vorhanden, wesentliche Teile fehlen
nicht erfüllt	0 %	nicht umgesetzt
nicht anwendbar	ohne	Scoping-Entscheidung – der Punkt geht nicht in die Wertung ein
offen	ohne	Zustand vor der Prüfung, keine Bewertung

Der Erfüllungsgrad ist der Mittelwert über die bewerteten Punkte. Offene und als nicht anwendbar gekennzeichnete Punkte bleiben außen vor – sonst zeigte eine halb ausgefüllte Prüfliste ein falsch schlechtes Bild. Angezeigt wird er in der Prüfliste, kapitelweise in der Detailsicht des Audits, als Spalte in der Auditübersicht und in beiden Excel-Exporten. Die Stufen stehen in audit.py in BEWERTUNG und BEW_PROZENT.

© 2012–2026 WolfInfoSec GmbH

Die Routen des Moduls dazu:

- **/audit/pruefliste_setup**: Framework und Kapitel wählen und die Anforderungen übernehmen. Wahlweise nur die als anwendbar gekennzeichneten (Normen.anwendbar = 1) und wahlweise die bestehende Prüfliste ersetzen. Ohne Ersetzen werden nur Anforderungen ergänzt, die noch fehlen – bereits vergebene Bewertungen bleiben also erhalten
- **/audit/pruefliste**: bewerten, mit Filter nach Kapitel, Bewertung und Freitext. Gespeichert wird das ganze angezeigte Formular auf einmal (/audit/pruefliste_save)
- **/audit/pruefliste_reset**: die Prüfliste eines Audits leeren. Die daraus entstandenen Abweichungen bleiben bestehen – sie sind Feststellungen des Audits und hängen nicht am Prüfraster
- **/audit/pruef_export**: die Prüfliste eines Audits als Excel-Mappe, mit Erfüllungsgrad in der Titelzeile

Die Reihenfolge der Prüfpunkte ist die Reihenfolge des Frameworks: sortierung übernimmt die id des Controls. Später ergänzte Kapitel stehen dadurch an ihrer fachlichen Stelle und nicht am Ende der Liste.

Aus einem Prüfpunkt lässt sich unmittelbar eine Abweichung anlegen. Übernommen werden Framework-Punkt und Anforderung als Bezeichnung, die Feststellung samt Nachweis als Beschreibung sowie der Framework-Bezug; der Typ wird aus der Bewertung vorgeschlagen (nicht erfüllt → Hauptabweichung, teilweise erfüllt → Nebenabweichung, sonst Empfehlung) und ist danach frei änderbar. Verantwortlicher und Frist bleiben leer – deshalb führt der Weg direkt in die Maske der neuen Abweichung. Prüfpunkt und Abweichung bleiben über abweichung_id verbunden: Wird die Abweichung gelöscht, fällt nur der Verweis, und am Prüfpunkt lässt sich erneut eine anlegen.

16.4 KI-Assistent und GAP-Assistent

Die Prüfpunkte tragen nur Framework, Kapitel, Nummer und Titel der Anforderung. Zwei Hilfen kamen dazu, beide nur für Audits der Art gap mit Prüfliste und für Benutzer mit audit ≥ 2 außerhalb des Nur-Lese-Modus.

- **KI-Assistent (audit_ki_modul.py)**: seit v4.00 eine Aufgabe des Audit-KI-Assistenten (/audit/ki_assistent), der diesen Kern nutzt; formuliert je Anforderung Prüffragen – Grundlage sind Titel und, soweit vorhanden, der Normtext aus Normen.Beschreibung – und schreibt sie ohne Übernahmeschritt in die Spalte aud_pruefpunkte.beschreibung, die audit.py bei Bedarf per ALTER TABLE nachrüstet. Er arbeitet in Blöcken zu 20 Punkten (höchstens 500 je Lauf) im Hintergrund über ki_jobs mit Fortschrittsmeldung; jeder Block wird sofort gespeichert. Vorhandene Beschreibungen bleiben, wenn „Überschreiben“ nicht angehakt ist. Routen: /audit/ki_beschreibung und /audit/ki_stand, Bereich audit
- **GAP-Assistent (gap_assistent.py)**: /audit/gap?audit=<id> zeigt die Übersicht, /audit/gap/schritt eine Anforderung mit Bewertung, Feststellung und Nachweis. Geschrieben wird in dieselben Felder von aud_pruefpunkte wie in der Prüfliste; eigene Antworten speichert er nicht
- **Fortschritt**: Tabelle aud_gap_fortschritt (audit_id, kapitel als SHA-1 des Kapitelnamens, status begonnen | erledigt | uebersprungen, position = Prüfpunkt, bei dem es weitergeht, benutzer). Über den Namen statt der Position bleibt der Stand gültig, wenn später Anforderungen dazukommen. „Fortschritt zurücksetzen“ leert nur diese Tabelle
- **Fehlende Prüffragen**: Ist die KI nutzbar, holt die Seite sie per /audit/ki_beschreibung für die gezeigte und die nächsten neun leeren Anforderungen in einem Aufruf; vorhandene werden dort nie überschrieben

© 2012–2026 WolfInfoSec GmbH

16.5 Anhänge und Links

- **Dateien**: liegen unter UPLOAD_BASE/audit/<audit_id>/ mit UUID-Präfix im Dateinamen. validate_pdf() prüft Endung, Magic Bytes und Größe (16 MB); ausgeliefert wird immer als ATTACHMENT_MIME, der vom Browser gemeldete Content-Type wird nie benutzt
- **Links**: stehen in der zentralen Tabelle Links mit bereich='audit' – wie bei Controls und Maßnahmen. safe_url() entschärft beim Anzeigen alles, was kein gewöhnliches Web-Schema ist
- **Löschen eines Audits**: entfernt Abweichungen, Prüfliste, Anhänge (samt der Dateien auf der Platte) und Links. Die verknüpften Maßnahmen bleiben bestehen – sie gehören dem Maßnahmenmodul

16.6 Berechtigung und Menü

Maßgeblich ist die Spalte users.audit (0 = kein Zugriff, 1 = Lesen, 2 = Bearbeiten); dasselbe Recht trägt die Managementbewertung. Menüeintrag, Modulrecht und Modul-Gate kommen aus isms_helpers.MODULE_REGISTRY (Schlüssel audit) – das Modul lässt sich also unter /admin/module abschalten, ohne dass Daten verloren gehen.

Seit v3.90 erzeugt die Detailansicht einen Auditbericht als PDF (/audit/<id>/pdf); seine Daten fließen außerdem in die Managementbewertung – siehe Kapitel Berichte und Meldewesen.

© 2012–2026 WolfInfoSec GmbH

17. Dokumentenlenkung und Versionen

Jede Änderung an einem Dokument legt eine neue Zeile in der Tabelle Dokumente an; der bisherige Stand bleibt unverändert stehen. Für die Anwender ist das in der Benutzerdokumentation beschrieben – hier steht, was dabei in der Datenbank passiert.

17.1 Datenmodell

Vier Spalten halten die Fassungen eines Dokuments zusammen. Sie kamen nachträglich dazu und werden beim ersten Zugriff angelegt; jede Abfrage spricht sie namentlich an, damit die Stellung in der Tabelle keine Rolle spielt.

Spalte	Bedeutung
stamm_id	ID der ersten Fassung. Jede Version der Reihe trägt sie, die erste zeigt auf sich selbst
vorgaenger_id	Die Fassung, aus der diese hervorgegangen ist
ist_aktuell	Genau eine Fassung je Reihe: die, auf die die Verknüpfungen aus den anderen Bereichen zeigen – die zuletzt freigegebene, und solange es keine gibt, die neueste
historisch	Abgelöste Zwischenstände. Sie sind aus allen Listen, Auswertungen und Aufgabenläufen ausgeblendet und nur noch über die Versionsliste des Dokuments erreichbar

Sichtbar bleiben je Reihe höchstens zwei Fassungen: die zuletzt freigegebene und, solange sie noch nicht freigegeben ist, die neueste. Alles dazwischen ist historisch. Die Versionsnummer selbst steht wie bisher in Version und Subversion – Version zählt bei der Freigabe hoch, Subversion bei jedem Speichern dazwischen.

17.2 Migration des Bestands

Beim ersten Aufruf des Dokumentenmoduls nach dem Update legt WolfISMS die vier Spalten an und ordnet den Bestand den Versionsreihen zu. Merkmal ist die Dokumentennummer: Sie war schon vorher das, was zusammengehörende Fassungen verbunden hat – die frühere Funktion „Neue Version" hat den Datensatz unter derselben Nummer angelegt. Die älteste Zeile einer Nummer wird zum Stamm, Zeilen ohne Nummer bleiben für sich. Kommentare ziehen dabei auf die Reihe um, damit ein Diskussionsfaden beim nächsten Speichern nicht abreißt. Ein Merker in isms_settings (dok_versionierung_migriert) sorgt dafür, dass das genau einmal läuft.

17.3 Verknüpfungen

Verknüpfungen gibt es in zwei Ausprägungen, und sie verhalten sich beim Versionswechsel unterschiedlich:

- **Am einzelnen Datensatz** hängen Anhänge (dok_anhaenge), Lesepflicht (dok_lesepflicht) und Links (Links). Sie werden bei jeder neuen Fassung mitkopiert, damit eine alte Version vollständig ist, wenn man sie wieder aufruft. Die Anhangsdatei selbst wird nicht dupliziert – mehrere Fassungen zeigen über rel_path und stored_name auf dieselbe Datei; gelöscht wird sie erst, wenn keine Fassung sie mehr führt.
- **An der Versionsreihe** hängen Normen.dokumente, Risiko.dokumente, Dokumente.dok_verknuepfung, prozesslandkarte.dokument_id und netzwerkzonen.dokument_id.

© 2012–2026 WolfInfoSec GmbH

Sie zeigen auf die freigegebene Fassung und bleiben dort stehen, solange an einer neuen gearbeitet wird.

Bei der Freigabe werden die Verweise der zweiten Gruppe in einem Zug auf die neue Fassung umgehängt, die Vorgänger auf Status 7 gesetzt und aus der Prozessverknüpfung genommen. Zeigte ein Datensatz bereits auf beide Fassungen, behält er den Verweis genau einmal. Wird eine Fassung gelöscht, rückt die verbliebene nach und übernimmt die Verweise; ist es die letzte der Reihe, werden sie entfernt statt ins Leere zeigen zu lassen.

Dokument↔Dokument: Dokumente.dok_verknuepfung steht weiterhin nur beim Dokument, an dem die Verknüpfung angelegt wurde. Seit v3.94 liest _dok_verknuepft_ids() zusätzlich die Rückverweise der Gegenseite; Seitenleiste, Verknüpfungsansicht und KI-Verknüpfungen zeigen beide Richtungen ohne Dubletten. Ein Verweis aus einer abgelösten Version zählt nicht, einer aus einem Entwurf erscheint als die gültige Fassung. Entknüpfen von der Gegenseite entfernt den Eintrag beim anlegenden Dokument; abgelöste Versionen bleiben als Nachweis unverändert. Beidseitig gespeichert wird bewusst nicht: ein Entwurf übernimmt die Liste seiner Vorversion, eine zwischenzeitlich auf der Gegenseite eingetragene Verknüpfung fehlte nach der Freigabe

Alte IDs bleiben gültige Adressen.
Ein Lesezeichen oder ein Link auf eine abgelöste Fassung führt weiterhin zu genau dieser Fassung – sie ist der Nachweis eines vergangenen Standes. Die Detailansicht weist darauf hin und führt zur geltenden Fassung. Geschrieben wird an einer abgelösten Fassung nicht: Bearbeiten, Verknüpfen, Anhänge und KI-Bearbeitung sind gesperrt, auch gegen einen direkt abgesetzten POST.

17.4 Auswirkung auf Listen und Auswertungen

Jede Abfrage, die den gültigen Bestand meint, blendet historische Fassungen aus. Dafür gibt es dok_aktuell_bedingung() in isms_helpers: Der Helfer liefert die WHERE-Bedingung und legt die Spalten bei Bedarf an. Betroffen sind Startseite und Statusdiagramme, Aufgaben und Aufgaben-Mails, Policy Portal, Personal, Risiken, Controls, Prozesslandkarte, Netzwerkzonen, Volltextsuche und der Dokumentenkontext der KI. Abfragen auf eine einzelne, namentlich bekannte ID brauchen die Bedingung nicht – eine historische Fassung soll über ihre ID ja gerade erreichbar sein.

