WolfISMS

Wolf's Information Security Management System



Administration Documentation

Version 4.01

As of: November 2026

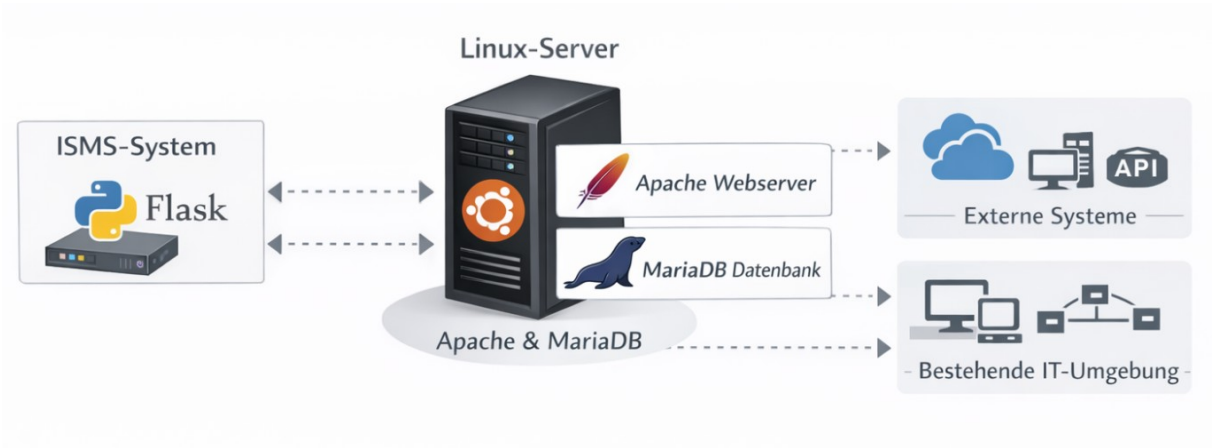
© 2012–2026 WolfInfoSec GmbH

1. System Overview

WolfISMS is a web-based information security management system (ISMS) built on the Python framework Flask. The application supports organisations in implementing and monitoring information security controls in line with common standards such as ISO/IEC 27001:2022, NIS 2, GDPR and BSI IT-Grundschutz.

1.1 Basic Technical Data

Component	Details
Framework	Python / Flask (blueprint architecture)
Web server	Apache / mod_wsgi (WSGI interface via isms.wsgi)
Database	MariaDB/MySQL, local or on a server of its own (see Database Connection) – database: "isms"
Version	4.01
Installation path	/var/www/isms/
Session directory	/var/www/isms/wolfisms/sessions/
Upload directory	/var/www/isms/wolfisms/uploads/
Request limit	512 MB per request (MAX_UPLOAD_MB) – deliberately large because restoring a backup accepts a complete archive
Attachment limit	16 MB per PDF attachment (MAX_PDF_MB, enforced by validate_pdf())
Rate limiting	2000 req/day, 500 req/hour (global); REST API exempt (limiter.exempt)
Download source	https://update.wolfisms.de/download – online framework catalogue and manuals; since v4.00 there is no software update from within the application



System architecture: Flask application on Apache and MariaDB, connected to the existing IT landscape and external systems

1.2 Database Connection

config.py reads the connection like every other value from the environment or from /etc/wolfisms/secret.env. Since v3.92 isms_db.py bundles everything that depends on the database type and the connection path: driver, port, TLS and the arguments for the command-line

© 2012–2026 WolfInfoSec GmbH

tools of the data backup. This also allows the application to be connected to a remote database server, for example a central MariaDB cluster.

Variable	Default	Meaning
DB_TYPE	mariadb	mariadb or mysql; mssql is reserved and currently rejected at start-up
DB_HOST	localhost	Host name or IP address of the database server
DB_PORT	(empty)	Empty = default port of the type (3306)
DB_NAME / DB_USER	isms / wolf	Database and user
DB_PASSWORD	–	Mandatory, deliberately without a default
DB_SSL_MODE	(empty)	empty = behaviour of driver and client unchanged; disabled = without TLS; required = TLS without certificate check; verify-full = TLS, certificate and host name checked against DB_SSL_CA
DB_SSL_CA	(empty)	CA file (PEM); mandatory for verify-full
DB_SSL_CERT / DB_SSL_KEY	(empty)	Client certificate and key, if the server requires them

- **Check at start-up**: isms_db.aktivieren() is called at the end of config.py. An unknown DB_TYPE, a type not yet supported, an invalid DB_SSL_MODE or verify-full without DB_SSL_CA stops the start with a clear message instead of incomprehensible SQL errors later on
- **Applies to every connection**: the modules still connect themselves via pymysql.connect(host=config.dbhost, …). isms_db adds port and TLS there as long as the caller does not set them itself – but only for the configured database host. New code uses isms_db.