17.5 Mengengrenze, Datensicherung und Import

- **Mengengrenze** der freien Version zählt Dokumentennummern, nicht Zeilen. Versionen verbrauchen daher kein Kontingent – auch mit zehn Dokumenten lässt sich weiterarbeiten.
- **Datensicherung**: die Tabelle Dokumente wächst mit der Versionierung deutlich, weil jede gespeicherte Änderung eine Zeile mit vollständigem Text ist. Das Backup umfasst sie unverändert; bei der Planung der Größe ist das einzukalkulieren.
- **Datenimport**: beim Import über Daten SQL-Export/Import werden stamm_id und vorgaenger_id verworfen – sie zeigen auf IDs der Quellinstanz. Die Zuordnung baut sich auf der Zielinstanz über die Dokumentennummer wieder auf.

17.6 Word-Import

Über dem Bearbeitungsformular eines Dokuments steht ein Kasten, der eine .docx-Datei entgegennimmt (POST /dokumente/word_import). Die Umwandlung nach HTML macht word_import.py mit python-docx; das Ergebnis geht zurück in das Formular und nicht in die

© 2012–2026 WolfInfoSec GmbH

Datenbank. Gespeichert wird erst mit dem regulären Speichern – über denselben Weg mit Prüfung, Sanitizer und Protokoll wie jede Eingabe von Hand.

Erzeugt werden ausschließlich Elemente, die isms.ALLOWED_TAGS zulässt; anschließend läuft derselbe Sanitizer wie beim Speichern über das Ergebnis. Die Vorschau im Editor zeigt damit genau das, was später in der Spalte steht. Bilder wandern als data:-URL in den Text – so hält sie auch der Editor, und der Word-Export macht daraus wieder eine Bilddatei.

Grenzen

Grenze	Wert und Ort
Dateigröße	16 MB (MAX_DOCX_MB in dokumente_modul.py). MAX_CONTENT_LENGTH taugt hier nicht als Schutz: der Wert steht wegen der Backup-Wiederherstellung sehr hoch, deshalb wird selbst gemessen
Einzelnes Bild	2 MB (BILD_MAX_BYTES in word_import.py)
Alle Bilder	8 MB (BILD_BUDGET). Die Aufblähung durch base64 zählt mit dem Faktor 4/3 gegen beide Grenzen
Umgewandelter Inhalt	16 MB – die Spalte Dokumente.Text ist ein MEDIUMTEXT
Dateityp	Nur .docx. Geprüft werden die Endung und die ZIP-Signatur PK\x03\x04; eine umbenannte .doc-Datei wird abgewiesen

Rechte und Sperren

- **Schreibrecht**: Der Import verlangt Schreibrecht auf Dokumente (dok ≥ 2) und ist für Nur-Lese-Zugänge gesperrt.
- **Dieselben Regeln wie beim Bearbeiten**: An einer abgelösten Fassung (historisch = 1), an einem schreibgeschützten Dokument und ohne Besitzer- oder Erstellerrecht wird abgewiesen – auch gegen einen direkt abgesetzten POST.
- **Mengengrenze**: Beim Anlegen eines neuen Dokuments greift die Mengengrenze der freien Version wie bei jedem anderen Weg.

Vollständigkeit und Betrieb

- **Was draußen bleibt**: Kopf- und Fußzeilen (der Briefkopf kommt aus branding.vorlage_path), das Inhaltsverzeichnis der Word-Datei, Text in Textfeldern und Bilder in Formaten, die ein Browser nicht anzeigt (EMF/WMF). Ausgelassene Bilder werden gezählt und gemeldet.
- **Vollständigkeitskontrolle**: Nach der Umwandlung vergleicht _fehlender_anteil() den Text aller w:t-Knoten mit dem sichtbaren Text des erzeugten HTML. Fehlen mehr als 10 Prozent, steht ein Hinweis über dem Formular – ein zu kurzer Import sähe sonst aus wie ein vollständiger.
- **python-docx** ist Pflicht (requirements.txt). Fehlt es, meldet das Modul „Word-Import nicht verfügbar“; die Dokumentenseite bleibt bedienbar.
- **Datensicherung**: Eingebettete Bilder machen Dokumententexte deutlich größer. Zusammen mit der Versionierung – jede Fassung trägt den vollen Text – ist das bei der Größe des Backups einzukalkulieren.

© 2012–2026 WolfInfoSec GmbH

18. On-/Offboarding

Das Personalmodul (personal.py) hat seit v3.72 einen zweiten Bereich: /personal/onboarding verfolgt Ein- und Austritte. Die Bedienung steht in der Benutzerdokumentation; hier stehen das Datenmodell und die Wirkung auf Benutzerkonten.

18.1 Datenmodell

Auch diese Tabellen legen sich beim ersten Aufruf selbst an (_ensure_onb_tables, CREATE TABLE IF NOT EXISTS).

Tabelle	Aufbau und Zweck
pers_onboarding	Der Vorgang: art (on | off), vorname, nachname, email, firmenemail, abteilung und abteilung_id, aufgabe, datum (Ein- oder Austrittsdatum), user_id (verknüpftes Konto aus users), die drei festen Punkte s_vertrag, s_mail und s_isms, bemerkung, abgeschlossen
pers_onb_zugaenge	Die Systemzugänge: vorgang_id, bezeichnung, benutzername (Konto), erledigt und erledigt_am, bemerkung, sortierung
pers_onb_equipment	Die Ausstattung – derselbe Aufbau; benutzername trägt hier die Inventar- oder Seriennummer

Die drei festen Punkte kennen drei Werte: 0 = offen, 1 = erledigt, 2 = nicht erforderlich (zählt für den Fortschritt nicht mit). s_vertrag zählt nur beim Onboarding. Der Punkt „ISMS-Benutzer" steht in keiner eigenen Spalte, sondern wird abgeleitet: beim Onboarding gilt er als erledigt, sobald user_id gesetzt ist; beim Offboarding erst, wenn users.aktiv des verknüpften Kontos auf 0 steht.

18.2 Benutzerkonten

Aus einem Vorgang heraus lässt sich das verknüpfte Konto deaktivieren (users.aktiv = 0). Angelegt werden Konten weiterhin ausschließlich in der Benutzerverwaltung – der Vorgang verweist nur darauf und übernimmt beim Verknüpfen die E-Mail-Adresse des Kontos, solange im Vorgang noch keine Firmen-E-Mail steht.

Dieselben Sicherungen wie in der Benutzerverwaltung.
Das Sperren folgt den Regeln von /admin/toggle_user_active: das eigene Konto sowie der jeweils letzte aktive Administrator und der letzte aktive Masteruser bleiben aktiv – sonst sperrt sich die Installation selbst aus.

18.3 Berechtigung und Protokoll

Maßgeblich ist die Spalte users.pers (1 = Lesen, 2 = Bearbeiten); Readonly-Benutzer schreiben auch mit Stufe 2 nicht. Ein abgeschlossener Vorgang ist gegen Änderungen gesperrt, bis er wieder geöffnet wird. Anlegen, Ändern, Abschließen, Wiederöffnen und Löschen stehen in der Aktivitätentabelle (siehe Kapitel Aktivitäten und Protokollierung).

© 2012–2026 WolfInfoSec GmbH

19. Schulungen (Personal)

Der dritte Reiter des Personalmoduls (seit v3.93, /personal/schulungen) verwaltet Schulungstermine und ihre Teilnehmer. Die Bedienung steht in der Benutzerdokumentation. Beide Tabellen legt personal.py beim ersten Aufruf selbst an (CREATE TABLE IF NOT EXISTS).

Tabelle	Aufbau und Zweck
pers_schulungen	Der Termin: titel (Pflicht), datum (Pflicht), ort, referent, beschreibung, ersteller/erstellt_am, aenderer/geaendert_am
pers_schulung_teilnehmer	Die Zuweisung: schulung_id, user_id, besucht (0/1), besucht_am, zugewiesen_am; eindeutig je Schulung und Benutzer

- **Berechtigung**: users.pers ≥ 1 zum Ansehen, ≥ 2 zum Anlegen, Zuweisen und Setzen der Teilnahme; Readonly-Benutzer schreiben nicht
- **Zuweisen**: „Alle Benutzer zuweisen“ nimmt nur aktive Konten (users.aktiv = 1); bereits Zugewiesene behalten ihren Status
- **Teilnahme**: besucht_am ist das Datum der Schulung. Offene Zuweisungen zu einem vergangenen Termin gelten als versäumt
- **Anzeige**: Personalliste (Spalte „Offene Schulungen“), Kacheln und Detailseite einer Person lesen dieselben Tabellen
- **Protokoll**: Änderungen stehen in den Aktivitäten als „Schulung-<id>“; jede Eingabe läuft vorher durch Pruefung

© 2012–2026 WolfInfoSec GmbH

20. Changemanagement

Das Modul Changemanagement (changemanagement_modul.py, /change) steuert Änderungen an Systemen und Konfigurationen nach ISO/IEC 27001 A.8.32: Antrag, Freigabe im Gremium, Umsetzung, Nachbetrachtung. Es steht in jeder Edition zur Verfügung und lässt sich unter /admin/module abschalten (Registry-Schlüssel change).

20.1 Berechtigung

- **Recht assets**: Das Modul hat keine eigene Spalte in users. Ein Change ist immer eine Änderung an Assets oder Konfigurationen – wer das Assetregister lesen darf, liest auch die Changes (1), wer es pflegt, pflegt sie (2).
- **Besitzer**: Bearbeiten, Statuswechsel, Gremium, Nachbetrachtung und Löschen setzen Schreibrecht und Besitz voraus – bei mehreren Besitzern genügt einer davon; Masteruser dürfen immer. Der Ersteller zählt – wie bei den Incidents – nicht automatisch mit.
- **Genehmiger**: Wer im Gremium benannt ist, gibt seine Stimme auch mit reinem Leserecht ab. Portal-Only-Konten dürfen nichts davon.

20.2 Datenmodell

Die drei Tabellen legt das Modul beim ersten Aufruf selbst an; ein Migrationsskript ist nicht nötig.

Tabelle	Inhalt
chm_changes	Der Änderungsantrag: Nummer CHG-NNNNN, Titel, Beschreibung, Begründung, Typ, Kategorie, Status, Priorität, Risiko und Auswirkung der Änderung, geplantes Zeitfenster und Ausfallzeit, Umsetzungs-, Test- und Rückfallplan, Besitzer und Umsetzer, Zeitstempel je Schritt sowie die Felder der Nachbetrachtung (pir_*). Verknüpfungen zu Assets, Risiken, Maßnahmen, Incidents und Dokumenten stehen als Komma-Liste von IDs im Datensatz
chm_freigaben	Das Gremium: eine Zeile je Genehmiger mit Stufe (1–3), Rolle, Votum (offen, zugestimmt, abgelehnt), Kommentar und Zeitpunkt. Ein Benutzer steht je Change nur einmal darin
chm_historie	Jeder Statuswechsel und jede Stimme: wer, wann, von welchem zu welchem Status, mit Bemerkung

20.3 Typen und Ablauf

Typ	Freigabe
Standard	Vorab genehmigte Routineänderung – kein Gremium, mit dem Einreichen genehmigt
Normal / Wesentlich	Gremium erforderlich; ohne benannten Genehmiger lässt sich der Change nicht einreichen
Notfall	Darf vor der Freigabe umgesetzt werden, schon aus dem Entwurf heraus; das Gremium stimmt nachträglich ab. Ein späteres Nein setzt den Change nicht zurück, sondern steht als Beanstandung in der Historie

© 2012–2026 WolfInfoSec GmbH

Schritt	Von	Nach
Einreichen	Entwurf	Eingereicht (Standard: Genehmigt)
Abstimmung	Eingereicht	Genehmigt, sobald alle zugestimmt haben; Abgelehnt bei der ersten Ablehnung
Umsetzung starten	Genehmigt (Notfall auch Entwurf, Eingereicht)	In Umsetzung
Umsetzung beendet	In Umsetzung	Umgesetzt
Zurückrollen	In Umsetzung, Umgesetzt	Zurückgerollt
Abschließen	Umgesetzt, Zurückgerollt	Abgeschlossen – nur mit erfasster Nachbetrachtung
Zurückziehen	Entwurf, Eingereicht	Zurückgezogen
Überarbeiten	Abgelehnt, Zurückgezogen	Entwurf – alle Stimmen werden auf offen gesetzt

- **Status nur über Schritte**: Der Status wird ausschließlich über /change/status gewechselt, nie über das Bearbeitungsformular. Welcher Schritt angeboten wird, hängt vom aktuellen Status ab.
- **Stufen**: Eine Stufe stimmt erst ab, wenn die darunterliegende vollständig zugestimmt hat. Das Gremium pflegt der Besitzer unter /change/cab.
- **Nachbetrachtung (PIR)**: Mit „Umsetzung beendet“ oder „Zurückrollen“ wird sie 14 Tage später fällig (PIR_FRIST_TAGE). Erfasst werden Datum, Ergebnis (erfolgreich, mit Einschränkungen, fehlgeschlagen …) und Bemerkung. Ohne sie lässt sich ein Change nicht abschließen – das ist der Nachweis, den ein Audit sehen will.
- **Löschen**: nimmt Gremium und Historie des Changes mit.

20.4 Aufgaben und Kalender

Drei Termine eines Changes erscheinen in /aufgaben und in der Erinnerungsmail: das geplante Zeitfenster beim Umsetzer (ersatzweise bei jedem Besitzer), eine offene Stimme, deren Stufe an der Reihe ist, beim Genehmiger – fällig spätestens zum Beginn des Zeitfensters – und die fällige Nachbetrachtung bei jedem Besitzer. aufgaben_mailer.py führt die Abfrage bewusst als eigene Kopie, damit er ohne die UI-Module auskommt. Der Änderungskalender (/change/kalender) zeigt alle Changes mit geplantem Beginn im Monatsraster.

© 2012–2026 WolfInfoSec GmbH

21. Projekte

Seit v4.00 hat das Projektmodul (projekte.py, /projekte) einen ISMS-Rahmen nach ISO/IEC 27001 A.5.8 „Informationssicherheit im Projektmanagement“: jedes Projekt führt Klassifizierung, Sicherheitsverantwortung, Sicherheitsanforderungen, Risikobewertung, Prüfpunkte im Ablauf und eine geordnete Übergabe mit Lessons Learned mit. Rahmen und Ampel liegen in projekte_isms.py. Das Modul teilt sich das Recht mass und gehört zur Edition Advanced und zu Enterprise.

Eigenschaft	Wert
Neue Spalten in Projekte	auftraggeber, sicherheitsverantwortung (Benutzer-IDs), klassifizierung, umfang, ablage, sicherheitsanforderungen, risikobewertung (+ _datum), ergebnis, uebergabe, lessons_learned, restrisiko (+ _datum), risiken, assets, dokumente, lieferanten – per ALTER TABLE beim ersten Aufruf; „Ziel und Nutzen“ bleibt in Bezeichnung (mit _en)
projekt_rollen	projekt_id, rolle, user_id oder person (Freitext für Externe), reihenfolge; vorgeschlagen Auftraggeber, Projektleitung, Sicherheitsverantwortung
projekt_gates	projekt_id, name, termin, kriterium, stand (offen, bestanden, bestanden mit Auflage, nicht bestanden), auflage, nachweis, reihenfolge; Vorschlag G0 Initiierung bis G3 Übergabe & Abschluss
Phasen	Planung (offen, in Planung), Umsetzung (genehmigt, in Umsetzung), Abschluss (umgesetzt, archiviert) – je Phase fragt die Ampel andere Abschnitte ab
ISMS-Ampel	pruefpunkte(): was in der aktuellen Phase fehlt – nur Hinweis, keine Sperre
Rechte	lesen mit mass ≥ 1, schreiben mit mass ≥ 2 ohne Nur-Lese-Zugang; ändern Besitzer oder Master

© 2012–2026 WolfInfoSec GmbH

22. Business Continuity Management

Das BCM-Modul (bcm_modul.py, /bcm, Recht bcm) führt die Business Impact Analyse (BIA), die Notfallpläne und die BCM-Übungen. Seit v3.84 hängen alle drei an einem Asset. Die frühere eigene Ebene „Kritische Prozesse“ ist entfallen: Die Wiederanlaufziele gehören an das Asset, und eine parallele Prozessliste im BCM war doppelt gepflegt. Geschäftsprozesse werden als primäre Assets geführt.

22.1 Datenmodell

Tabelle / Spalte	Bedeutung
Assets.rto_stunden / rpo_stunden / maz_stunden	Wiederanlaufziele in ganzen Stunden, NULL = nicht festgelegt. Feste Stufen 1, 4, 8, 24, 72, 168, 720; RPO zusätzlich 0 (kein Datenverlust). Die RTO darf die MAZ nicht übersteigen
bcm_bia.asset_id	Das bewertete Asset
bcm_bia.nummer	BIA-Nummer im Format BIA-NNNNN
bcm_notfallplaene.asset_id, bcm_uebungen.asset_id	Das Asset, für das der Plan gilt bzw. das geübt wurde
bcm_uebungen.massnahmen	Seit v3.95: verknüpfte Maßnahmen als Komma-Liste der IDs wie aud_abweichungen.massnahmen. Pflege über /bcm/uebung/massnahmen (bcm ≥ 2), neue Maßnahme zusätzlich mit mass ≥ 2; die Maßnahme zeigt die Übung über _RUECK_QUELLEN['bcm']
bcm_prozesse, prozess_id	Altbestand – bleibt lesbar stehen, wird aber nicht mehr gepflegt

Die Spalten am Asset legt assets_modul._ensure_spalten() an, die an den BCM-Tabellen die _ensure_*-Helfer in bcm_modul.py – beide idempotent und ohne Migrationsskript. Das BCM-Modul ruft die Asset-Prüfung selbst auf, weil es vor dem Assetmodul geöffnet werden kann.

22.2 Übernahme des Altbestands

- **Einmal je Serverprozess**: _ensure_en_columns_bia() ergänzt asset_id und nummer, macht prozess_id nullable und vergibt fehlende BIA-Nummern in id-Reihenfolge. migration_nummern.py kennt den Präfix BIA ebenfalls.
- **Keine automatische Zuordnung**: Aus einem früheren Prozess wird kein Asset geraten. Eine BIA ohne Asset zeigt „früher Prozess: …“; wer bearbeiten darf, sieht über der Liste, wie viele noch zuzuordnen sind.
- **RTO/RPO/MAZ neu erfassen**: Die Werte der früheren Prozesse werden nicht übernommen – sie gehören an das jeweilige Asset und werden dort eingetragen.
- **Alte Adressen**: /bcm/prozess, /bcm/prozess/neu und /bcm/prozess/edit leiten auf die BIA-Übersicht um; /bcm öffnet direkt die BIA.
- **bcm_prozesse nicht löschen**: Die Tabelle trägt die Namen, über die Altbestände nachvollziehbar bleiben.

© 2012–2026 WolfInfoSec GmbH

22.3 Auswertung und Schnittstellen

- **BIA-Score**: Mittelwert aller 20 Bewertungen (vier Dimensionen finanziell, operativ, regulatorisch, Reputation über die Zeitschritte 1 h, 4 h, 24 h, 72 h, 1 Woche; Skala 1–4). Ab 3,5 kritisch, ab 2,5 hoch, ab 1,5 mittel.
- **Primäre Assets ohne BIA**: werden als Kennzahl gezählt. Sekundäre Assets erben ihre Anforderungen von den primären und brauchen keine eigene BIA.
- **Asset-Export und -Import**: Die Mappe führt RTO, RPO und MAZ als eigene Spalten („4h“, „3d“); der Import liest dieselbe Schreibweise und prüft RTO ≤ MAZ.
- **REST-API**: Der Bereich Assets liefert und nimmt rto_stunden, rpo_stunden und maz_stunden, filtert über rto_max/rpo_max/maz_max und sortiert danach – siehe Dokument „WolfISMS REST-API“.

© 2012–2026 WolfInfoSec GmbH

23. Playbooks

Das Playbook-Modul (playbooks_modul.py, /playbooks) wurde mit v4.00 neu aufgebaut – nach dem Rahmen aus ISO/IEC 27035 und NIST SP 800-61, ISO/IEC 27001 A.5.24–A.5.28, NIS-2 Art. 23 und DSGVO Art. 33/34. Ein Playbook beschreibt, wie auf ein Szenario reagiert wird (Ransomware, Phishing, Ausfall eines Dienstleisters, meldepflichtiger Vorfall). Datenmodell, Ampel und Darstellungsbausteine liegen in playbooks_isms.py, den KI-Assistenten beschreibt der Abschnitt KI-Assistenten der übrigen Fachmodule. Das Modul gehört zur Edition Advanced und zu Enterprise.

Eigenschaft	Wert
Routen	/playbooks[?pbdetail=<id>] (Liste bzw. Detail), /playbooks/bearbeiten[?id=] (neu und ändern), /playbooks/link?id=&view= (Verknüpfungen), /playbooks/zuordnen (Datei oder Link zuordnen), /playbooks/loeschen (POST); /playbooks/ansicht leitet auf die Detailansicht um
Tabellen	playbook_texte (Steckbrief und Abschnitte, um Spalten per ALTER TABLE ergänzt), playbook_schritte, playbook_rollen, playbook_tests – legt das Modul beim ersten Aufruf selbst an. Dateien bleiben in playbooks (neu playbook_id), Links in Links (bereich playbooks, bereich_id); play_upload und play_attachment liegen weiter in isms.py
Steckbrief	nummer (Vorschlag PB-001), art (Incident Response, Notfall und Wiederanlauf, Meldung, Betrieb), Beschreibung als Szenario, schweregrad (ab niedrig/mittel/hoch/kritisch), besitzer, stellvertretung, freigabestand (Entwurf, in Prüfung, freigegeben, in Überarbeitung), version, freigabe_datum, freigegeben_von, naechste_pruefung, archiviert
Abschnitte	zweck, ausloeser, erstbewertung, eindaemmung, beseitigung, wiederherstellung, kommunikation, nachbereitung (MEDIUMTEXT, je mit Hinweis); der bisherige Freitext inhalt bleibt als „Weitere Inhalte“
Freigabe	beim Wechsel auf „freigegeben“ setzt freigabe_werte() Datum und Freigebende/n, die nächste Überprüfung auf zwölf Monate (wenn leer oder vorbei) und die Version auf 1.0 (wenn leer)
Ampel	pruefpunkte(): Verantwortliche und Stellvertretung, Art und Szenario, alle acht Abschnitte, mindestens drei Ablaufschritte, jeder mit Rolle, Rollen besetzt, freigegeben, nächste Überprüfung, Test in den letzten zwölf Monaten. Rot bei überfälliger Überprüfung oder nicht erfolgreichem letztem Test – nur Hinweis, keine Sperre
Verknüpfungen	ID-Listen in playbook_texte: assets, risiken, incidents, dokumente, lieferanten, behoerden, dienstleister (Kontakte), massnahmen
Rechte	lesen mit play ≥ 1 oder Admin, schreiben mit play ≥ 2 ohne Nur-Lese-Zugang; ein bestehendes Playbook ändern und löschen Verantwortliche, Stellvertretung, der Ersteller (solange keine Verantwortlichen benannt sind) oder Master
Löschen	entfernt Schritte, Rollen und Tests; zugeordnete Dateien und Links bleiben und stehen danach wieder ungeordnet in der Liste
Eingaben	Abschnitte werden wie Dokumente bereinigt (sanitize); Pflichtfelder und Längen prüft isms_helpers.Pruefung vor dem SQL

© 2012–2026 WolfInfoSec GmbH

24. Weitere Neuerungen im Datenmodell (v4.00)

Die folgenden Änderungen legen die Module beim ersten Aufruf selbst an; ein Eingriff ist nicht nötig. Sie stehen hier, damit beim Blick in die Datenbank klar ist, woher die Spalten kommen.

Bereich	Änderung
Rechtsregister	fester Feldsatz mit zehn Feldern (Nummer, Rechtsgrundlage, Kategorie, Anforderung, Anwendbarkeit, Umsetzung mit verknüpften Maßnahmen/Dokumenten/Controls, Verantwortlich, Status, offene Punkte mit Frist, letzte/nächste Prüfung). Neue Spalten u. a. fundstelle, anwendbarkeit (+ _en), massnahmen_ids, dokumente_ids, normen_ids, verantwortlich_id, frist, naechste_pruefung. Die Altspalten relevanz, bereiche, verantwortlicher, dokumentenreferenz und faelligkeit übernimmt der Prozess, der die Spalte anwendbarkeit anlegt, genau einmal; danach werden sie nicht mehr geschrieben
Verwandte Controls	Normen.verwandte_normen (kommagetrennte IDs), immer auf beiden Seiten gepflegt (normen_mapping.py); Detail, Verknüpfungsansicht und Abdeckung je Norm zeigen sie
ISO/IEC 27001-Fassungen	Bericht, SoA, „SoA als Dokument ablegen“ und das Statusdiagramm arbeiten mit jeder eingespielten Fassung (Norm ISO/IEC 27001:*), standardmäßig mit der neuesten; ?fassung= wählt eine andere. Sind mehrere eingespielt, schaltet ein Umschalter im Kopf um. Das Paket normen_ISO27001_2024.tar.gz enthält DIN EN ISO/IEC 27001:2024-01 mit Nummern und Bezeichnungen, ohne Normtext
Maßnahmen ↔ Assets	Massnahmen.assets (ID-Liste); ein Asset zeigt seine Maßnahmen direkt und über Risiken, Schwachstellen, Changes und Incidents
Operational Plan	Status je offener Instanz offen → in Bearbeitung → erledigt; erst „erledigt“ archiviert die Instanz als Nachweis und legt den nächsten Termin an – bei stark überfälligen Aufgaben ab heute gerechnet
Schwachstellen	swm_fristen – Behebungsfristen je Schweregrad, siehe Risiko-Parameter
Lieferantenbewertung	„Früherer Sicherheitsvorfall beim Lieferanten?“ war verdreht gewichtet, und das Tier wirkte nicht auf den Score. Beides wird einmal je Installation korrigiert, danach Stufe und Score aller Lieferanten neu berechnet; selbst gepflegte Gewichte bleiben unberührt
KI-Dialoge	ki_dialog trägt je Benutzer eine Zeile je Assistent (Bereich), ki_jobs die Hintergrundaufträge aller Assistenten

© 2012–2026 WolfInfoSec GmbH

25. Datenschutz

Seit v4.01 hat WolfISMS ein Datenschutzmodul (datenschutz_modul.py, Menüpunkt „Datenschutz“ in der Gruppe Governance & Compliance, /datenschutz). Es führt das Verzeichnis von Verarbeitungstätigkeiten nach Art. 30 DSGVO, die Schwellwertanalyse und die Datenschutz-Folgenabschätzung nach Art. 35 und die Angaben zur Auftragsverarbeitung nach Art. 28 in der Lieferantenakte. Datenmodell und Regeln liegen in datenschutz_isms.py, die DSFA in datenschutz_dsfa.py, der AVV-Block in datenschutz_avv.py. Das Modul gehört zu den Kernmodulen und steht in jeder Edition zur Verfügung, auch in der freien Version.

25.1 Berechtigung

Neues Modulrecht datenschutz (Spalte users.datenschutz, 0 = kein Zugriff, 1 = Lesen, 2 = Bearbeiten, Kürzel DSC in Rechte-Matrix und Excel-Export). Die Spalte legt isms_helpers.kartenrechte_sicherstellen() mit 0 an – auch für vorhandene Benutzer, eine Übernahme wie bei Prozesslandkarte und Netzwerkzonen gibt es nicht. Der Erstbenutzer einer neuen Installation bekommt Stufe 2. Administratoren dürfen immer lesen und bearbeiten. Ändern und Löschen eines Eintrags dürfen wie in den anderen Modulen Besitzer und Masteruser; die Muss-Liste pflegen nur Administratoren.

25.2 Datenmodell

Tabelle	Inhalt
ds_vvt	eine Zeile je Verarbeitung: nummer (VVT-…), typ (verantwortlicher | auftragsverarbeiter), titel, beschreibung, zweck, rechtsgrundlagen (Schlüssel 6a…9j, mit „|“ getrennt), interesse, betroffene, datenkategorien, besondere, Löschfrist (loesch_zahl, loesch_einheit, loesch_ausloeser), gemeinsam verantwortlich (+ Partner), abteilung_id; als Auftragsverarbeiter auftraggeber_id (Kundenkontakt), kategorien_av, avv_datum, avv_dokument; Verknüpfungen assets, controls, massnahmen, unterauftrag als ID-Listen; status, review; Schwellwertanalyse swa_muss, swa_kriterien, swa_ergebnis, swa_begruendung, swa_von, swa_am; Besitzer, Ersteller, Aenderer
ds_vvt_empfaenger	Empfänger je Verarbeitung: art (intern, lieferant, behoerde, sonstige), lieferant_id oder abteilung_id oder name, land (ISO-Code), garantie (Angemessenheitsbeschluss, Data Privacy Framework, Standardvertragsklauseln, BCR, Art. 49, keine). Ein Land außerhalb von EU und EWR gilt als Drittland
ds_mussliste	Muss-Liste nach Art. 35 Abs. 4: nr, text, beispiel, quelle, aktiv. Beim ersten Anlegen mit den 17 Einträgen der DSK-Liste (Version 1.1 vom 17.10.2018) belegt, danach pflegt die Installation sie selbst
ds_dsfa	eine Zeile je Folgenabschätzung: nummer (DSFA-…), vvt_id, titel, status, die Abschnitte beschreibung, notwendigkeit, abhilfe, stellungnahme, standpunkt (MEDIUMTEXT), ergebnis (tragbar | konsultation), ergebnis_text, massnahmen, controls, stellungnahme_von/_am, abgeschlossen_am, review
ds_dsfa_risiken	Risiken aus Sicht der Betroffenen: quelle, schaden, w und s (je 1–4), rest_w, rest_s, massnahmen, risiko_id (Eintrag im Risikoregister nach der Übergabe)
ds_avv	AVV-Angaben je Lieferant (lieferant_id = vendor.id): avv, datum, dokument_id, drittland, land, garantie, tia_dokument, bemerkung
ds_unterauftrag	Unterauftragsverarbeiter je Lieferant: name, leistung, land, ul_lieferant_id (falls selbst im Lieferantenregister)

© 2012–2026 WolfInfoSec GmbH

Alle Tabellen legen die Module beim ersten Aufruf selbst an (einmal je Serverprozess); ein Eingriff ist nicht nötig. Sie sind Teil der normalen Datensicherung.

25.3 Routen

Route	Zweck
/datenschutz	Liste als Verantwortlicher; ?typ=auftragsverarbeiter, ?typ=dsfa schalten um, ?vvtdetail=<id> zeigt einen Eintrag
/datenschutz/edit	Anlegen und Bearbeiten (?id=, ?typ=)
/datenschutz/swa	Schwellwertanalyse einer Verarbeitung (?id=)
/datenschutz/status, /empfaenger, /link, /delete/<id>	Status setzen, Empfänger, Verknüpfungen, Löschen (POST)
/datenschutz/dsfa…	Folgenabschätzung: neu (aus einer Verarbeitung als Verantwortlicher), Detail, edit, risiko, status, link, delete
/datenschutz/avv	AVV-Block der Lieferantenakte speichern (POST)
/datenschutz/mussliste	Muss-Liste pflegen – Einträge ergänzen, ab- und einschalten; nur Administratoren
/datenschutz/export	Excel-Export je Verzeichnis (?typ=)
/datenschutz/ki_assistent, /ki_tom_download	KI-Assistent und Download der TOM-Anlage

25.4 Regeln

- **Statuskette Verarbeitung:** Entwurf → in Prüfung → Freigegeben. Als Verantwortlicher lässt sich ein Eintrag erst freigeben, wenn Zweck, mindestens eine Rechtsgrundlage und eine Schwellwertanalyse mit Ergebnis und Begründung vorliegen; als Auftragsverarbeiter braucht er einen Auftraggeber. Ist bei der Freigabe keine Wiedervorlage gesetzt, wird sie auf heute + 12 Monate gesetzt
- **Schwellwertanalyse:** regel_ergebnis() schlägt „DSFA erforderlich“ vor, sobald ein Eintrag der Muss-Liste getroffen ist oder mindestens zwei der neun Kriterien aus WP 248 Rev. 01 zutreffen. Das Ergebnis setzt der Mensch; weicht es vom Vorschlag ab, zeigt die Seite das an
- **Hinweise:** Datenkategorien nach Art. 9/10 ohne Rechtsgrundlage aus Art. 9 und Empfänger im Drittland ohne Garantie werden markiert
- **Statuskette DSFA:** Entwurf → in Bearbeitung → Stellungnahme DSB → Abgeschlossen. Score = Wahrscheinlichkeit × Schwere (je 1–4 nach DSK-Kurzpapier Nr. 18), immer berechnet. Das Restrisiko zählt nur umgesetzte Maßnahmen. Abschließen erst mit Stellungnahme, Ergebnis und Restwerten je Risiko; die Stellungnahme schließt jeder mit Recht datenschutz = 2 ab
- **Überprüfung:** ändert sich die Verarbeitung nach dem Abschluss der DSFA, zeigt die DSFA, dass sie zu überprüfen ist (Art. 35 Abs. 11)
- **Übergaben:** ein DSFA-Risiko geht über /risiko/ki_assistent?ausgang=dsfa:<id> ins Risikoregister (risiko_id verweist danach dorthin), Abhilfemaßnahmen legt der Maßnahmen-Assistent mit Ausgang „DSFA“ an und verknüpft sie
- **AVV-Block:** erscheint in der Detailansicht eines Lieferanten, sobald das Kriterium „Personendaten“ auf Ja oder Teilweise steht oder schon AVV-Daten vorliegen. Lesen darf, wer die Lieferantenakte sieht; gepflegt wird er mit datenschutz = 2, nicht mit dem Lieferanten-Recht – so pflegt der Datenschutzbeauftragte den AVV, ohne die übrigen Lieferantendaten ändern zu dürfen

© 2012–2026 WolfInfoSec GmbH

25.5 KI-Assistent für den Datenschutz

datenschutz_ki_assistent.py (Aufgaben in datenschutz_ki_neu.py), Bereich datenschutz in ki_dialog und in der Verbrauchsauswertung. Recht: datenschutz ≥ 2 oder Administrator, kein Nur-Lese-Zugang, dazu „KI-Nutzung“. Aufgaben: Verarbeitung anlegen, als Auftragsverarbeiter erfassen, Lücken aus den Assets (Assets mit Personenbezug ohne Verarbeitung – nur auf Knopfdruck), Eintrag prüfen, Schwellwertanalyse vorschlagen, DSFA entwerfen, TOM-Anlage nach Art. 32 Abs. 1 aus Controls und Maßnahmen als Word-Datei.

Keine Rechtsberatung
Rechtsgrundlage, Löschfrist und Ergebnis der Schwellwertanalyse bleiben Vorschläge. Der Assistent setzt nie „Freigegeben“, schließt keine DSFA ab, schreibt keine Stellungnahme des DSB und legt weder Assets noch Lieferanten an.

© 2012–2026 WolfInfoSec GmbH

26. Berichte und Meldewesen

Mit v3.90 kamen mehrere Berichte, Exporte und der Meldeassistent für Sicherheitsvorfälle hinzu. Dieses Kapitel beschreibt Aufbau, Tabellen und Voraussetzungen; die Bedienung steht in der Benutzerdokumentation, Kapitel Berichte und Incidents.

26.1 Berichtsübersicht (/berichtsuebersicht)

berichte_modul.py stellt die Seite bereit. Sie steht mit dem Schlüssel berichte und recht None in isms_helpers.MODULE_REGISTRY und hat damit einen Menüpunkt für alle schreibenden Benutzer; reine Portal-Benutzer sehen ihn nicht. Jede Karte gehört zu einem Modul der Registry und erscheint nur, wenn modul_aktiv() für dieses Modul zutrifft und der Benutzer dessen Modulrecht hat. Die Auswahl einzelner Audits, Managementbewertungen und Übungen liest die jüngsten Einträge direkt aus aud_audits, mgmt_review und bcm_uebungen; fehlt eine Tabelle, entfällt nur die Auswahl. Die frühere Übersicht /berichte im Controls-Modul leitet hierher weiter, ihre Einzelberichte (?berdetail=1/2/3) bleiben.

26.2 PDF-Berichte

Bericht	Adresse	Datei
Auditbericht	/audit/<id>/pdf, Kurzfassung ?pruefliste=0	audit.py
BCM-Übungsbericht	/bcm/uebung/<id>/pdf	bcm_modul.py
Kennzahlenbericht	/ziele/bericht?von=JJJJ-MM-TT&bis=JJJJ-MM-TT	ziele_modul.py
Awareness-Bericht	/personal/awareness	personal.py

Alle vier entstehen über isms_helpers.pdf_briefpapier() und pdf_erzeugen(): HTML wird mit WeasyPrint gerendert, ein unter den Einstellungen hinterlegtes Briefpapier (isms_settings pdf_briefpapier_path und die vier Randwerte pdf_briefpapier_margin_*) wird unter jede Seite gelegt, Emoji werden entfernt. Scheitert nur das Briefpapier, geht der Bericht ohne hinaus. Voraussetzung ist WeasyPrint auf dem Server – dieselbe wie für die Managementbewertung. Ein Export steht im Aktivitätenprotokoll des Audits.

26.3 Excel-Exporte

Adresse	Tabellenblätter	Recht
/schwachstellen/export	Patchstand, Schwachstellen	schwachstellen
/change/export	Kennzahlen, Changes	assets
/bcm/export	Abdeckung, BIA, Notfallpläne, Übungen	bcm
/ziele/export?von=…&bis=…	Übersicht, Werte	ziele

Aufgebaut werden sie mit isms_helpers.xlsx_blatt() und xlsx_antwort() (openpyxl): Titelzeile in der Modulfarbe, fixierte Kopfzeilen, Autofilter, Datumswerte als echte Datumszellen. Die Exporte aus Modullisten übernehmen deren Filterparameter. Die älteren Exporte (Incidents, Assets …) sind unverändert.

© 2012–2026 WolfInfoSec GmbH

26.4 Meldeassistent (meldungen_modul.py)

Der Blueprint meldungen hängt am Registry-Eintrag incidents und wird mit dem Incident-Modul ein- und ausgeschaltet. Welche Regelwerke gelten, bestimmen weiter die Spalten DSGVO, NIS2 und CRA der Tabelle Incidents; Incidents selbst bleibt unverändert. Zwei neue Tabellen legen sich beim ersten Aufruf selbst an:

Spalte	Typ	Bedeutung
incident_id, regelwerk	INT, VARCHAR(10)	Vorfall und Regelwerk (NIS2, DSGVO, CRA); eindeutig je Paar
kenntnis	DATETIME	Zeitpunkt der Kenntnis, ab dem die Fristen laufen
entscheidung	VARCHAR(30)	offen, meldepflichtig oder nicht meldepflichtig
begruendung	TEXT	Pflicht bei „nicht meldepflichtig“
Aenderer, geaendert	INT, DATETIME	letzte Änderung

Tabelle inc_meldepflicht – die Bewertung je Vorfall und Regelwerk.

Spalte	Typ	Bedeutung
incident_id, regelwerk, stufe	INT, VARCHAR	Vorfall, Regelwerk und Meldestufe (fruehwarnung, meldung, zwischen, abschluss, betroffene)
gemeldet_am	DATETIME	Zeitpunkt der Meldung
stelle, aktenzeichen	VARCHAR(200), VARCHAR(100)	Empfänger und Vorgangsnummer der Behörde
inhalt	MEDIUMTEXT	übermittelter Text
Ersteller, erstellt	INT, DATETIME	wer die Meldung erfasst hat

Tabelle inc_meldungen – je abgegebene Meldung eine Zeile.

- **Fristen:** Stunden ab Kenntnis (24 h bzw. 72 h); die Abschlussmeldung rechnet einen Kalendermonat ab der tatsächlichen 72-Stunden-Meldung, ersatzweise ab deren Fristende. Ohne Bewertung gilt der Erkennungstag 00:00 Uhr. Die Stufen stehen in REGELWERKE am Kopf der Datei.
- **Erledigt:** eine Stufe, sobald eine Meldung dieser Stufe erfasst ist; liegt sie nach der Frist, gilt sie als verspätet.
- **Nachführen:** Beim Erfassen einer Meldung werden Meldungwann, Meldungwem, Änderer und geändert des Vorfalls gesetzt, damit Liste, Export und REST-API stimmen; eine offene Bewertung wird zu meldepflichtig.
- **Übersicht:** offene_fristen() liefert alle offenen Pflichtstufen, archivierte Vorfälle ausgenommen – für die Liste /incidents/meldefristen, die Leiste auf der Incident-Startseite und die Berichtsübersicht.
- **Aufgaben:** seit v3.94 liefert aufgaben_meldefristen(tage) die offenen und überfälligen Pflichtstufen, deren Frist innerhalb der nächsten Tage endet – je Besitzer des Vorfalls ein Posten (Hauptbesitzer und objekt_besitzer), ohne Besitzer nur in der Ansicht „Alle“ der Master-Benutzer. /aufgaben zeigt sie in einer eigenen Sektion samt Kachel, wenn das Incident-Modul aktiv ist und users.incidents ≥ 1
- **Rechte:** Lesen mit users.incidents ≥ 1; Bewerten, Erfassen und Löschen mit Schreibrecht und besitzer_check() auf den Vorfall. Jede Aktion steht im Aktivitätenprotokoll unter der Vorfallnummer.

© 2012–2026 WolfInfoSec GmbH

26.5 Protokoll der Lesebestätigungen (dok_lesebestaetigung)

users.dok_read hält nur, welche Dokumente ein Benutzer bestätigt hat. Seit v3.90 schreibt die Bestätigung im Policy Portal zusätzlich eine Zeile (user_id, dok_id, gelesen_am) in die Tabelle dok_lesebestaetigung; der Awareness-Bericht liest daraus die jüngste Bestätigung (isms_helpers.lesebestaetigungen_map()). Frühere Bestätigungen haben keinen Eintrag und erscheinen ohne Datum. Das Zurücksetzen des Lesestatus löscht das Protokoll bewusst nicht – es ist die Historie, der aktuelle Stand bleibt dok_read. Ein Fehler beim Protokollieren verhindert die Bestätigung nie.

26.6 Audit-Daten in der Managementbewertung

audit.mr_auswertung(von, bis) liefert der Managementbewertung zwei Listen: die Audits, deren Zeitraum sich mit dem Betrachtungszeitraum überschneidet (ohne Enddatum zählt der Beginn), und die Nichtkonformitäten – Haupt- und Nebenabweichungen, die entweder offen sind oder aus einem Audit des Zeitraums stammen. Empfehlungen zählen nur als Zahl beim Audit. Die Daten werden wie die übrigen Live-Abschnitte bei jedem Öffnen und Export neu berechnet. Ist das Audit-Modul abgeschaltet oder nicht lizenziert, entfallen beide Abschnitte, die Freitexte bleiben.

Datensicherung und SQL-Export
Die Datensicherung erfasst inc_meldepflicht, inc_meldungen und dok_lesebestaetigung wie jede Tabelle. Der bereichsweise SQL-Export (Kapitel Daten SQL-Export/Import) überträgt sie nicht.

© 2012–2026 WolfInfoSec GmbH

27. Risiko-Parameter

Administratoren konfigurieren die automatische Risikoberechnung über /admin/risiko_parameter. Das Modul verwaltet Vendor-Gewichte (Lieferantenbewertung) und Risikostufen-Schwellenwerte (alle Risikoarten).

27.1 Vendor-Gewichte

Für jeden Lieferanten werden Fragen in zwei Gruppen bewertet. Jede Antwort (Ja / Teilweise / Nein) ergibt Punkte. Der Durchschnitt je Gruppe liefert den Wert 1–5 für Eintrittswahrscheinlichkeit bzw. Schadenshöhe.

Gruppe	Standardfrage	Ja	Teilw.	Nein
Eintrittswahrsch.	Zertifizierung vorhanden? (ISO 27001, SOC 2 etc.)	0	2	5
Eintrittswahrsch.	Sicherheitsfragebogen ausgefüllt?	0	2	5
Eintrittswahrsch.	Vertragliche Regelung / AVV vorhanden?	0	2	5
Eintrittswahrsch.	Früherer Sicherheitsvorfall beim Lieferanten?	5	2	0
Schadenshöhe	Verarbeitung personenbezogener Daten (DSGVO)?	5	2	0
Schadenshöhe	Hoher Schutzbedarf der Systeme / Daten?	5	2	0
Schadenshöhe	Tier-Kritikalität (1 = unkritisch, 3 = kritisch)	2	4	5
Schadenshöhe	Rechtliche / Compliance-Anforderungen verletzt?	5	2	0

Die Tabelle vendor_gewichte wird beim ersten Aufruf von /admin/risiko_parameter angelegt, falls sie noch nicht existiert. Änderungen wirken sofort auf neu berechnete Lieferantenrisiken. Jede Neuberechnung wird zusätzlich in vendor_historie fortgeschrieben, sodass sich die Entwicklung eines Lieferantenrisikos nachvollziehen lässt.

Gewicht des Tiers: Seit v4.01 zählt das Tier in der Schadenshöhe doppelt – (Punkte der drei übrigen Schadensfragen + 2 × Tier-Punkte) / 5, gerundet – und setzt eine Untergrenze von Tier-Punkten − 1: missionskritisch bzw. Tier 1 mindestens 4, geschäftskritisch mindestens 3, Tier 2 mindestens 2. Die Tier-Punkte ergeben sich aus der Bezeichnung in vendor_tier, nicht aus der Zeile Tier-Kritikalität: eine Nummer im Namen gilt direkt (Tier 1 = 5, Tier 2 = 3, Tier 3 = 0), Bezeichnungen ohne Nummer zählen nach ihrer Position (bei vier Stufen 0, 2, 4, 5). Bisher kam ein missionskritischer Lieferant ohne Personendaten fast immer auf „niedrig“. Dieselbe Regel gilt in REST-API und Risikohistorie; Stufe und Score aller Lieferanten werden einmal je Installation neu berechnet (Merker vendor_bewertung_2026_11_tier in isms_settings).

27.2 Risikostufen-Schwellenwerte

Risikowert = Eintrittswahrscheinlichkeit × Schadenshöhe (je 1–5, gesamt 1–25). Die Schwellen gelten für alle Risikoarten.

Risikostufe	Standard-Schwelle (< X)	Bedeutung
niedrig	5	Risikowert < 5 → niedrig
mittel	10	Risikowert < 10 → mittel

© 2012–2026 WolfInfoSec GmbH

Risikostufe	Standard-Schwelle (< X)	Bedeutung
hoch	20	Risikowert < 20 → hoch
kritisch	≥ 20	Automatisch kritisch (kein eigener Schwellenwert)

27.3 Behebungsfristen für Schwachstellen

Seit v4.00 legt der Abschnitt „Fristen“ fest, wie viele Tage eine Schwachstelle je Schweregrad offen bleiben darf (ISO/IEC 27001 A.8.8, NIS-2 Art. 21 Abs. 2 lit. e). Die Tage stehen in der Tabelle swm_fristen (schwachstellen_fristen.py). Fälligkeit = Erfassungsdatum + Frist des aktuellen Schweregrads; sie wird nie gespeichert, sondern immer frisch berechnet – eine geänderte Frist oder ein geänderter Score wirkt sofort. Ein leeres Feld heißt „keine Frist“; behobene, akzeptierte und verworfene Schwachstellen sind nie fällig.

Schweregrad	CVSS	Vorgabe
Kritisch	9.0 – 10.0	7 Tage
Hoch	7.0 – 8.9	30 Tage
Mittel	4.0 – 6.9	90 Tage
Niedrig	0.1 – 3.9	180 Tage
Information	0.0	keine Frist
Ohne Score	–	30 Tage

Liste und Detailansicht der Schwachstellen zeigen die Fälligkeit, die Kachel „Überfällig“ zählt die überschrittenen; der Schwachstellen-Assistent nutzt dieselbe Berechnung.

© 2012–2026 WolfInfoSec GmbH

28. Risikoscore-Historie und Cron-Zeiten

WolfISMS zeichnet den Verlauf dreier Durchschnitts-Scores täglich auf (historie_modul.py / risiko_historie.py): ⌀ offene Risiken, ⌀ alle Risiken, ⌀ aktive Lieferanten. Dieses Kapitel beschreibt die Erfassung und die Steuerung der Hintergrund-Jobs; die Auswertung unter /risiko/historie steht in der Benutzerdokumentation.

28.1 Snapshot-Mechanik

- ensure_today(db) prüft bei jedem Aufruf von /risiko/historie, ob für den aktuellen Tag bereits ein Snapshot existiert, und legt bei Bedarf einen an
- Auf der Seite selbst lässt sich über action=snapshot (POST /risiko/historie) ein Snapshot jederzeit manuell erzwingen

28.2 Cron-Zeiten-Verwaltung (/admin/cron_zeiten)

admin_cron_zeiten.py verwaltet Uhrzeit, Rhythmus und Aktiv-Status der täglichen Hintergrund-Jobs in der Tabelle cron_settings. Aktuell eingebunden: die Aufgaben-Erinnerungs-E-Mail (aufgaben_mailer.py) und der Risikoscore-Snapshot (risiko_historie.py).

- Rhythmus wählbar: jeden Tag, alle 2/3/14/30 Tage, wöchentlich
- Über "Jetzt ausführen" lässt sich jeder Job sofort manuell starten, unabhängig von Uhrzeit, Rhythmus und Aktiv-Status

Wichtige Voraussetzung:
Damit eine unter /admin/cron_zeiten gespeicherte Uhrzeit tatsächlich greift, muss die Server-Crontab die betroffenen Skripte häufiger aufrufen (empfohlen: alle 5 Minuten), z. B.
*/5 * * * * /usr/bin/python3 /var/www/isms/aufgaben_mailer.py
Jeder Lauf prüft selbst, ob die eingestellte Zeit erreicht ist, und führt die eigentliche Arbeit höchstens einmal pro Tag (bzw. gemäß Rhythmus) aus. Dieser Zeitplan steuert nicht die Datensicherung – siehe Kapitel Datensicherung (Backup).

28.3 Aufgaben-Erinnerungen und Lesebestätigungen

Der Aufgaben-Mailer versendet die Erinnerung an fällige Aufgaben. Zu den erfassten Aufgaben zählen neben Maßnahmen (für jeden ihrer Besitzer), Reviews und Dokument-Workflows auch offene Lesebestätigungen: Schulungs- und Richtliniendokumente, die ein Benutzer laut Lesepflicht-Matrix lesen muss und im Policy Portal noch nicht als gelesen bestätigt hat (users.dok_read). Welche Dokumenttypen überhaupt eine Lesebestätigung kennen, legt die Positivliste LESE_TYPEN in portal_modul.py fest; wer davon welches lesen muss, entscheidet die Matrix dok_lesepflicht × user_rollen – siehe Kapitel Rollen und Lesepflicht. Ein Dokument ohne Rollenzuordnung und ein Benutzer ohne Rolle lösen keine Erinnerung aus.

Meldefristen: Seit v3.94 nimmt der Mailer auch die Meldefristen der Incidents auf, mit dem üblichen Vorlauf VORLAUF_TAGE = 14 und der Umleitung an Vertreter. Er importiert dazu meldungen_modul, statt die Fristberechnung nachzubauen; schlägt der Import fehl, steht eine Warnung im Log und der übrige Versand läuft weiter. Die Mail geht einmal täglich hinaus – eine 24-Stunden-Frist kann vor der ersten Erinnerung ablaufen; maßgeblich bleibt die Seite.

Seit v3.48 wird zusätzlich unmittelbar bei der Freigabe eines Dokuments benachrichtigt, statt erst mit dem nächsten Lauf des Aufgaben-Mailers (dokumente_modul._notify_lesepflicht_freigabe):

© 2012–2026 WolfInfoSec GmbH

- **Auslöser**: die Freigabe eines Dokuments, dessen Typ in LESE_TYPEN steht. Alle anderen Typen werden im Portal nur angezeigt und lösen keine Mail aus
- **Empfänger**: die Benutzer der Rollen, die das Dokument laut Lesepflicht-Matrix lesen müssen – sofern aktiv, mit E-Mail-Adresse und ohne abbestellten Mailversand (users.aktiv, users.no_mail). Fehlen diese Spalten in einer Altinstallation, wird der jeweilige Filter übersprungen. Ist keine Rolle zugeordnet, geht keine Mail hinaus; das Dokumentenmodul weist den Freigebenden darauf hin
- **Sprache**: je Empfänger nach users.lang; die Mail enthält Nummer, Name, Version, Freigebenden und einen Direktlink in das Policy Portal
- **Versand**: alle Mails gehen über eine einzige SMTP-Verbindung (aufgaben_mailer.send_email_bulk). Ein Fehler bei einem Empfänger bricht den Versand an die übrigen nicht ab
- **Fehlertoleranz**: fehlender oder abgeschalteter SMTP-Zugang wird protokolliert, aber nie an den Request weitergereicht – die Freigabe selbst funktioniert auch ohne Mailversand

Seit v3.90 wird jede Bestätigung zusätzlich mit Zeitstempel in dok_lesebestaetigung protokolliert – siehe Kapitel Berichte und Meldewesen.

© 2012–2026 WolfInfoSec GmbH

29. Diagnose- und Wartungsskripte

Neben den Web-Blueprints liegen im Installationsverzeichnis einige eigenständige Kommandozeilen-Skripte für Diagnose und einmalige Migrationen. Sie werden manuell mit sudo python3 <script>.py im Anwendungsverzeichnis ausgeführt und sind nicht über den Webserver erreichbar.

- **check_config.py**: zeigt alle von config.py aufgelösten Werte (inkl. verwendeter Env-Datei) an, um sie vor einer Umstellung mit der alten Konfiguration abzugleichen; dbpass wird nur maskiert ausgegeben
- **debug_mailer.py**: prüft, ob secret.env gefunden und gelesen werden kann, und vergleicht die daraus geladenen DB-Zugangsdaten mit den gesetzten Umgebungsvariablen – hilfreich bei Startproblemen des Aufgaben-Mailers
- **diag_lese.py <email>**: zeigt für einen einzelnen Benutzer exakt den Lesestatus-Wert (dok_read) sowie die zugehörigen Schulungsdokumente, wie ihn auch aufgaben_mailer.py berechnet
- **migration_nummern.py**: vereinheitlicht bestehende Objektnummern (Dokumente, Risiken, Maßnahmen, Assets, Lieferanten, Projekte, Incidents, BIA) chronologisch auf das Format PREFIX-NNNNN; Default ist ein Dry-Run, erst --commit schreibt in die Datenbank
- **read_version.py [datei]**: liest die Version-Variable aus isms.py (oder einer angegebenen Datei) aus, z. B. für Skripte oder CI/CD-Pipelines. Fehlen die .py-Quellen, weil sie nach dem Kompilieren entfernt wurden, wird die Version aus der daneben liegenden .pyc gelesen, ohne das Modul auszuführen

Hinweis: Diese Skripte sind Wartungswerkzeuge für Administratoren mit Server-Zugriff, nicht Teil der Web-Oberfläche. migration_nummern.py verändert bei --commit historische Objektnummern dauerhaft – nur nach vorherigem Backup und in Abstimmung mit den Anwendern ausführen.

© 2012–2026 WolfInfoSec GmbH

30. Aktivitäten und Protokollierung

WolfISMS protokolliert alle sicherheitsrelevanten Benutzeraktionen in der Datenbanktabelle activities sowie Login-Ereignisse in einer separaten Logdatei.

30.1 Aktivitätentabelle (Datenbank)

Spalte	Inhalt
id	Primärschlüssel
datumuhrzeit	Zeitstempel der Aktion
user	Benutzer-ID
nummer	Datensatznummer / ID des betroffenen Objekts
bereich	Modulbereich (z. B. Risiken, Maßnahmen, Assets …)
was	Beschreibung der Aktion (erstellt, geändert, gelöscht)

30.2 Weitere Protokolle

Protokoll	Ort	Inhalt
Login-Protokoll	/var/www/isms/login.log	Erfolgreiche und fehlgeschlagene Anmeldungen im Format YYYY-MM-DD HH:MM:SS – [Nachricht]

Auswertung des Login-Protokolls empfohlen mit: tail -f /var/www/isms/login.log

Entfallen sind mit v4.00 die Protokolle von SQL- und REST-Konsole und des Software-Updates. Die Tabellen sql_console_log und rest_console_log bleiben in bestehenden Datenbanken samt Inhalt liegen, werden aber von keiner Route mehr geschrieben oder gelesen – wer sie nicht mehr braucht, löscht sie bewusst per DROP TABLE. Eine vorhandene update_log.txt stammt vom früheren Software-Update.

© 2012–2026 WolfInfoSec GmbH

31. Sicherheitshinweise

31.1 Zugriffskontrolle

- Der Admin-Bereich ist durch das Flag admin=1 in der Datenbank geschützt
- Alle Routen nutzen @login_required und prüfen zusätzlich die jeweils erforderliche Berechtigung
- **Controls löschen:** seit v3.95 nur Administratoren – die Löschroute in normen.py weist alle anderen ab, auch bei direkt abgesetzten Anfragen; der ¨ -Knopf in Kapitelliste und Detailansicht erscheint nur für sie. Ein ganzes Framework war schon vorher Admin-Sache
- CSRF-Schutz ist für alle POST-Formulare aktiv (Flask-WTF CSRFProtect); die fünf REST-API-Blueprints sind explizit CSRF-exempt und authentifizieren stattdessen per API-Key. Das öffentliche Fragebogenformular behält den CSRF-Schutz und wird zusätzlich über seinen Einmal-Token geschützt
- **Schreibende Links nur per POST:** Seit v3.93 stehen alle Routen, die über einen gewöhnlichen Link Daten ändern, in _NUR_POST (isms.py) – Verknüpfen und Lösen in sieben Modulen, die Lieferantenaktionen, das Löschen von Kontakten, die Lesebestätigung im Policy Portal und der Behandlungsplan. Ein GET darauf wird abgewiesen. Die Links in den Modulen bleiben, ein globales Skript in template.html schickt sie als POST mit CSRF-Token, auch nachträglich eingefügte. Erkannt werden sie am Pfad: vor die Route kommt seit v3.95 nur der Pfadteil von WEBPATH – in Docker-Installationen ist WEBPATH eine vollständige URL, bis dahin griff dort kein Eintrag und jeder Link wurde als GET abgewiesen. Folgen: in einem neuen Tab geöffnet, läuft eine solche Aktion nicht; bei abgelaufener Lizenz sind sie für normale Benutzer ebenso gesperrt wie Formulare. Neue Module tragen schreibende Links dort ein
- Rate Limiting: global 2000 Req/Tag und 500 Req/Stunde, bei Überschreitung HTTP 429. Die REST-API ist davon ausgenommen (limiter.exempt), damit Sync-Clients nicht ausgebremst werden – dort schützt allein der API-Key
- Passwörter werden mit Werkzeug gehasht – niemals im Klartext gespeichert
- API-Keys werden nur als SHA-256-Hash gespeichert – Klartext nach der Erstellung nicht wiederherstellbar
- Deaktivierte Benutzerkonten (aktiv=0) können sich weder per Passwort noch per SSO anmelden

31.2 Uploads und Anhänge

- **Request-Grenze**: MAX_CONTENT_LENGTH liegt bei 512 MB. Der Wert ist bewusst hoch, weil die Backup-Wiederherstellung ein komplettes Archiv aus SQL-Dump und Upload-Verzeichnis entgegennimmt – er ist deshalb kein Schutz für die Anhang-Routen
- **Anhang-Grenze**: für einzelne Anhänge gilt die deutlich engere Grenze von 16 MB, die validate_pdf() durchsetzt
- **Dateityp-Prüfung**: Anhänge müssen echte PDF-Dateien sein: geprüft werden Dateiendung, Magic Bytes (%PDF) und der Dateiname
- **Content-Type beim Download**: der vom Client gemeldete Content-Type wird weder gespeichert noch beim Download verwendet. Anhänge werden immer mit application/pdf ausgeliefert, damit eine als PDF getarnte HTML-Datei nicht im Origin der Anwendung ausgeführt werden kann
- **Word-Import**: Die .docx-Datei wird nicht gespeichert, sondern im Arbeitsspeicher umgewandelt und wieder verworfen. Geprüft werden Endung, ZIP-Signatur und eine eigene Größengrenze von 16 MB; das erzeugte HTML durchläuft denselben Sanitizer wie jede Eingabe im Editor

31.3 Eingabe- und Ausgabesicherheit

- Richtext-Felder werden serverseitig mit bleach bereinigt

© 2012–2026 WolfInfoSec GmbH

- **Daten in Skriptblöcken:** Werte, die in einem <script>-Block landen, gehen durch js_daten() (isms_helpers.py), das <, > und & maskiert – json.dumps allein lässt ein </script> durch. URL-Parameter, die eine ID meinen (etwa /bpmn?id=, /risiko/mass_neu?riskdetail=), werden nur als Zahl angenommen
- Freitext, der in HTML-Attribute oder Textbereiche gelangt, wird konsequent escaped – das betrifft insbesondere Parameter der Dokument-Workflow-Routen (/pruefen, /zupruefen, /freigabe), das Änderungsprotokoll und die Benutzernamen in Auswahllisten. Benutzernamen gibt der Helfer user_label() überall einheitlich als „Nachname, Vorname" aus – in derselben Reihenfolge, nach der die Auswahllisten sortiert sind
- Gespeicherte Links werden über den Helfer safe_url() (isms_helpers.py) mit einer Schema-Allowlist geprüft, damit javascript:-URLs nicht ausgeführt werden können; url_param() maskiert Werte, die in URL-Parameter gelangen
- Das Login meldet unbekannte Benutzer und falsche Passwörter mit demselben Text, damit sich gültige Konten nicht durchprobieren lassen; die Unterscheidung bleibt im Aktivitäten-Log erhalten

31.4 Eingabeprüfung vor der Datenbank

Ein Formularwert, der nicht in seine Spalte passt, fiel bis 3.51 erst beim INSERT auf: MySQL meldete „Data too long for column" oder „Incorrect date value", der Anwender sah eine Fehlerseite und seine Eingaben waren verloren. Inzwischen prüft jeder Schreibpfad die Werte vorher gegen das Schema – zentral in isms_helpers.py (Klasse Pruefung, Funktion spalten_laenge).

- Die Grenzen kommen aus dem information_schema, nicht aus im Code gepflegten Konstanten: eine geänderte Spalte wirkt damit ohne Codeänderung
- VARCHAR wird in Zeichen gezählt, TEXT in Bytes – ein Umlaut belegt in utf8mb4 zwei Bytes, ein Emoji vier
- Die Meldung nennt die Beschriftung aus der Maske, nicht den Spaltennamen, und sammelt alle Beanstandungen eines Formulars statt beim ersten Fehler abzubrechen
- Die Formularattribute maxlength und type="date" bleiben bestehen, sind aber nur ein Vorschlag an den Browser – ein direkt abgesetzter POST umgeht sie
- Dateinamen von Anhängen werden auf die Spaltenbreite gekürzt (die Endung bleibt erhalten), damit ein INSERT nicht scheitert, nachdem die Datei bereits auf der Platte liegt
- Kommentare sind auf 4000 Zeichen begrenzt; zu lange Texte werden mit Hinweis abgewiesen statt still abgeschnitten

31.5 Konfigurationssicherheit

- Datenbankpasswort liegt nicht in config.py, sondern in /etc/wolfisms/secret.env
- secret.env: chmod 640 und chown root:www-data – kein Webzugriff möglich
- Flask Secret Key: ISMS_SECRET_KEY aus secret.env oder, wenn leer, die beim ersten Start erzeugte Datei .flask_secret_key im Session-Verzeichnis (0600). Er signiert Sitzungen und CSRF-Token und schützt seit v3.99 den KI-API-Key – ein Wechsel meldet alle Benutzer ab und macht den gespeicherten KI-Key unlesbar
- **ISMS_PROXY_HOPS:** Anzahl der Reverse-Proxys vor der Anwendung (Vorgabe 1, im Docker-Image gesetzt). Nur so viele Stationen der Header X-Forwarded-For/-Proto/-Host/-Port werden ausgewertet – Client-IP für das Rate-Limit, HTTPS-Erkennung, Hostname für Weiterleitungen und externe Links. Hängt die Anwendung direkt am Webserver (Apache mit mod_wsgi), gehört ISMS_PROXY_HOPS=0 in secret.env, sonst kann jeder Client die Header selbst setzen und seine IP oder den Hostnamen fälschen. Bis v3.98 waren Proto und Host fest eingeschaltet und For gar nicht: hinter einem Proxy sah das Rate-Limit nur dessen IP, und alle Benutzer teilten sich ein Login-Kontingent

© 2012–2026 WolfInfoSec GmbH

- config.py enthält keinen Fallback-Wert für DB_PASSWORD – fehlt die Variable, startet die App nicht
- secret.env niemals in Git-Repositories einchecken (.gitignore eintragen)
- SSO-Zugangsdaten (MS_*, GOOGLE_*) ebenfalls in secret.env – nicht in isms.conf
- Der Flask-Debug-Modus ist standardmäßig aus und darf im Produktivbetrieb nicht aktiviert bleiben

31.6 Authentifizierung und Sessions

- Sessions werden serverseitig im Dateisystem gespeichert (Flask-Session, Typ: filesystem). Bis 3.51 war das zwar konfiguriert, aber nie eingeschaltet – der Aufruf Session(app) fehlte, sodass Flask weiter das signierte Client-Cookie verwendete. Der Sitzungsinhalt war damit im Browser lesbar und liess sich serverseitig nicht entziehen. Seit 3.52 ist der Aufruf da; beim Start warnt die Anwendung im Log, falls doch wieder die Cookie-Variante aktiv sein sollte
- Session-Cookies: Secure=True, HttpOnly=True, SameSite=Lax, Lifetime = 1 Tag (SESSION_PERMANENT=True)
- **Angemeldet bleiben:** Das Remember-Cookie trägt seit v3.93 ebenfalls Secure, HttpOnly und SameSite=Lax. Wer vorher mit „angemeldet bleiben“ eingeloggt war, behält das alte Cookie bis zum Ablauf oder zur nächsten Anmeldung
- Max. 500 gleichzeitige Sessions (SESSION_FILE_THRESHOLD)
- Das Session-Verzeichnis wird beim Start auf 0700 gesetzt, die Sitzungsdateien auf 0600 – lesen darf sie nur der Benutzer, unter dem die Anwendung läuft (SESSION_DIR, Vorgabe /var/www/isms/wolfisms/sessions)
- Bei Kompromittierung: alle Session-Dateien löschen (rm -rf /var/www/isms/wolfisms/sessions/*)
- Rate-Limit-Speicher: die Vorgabe memory:// zählt je WSGI-Prozess – bei mehreren Workern vervielfacht sich damit jedes Limit. Für solche Installationen RATELIMIT_STORAGE_URI auf einen gemeinsamen Speicher setzen (z. B. redis://…). Im Container warnt die Anwendung beim Start, solange memory:// gilt – das Image startet drei gunicorn-Worker
- **Kontosperre:** Seit v3.99 sperrt WolfISMS ein Konto nach zehn falschen Passwörtern für 15 Minuten – gezählt in der Datenbank, unabhängig vom IP-Limit (Abschnitt Kontosperre nach Fehlversuchen). Bei unbekannter E-Mail-Adresse wird gegen einen Dummy-Hash geprüft, damit die Antwortzeit nicht verrät, ob ein Konto existiert
- Läuft ein Limit auf, antwortet WolfISMS passend zum Aufrufer: JSON-Clients erhalten die Fehlerform der API samt Retry-After, Browser eine Meldung, die zum überschrittenen Limit passt – nicht mehr pauschal „zu viele Anmeldeversuche"

31.7 Auslieferung: Bilder, Kompression und Header

- **Bilder als eigene Adressen**: Logo, Flaggen und die übrigen Bilder kommen seit v3.93 über /bild/<datei>?v=<Änderungszeit> und /branding-bild/… statt als base64 in jeder Seite. Der Browser hält sie ein Jahr; ändert sich eine Datei, ändert sich die Adresse. Beide Routen gehen ohne Anmeldung (die Login-Seite zeigt das Logo), sind vom Rate-Limit ausgenommen und liefern nur Bildformate aus – ../-Pfade und andere Dateien enden mit 404
- **Linien-Icons:** Menü- und Knopfsymbole sind seit v4.00 keine Bilddateien mehr, sondern SVG-Masken im Stylesheet (ui_icons.py). Sie färben sich mit der Schrift, auch im Dunkelmodus, und brauchen keinen eigenen Abruf
- **Kompression**: HTML und JSON gehen gzip-komprimiert hinaus – wichtig im Docker-Betrieb, wo gunicorn ohne vorgeschalteten Server ausliefert. Abschalten mit der Umgebungsvariable ISMS_GZIP=0, etwa wenn ein Reverse Proxy bereits komprimiert
- **Header**: Cache-Control: no-store für HTML und JSON als echter HTTP-Header, Strict-Transport-Security (nur bei HTTPS) und Cross-Origin-Opener-Policy: same-origin

© 2012–2026 WolfInfoSec GmbH

- **Templates**: korrekte Meta-Angaben, <html lang> nach der Sprache des Benutzers, alt-Texte an allen Bildern

31.8 Empfohlene Härtungsmaßnahmen

- Gültige Lizenzdatei einspielen – verhindert den Demo-Modus und hebt die Mengengrenzen auf
- HTTPS erzwingen (SSL/TLS im Apache-Vhost konfigurieren) – Session-Cookies sind bereits Secure=True
- Regelmäßige Backups einplanen (mindestens täglich per Cronjob) und den Restore jährlich testen
- Login-Log regelmäßig auf Fehlversuche und HTTP-429-Einträge überwachen
- Mindestens einen zweiten Admin-Account anlegen (Notfallzugang)
- Master-Rechte nur sehr sparsam vergeben – sie erlauben Freigaben, die Sicht auf alle Aufgaben und die Rollenpflege
- Nicht benötigte Benutzerkonten deaktivieren und API-Keys löschen
- Pro externer Anwendung einen eigenen API-Key verwenden
- Den Datenbankbenutzer nach dem Prinzip der geringsten Rechte ausstatten

© 2012–2026 WolfInfoSec GmbH

32. Administrator-Checkliste

32.1 Regelmäßige Wartung

Intervall	Prio	Aufgabe
Täglich	H	Backup erstellen und extern sichern
Täglich	H	Login-Log auf Fehlversuche prüfen
Wöchentlich	M	Auf neue WolfISMS-Versionen prüfen und einspielen (Docker: neues Image); JavaScript-Bibliotheken unter /admin/updates
Wöchentlich	M	Benutzerkonten und Berechtigungen prüfen; nicht mehr benötigte Konten deaktivieren
Monatlich	N	Aktivitätenprotokoll auswerten
Monatlich	N	Lizenzablauf prüfen
Quartalsweise	M	Rezertifizierung der Zugriffsmatrix – fällige Berechtigungen prüfen, insbesondere privilegierte (Stufen A und N)
Jährlich	H	Backup-Restore testen
Jährlich	H	Managementbewertung nach ISO 27001 Kap. 9.3 durchführen und dokumentieren

© 2012–2026 WolfInfoSec GmbH

33. Support und Kontakt

WolfISMS – Wolf's Informations-Sicherheits-Management-System
© 2012–2026 WolfInfoSec GmbH
Update-Server: https://update.wolfisms.de/downloads
Version: 4.01 | Stand: November 2026

Änderungshistorie dieses Dokuments

Revision	Version	Wesentliche Änderungen
Rev. 4	v2.13	REST-API, API-Key-Verwaltung, lizenzbasierter Secret Key, Azure-AD-Env-Variablen, Session-Cookie-Flags, Rate Limiting
Rev. 5	v2.30	Kommentarfunktion, Cron-Zeiten-Verwaltung, SQL-Konsole, Risikoscore-Historie, Rollen-/Bedrohungs-/Schwachstellen-Kataloge, Branding, Normen-Katalog-Direktinstallation, Container-Modus, korrigierte Beschreibung von Backup und Restore
Rev. 6	v3.28	Ziele-Modul, Operational-Plan-Modul, Dokumente-SQL-Export/Import, korrigierte Blueprint- und Datenbankspalten-Übersicht, neue Berechtigungskürzel ZIE und LNK
Rev. 7	v3.47	Neue Kapitel Managementbewertung, Zugriffsmatrix, REST-Konsole und Single Sign-On; Daten-SQL-Export/Import ersetzt den reinen Dokumentenexport; Abteilungen und Teams als Stammdaten; Rollenpflege in den Rollenkatalog verschoben; erweiterte Incidents-API (GET-Liste mit Filtern); gehärteter Backup-Restore; Debug-Modus standardmäßig aus; überarbeitetes Sicherheitskapitel (Upload-Grenzen, Anhang-MIME-Typ, XSS- und Enumerationsschutz); aktualisierte Konfigurations-, Paket- und Berechtigungsübersichten
Rev. 8	v3.51	Modulkapitel 19–23 auf die administrativen Angaben gekürzt – die Bedienung steht in der Benutzerdokumentation; neue Kapitel KI-Assistenten für Risiken und Vorfälle sowie Design und Farbschema; KI durchgängig an die Enterprise-Lizenz (ENT) gebunden; Anbieterliste um Azure AI Foundry (Copilot) erweitert und Hinweise zu Basis-URL und API-Version ergänzt; automatische Weiterleitung zu Microsoft mit Notfallzugang; Benachrichtigung der Leseverpflichteten bei der Dokumentenfreigabe; aktualisierte Blueprint- und Admin-Menü-Übersicht
Rev. 9	v3.63	Fünf REST-API-Bereiche statt nur Incidents, mit Synchronisationsstand und ohne Rate-Limit; neues Kapitel Lieferantenfragebogen; Passwort-vergessen-Selbstbedienung; Sperre nach TOTP-Fehlversuchen und tatsächlich aktivierte serverseitige Sessions; zentrale Eingabeprüfung vor der Datenbank; Benutzernamen einheitlich als „Nachname, Vorname"
Rev. 10	v3.68	Editionen und Funktionsumfang in der Lizenzverwaltung (Zugriffsmatrix, Managementbewertung und Operational Plan erst ab Enterprise); neues Kapitel Modulverwaltung – Module unter /admin/module ein- und ausschaltbar und im Menü sortierbar; neues Fachmodul Schwachstellen mit eigener Berechtigungsspalte (SWM) und sechstem REST-API-Bereich; drei Kapitelverweise richtiggestellt
Rev. 11	v3.70	Neues Kapitel Unternehmen und Strukturkarten (Unternehmensangaben, Organigramm, Prozesslandkarte, Netzwerkzonen); Editionen neu gefasst – kein Fachmodul mehr editionsabhängig, KI-Funktionen ab Standard (STD); SSO über /admin/sso in der Oberfläche einrichtbar; Excel-Export und -Import der Abteilungen
Rev. 12	v3.71	Neues Kapitel Rollen und Lesepflicht: Rollenverwaltung unter /admin/rollen, Mehrfachzuordnung über user_rollen, Lesepflicht-Matrix des Policy-Portals (dok_lesepflicht); Aufgaben-Erinnerungen und Freigabemail folgen der Matrix

Revision	Version	Wesentliche Änderungen
Rev. 13	v3.72	Neues Kapitel Auditmanagement – das Audit-Modul ersetzt die Fragenkataloge durch Audits und Abweichungen (aud_audits, aud_abweichungen, aud_anhaenge), die alten Tabellen bleiben unangetastet; neues Kapitel On-/Offboarding (pers_onboarding); Antwortlänge der KI je Modell mit Fortsetzungsrunden; externe Domain des Fragebogen-Links
Rev. 14	v3.75	Neuer Abschnitt Mengengrenzen der freien Version in der Lizenzverwaltung: DEMO ist auf je zehn Assets, Risiken, Dokumente und Lieferanten begrenzt, ab Standard unbegrenzt; Zählweise, geprüfte Schreibpfade einschließlich REST-API und Import sowie die Anzeige des Verbrauchs unter /admin/lizenz_upload
Rev. 15	v3.76	Neuer Abschnitt Prüflisten und Gap-Assessments im Auditmanagement: ein Audit übernimmt die Anforderungen einer Norm als Prüfpunkte (aud_pruefpunkte), bewertet sie fünfstufig und rechnet daraus den Erfüllungsgrad; Gap-Assessment als neue Art eines Audits; Abweichung aus einem Prüfpunkt
Rev. 16	v3.77	Editionsmodell neu gefasst: REST-API (samt API-Keys und REST-Konsole), Single Sign-On und Zugriffsmatrix gehören zur Enterprise-Edition, alle Fachmodule bleiben in jeder Edition; Enterprise ohne Nutzerobergrenze, Standard 5 und Professional 20 schreibende Benutzer; neuer Abschnitt Nutzergrenze der Editionen mit Zählweise und den drei geprüften Schreibpfaden
Rev. 17	v3.78	Neuer Abschnitt Verbrauch und Kosten im Kapitel KI-Einstellungen: WolfISMS schreibt den Tokenverbrauch jedes KI-Aufrufs aus den Antworten der Anbieter mit (ki_verbrauch), weil der Arbeits-Key keinen Kontostand liefert; Auswertung nach Modul, Benutzer und Modell unter /admin/ki_verbrauch, gepflegte Preisliste je 1 Mio. Token, Aufbewahrung 730 Tage
Rev. 18	v3.79	Neues Kapitel Dokumentenlenkung und Versionen: jede Änderung an einem Dokument legt eine eigene Fassung an (stamm_id, vorgaenger_id, ist_aktuell, historisch), der Bestand wird einmalig über die Dokumentennummer zu Versionsreihen zusammengefasst; Anhänge, Lesepflicht und Links wandern mit der Fassung, Normen, Risiken, Prozesslandkarte, Netzwerkzonen und Dokumentverknüpfungen werden bei der Freigabe umgehängt
Rev. 19	v3.80	Neuer Abschnitt Aufträge im Hintergrund im Kapitel KI-Einstellungen: lange KI-Läufe werden als Auftrag in der Tabelle ki_jobs geführt, weil der Webserver die Anfrage nicht so lange offen hält. Neuer Abschnitt Word-Import im Kapitel Dokumentenlenkung und Versionen: Umwandlung einer .docx-Datei in den Dokumententext samt Grenzen, Rechteprüfung und Vollständigkeitskontrolle
Rev. 20	v3.84	Neue Kapitel Changemanagement (Änderungsanträge, Freigabegremium in bis zu drei Stufen, Nachbetrachtung) und Business Continuity Management – BIA, Notfallpläne und Übungen hängen jetzt an Assets, RTO/RPO/MAZ stehen am Asset, die Ebene Kritische Prozesse ist entfallen. Neue Abschnitte Abwesenheit und Vertretung, Ticketsystem-Verknüpfung, Pläne & Register sowie Zusätze je KI-Abfrage
Rev. 21	v3.85	Editionsmodell neu gefasst: sechs Stufen mit den Plus-Editionen STD+ und PRO+. Acht Fachmodule gehören jetzt zu Professional (Projekte, Operational Plan, Audit, Managementbewertung, Schwachstellen, Rechtsregister, Playbooks, BCM), Zugriffsmatrix, Single Sign-On und REST-API zu den Plus-Stufen und zu Enterprise – nicht zu Standard und nicht zu Professional. Das Feld tier nimmt dafür auch eine Aufzählung erlaubter Stufen auf (tier_ok_an), und die Edition wird zentral in modul_gate() geprüft; tier_required der REST-API ist jetzt eine Liste
Rev. 22	v3.87	Das Modul Normen heißt in der Oberfläche jetzt Frameworks: Menüpunkt, Framework-Import, Framework-Bezug in Audit und Lieferantenfragebogen sowie die Berechtigungsübersicht sind umbenannt. Tabelle Normen, Recht norm und Adresse /normen bleiben unverändert.
Rev. 23	v3.89	Mehrere Besitzer je Objekt: neues Unterkapitel zur Tabelle objekt_besitzer (Hauptbesitzer, Rechte, Aufgaben, Dokumentversionen, REST-API, SQL-Export); PATCH-Felder der REST-API, Berechtigung und Termine im

© 2012–2026 WolfInfoSec GmbH

Revision	Version	Wesentliche Änderungen
		Changemanagement sowie Aufgaben-Mailer ergänzt.
Rev. 24	v3.90	Berichte und Meldewesen: neues Kapitel zu Berichtsübersicht, PDF-Berichten (Audit, BCM-Übung, Kennzahlen, Awareness), Excel-Exporten (Schwachstellen, Changes, BCM, Kennzahlen), Meldeassistent nach NIS-2, DSGVO und CRA (inc_meldepflicht, inc_meldungen), Protokoll der Lesebestätigungen (dok_lesebestaetigung) und Audit-Daten in der Managementbewertung.
Rev. 25	v3.91	Menügruppen: die Seitenleiste ordnet die Module in sechs Gruppen (MENU_GRUPPEN), die Reihenfolge aus /admin/module gilt innerhalb der Gruppe; zugeklappte Gruppen merkt sich der Browser, die Gruppe der aktuellen Seite bleibt offen; eigene Menübilder für sechs Module. Die Administrationsseite gliedert ihre Kacheln in sechs Gruppen (ADMIN_GRUPPEN).
Rev. 26	v3.92	KI-Assistenten für Assets (KI-Risikoanalyse legt Risiken an oder verknüpft sie, KI-Schutzbedarf schlägt CIAA vor), KI-Assistent und GAP-Assistent für Gap-Assessments, Haftungshinweis in allen KI-Dialogen, Übersicht der KI-Assistenten; Kategorien der Module zentral unter /admin/kategorien (stamm_kategorie); Datenbankanbindung mit DB_TYPE, DB_PORT und TLS (isms_db.py).
Rev. 27	v3.93	Bereich Frameworks heißt Controls; SoA als eigener Menüpunkt bei installierter ISO/IEC 27001; Lieferanten-Import (/vendor/import) und Cloud-Exit-Strategie (vendor_typ.cloud, vendor.cloud_exit); Schulungen im Personalmodul (pers_schulungen); KI-Anbieter nele.ai; schreibende Links nur per POST, js_daten(), Bild-Routen, gzip (ISMS_GZIP) und Antwort-Header.
Rev. 28	v3.94	Seitenleiste der Detailansichten mit höchstens fünf Einträgen je Abschnitt (sidebar_kuerzen) und Rückverweisen aus anderen Modulen (rueck_abschnitte_html); Dokument-Verknüpfungen in beiden Richtungen; KI-Verknüpfungen für Dokumente (dokument_ki_modul.py); Meldefristen in Aufgaben und Aufgaben-Mail; Risikomatrix, Kacheln und Tachos behalten im Farbschema die Ampelfarben (FARBE_FEST).
Rev. 29	v3.95	KI-Übersetzung für Dokumente (dokument_uebersetzung_modul.py), Risiko-Assistent mit Beschreibung, Business Impact und Chancen; Risiken aus Audit-Abweichungen (aud_abweichungen.risiken), Maßnahmen an BCM-Übungen (bcm_uebungen.massnahmen); Bemerkung je Schutzziel (bem_*), Kritikalität nach dem Maximumprinzip; Aktionsleisten ohne Verknüpfungs-Schaltflächen; Controls löschen nur Admin; _NUR_POST mit WEBPATH als vollständiger URL.
Rev. 30	v3.96	Neues Lizenzmodell: Essentials (ESS) und Advanced (ADV) für bis zu 5, 10 oder 25 schreibende Benutzer, jeweils auch mit „+“ für Single Sign-On und REST-API, dazu Enterprise (ENT). Ziele, Personal, Kontakte, Berichte und Zugriffsmatrix gehören jetzt zu Advanced; Free hat denselben Modulumfang wie Essentials; KI in jeder bezahlten Edition. Abschnitt Editionen und Funktionsumfang neu gefasst, Nutzergrenze, Modulverwaltung, REST-API, REST-Konsole, SSO und KI angepasst.
Rev. 31	v3.97	Tooltips: Formularfelder und Tabellenspalten aller Module tragen ein ⓘ mit Erklärtext (deutsch und englisch), Buttons und Einträge der Seitenleiste zeigen ihren Text beim Überfahren. Neuer Abschnitt Tooltips mit Funktionsweise und Ebenen; neue Kachel in der Gruppe Erscheinungsbild & Vorlagen zum Ein- und Ausschalten.
Rev. 32	v3.99	Sicherheit: KI-API-Key AES-256-GCM-verschlüsselt statt Base64 (Schlüssel aus dem Secret-Key der Anwendung, Altbestand wird beim ersten Lesen umgestellt); Secret-Key zufällig und dauerhaft statt aus der Lizenz abgeleitet (Berichtigung); ISMS_PROXY_HOPS steuert ProxyFix, X-Forwarded-For zählt für das Rate-Limit je Client; Startwarnung bei memory://-Limiter im Container; Kontosperre nach zehn falschen Passwörtern (neuer Abschnitt 2.6); SSO-Client-Secrets ebenfalls verschlüsselt; Restore setzt die Schema-Merker zurück.
Rev. 33	v4.00	Neue Oberfläche (Linien-Icons, Listenbausteine mit Live-Suche ohne Blättern, Titelkacheln, Symbolkacheln und kompakte Seitenleiste, Rücksprung zur Administration, Version in der Fußzeile); SQL- und REST-Konsole sowie das

© 2012–2026 WolfInfoSec GmbH

Revision	Version	Wesentliche Änderungen
		Software-Update entfernt; KI-Anmeldung per Entra ID; KI-Assistenten für Risiken und Assets als Beratergespräch auf eigener Seite (ersetzen Risiko-Dialog, KI-Risikoanalyse und KI-Schutzbedarf); Design auf Petrol, Klassisch und Schwarz/Grau/Weiß mit Hell/Dunkel je Benutzer und abschaltbaren Infokacheln neu beschrieben; Lieferanten-Beschreibung; neues Kapitel Playbooks (Editor); Importpaket NIS2-Standardrisiken; Zusätze je KI-Abfrage entfernt; Ziele, Prozesslandkarte und Netzwerkzonen im Hauptmenü mit eigenem Benutzerrecht; Kontakte unter Betrieb & Resilienz; KI-Assistenten aller Fachmodule (Lieferanten, Dokumente, Maßnahmen, Schwachstellen, Changes, Incidents, Projekte, Controls, Rechtsregister, Audits, Playbooks); Playbook-Modul nach ISO/IEC 27035 neu beschrieben; neue Kapitel Projekte (ISO/IEC 27001 A.5.8) und Weitere Neuerungen im Datenmodell; Behebungsfristen für Schwachstellen; Gewicht „Früherer Sicherheitsvorfall“ berichtigt; ISO/IEC 27001 in mehreren Fassungen; Copyright WolfInfoSec GmbH.
Rev. 34	v4.01	Neues Kapitel Datenschutz (Verzeichnis von Verarbeitungstätigkeiten als Verantwortlicher und als Auftragsverarbeiter, Schwellwertanalyse mit Muss-Liste, Datenschutz-Folgenabschätzung mit Risikotabelle, AVV-Block in der Lieferantenakte, KI-Assistent für den Datenschutz); Benutzerrecht „Datenschutz“; Tier in der Lieferantenbewertung stärker gewichtet.

© 2012–2026 WolfInfoSec GmbH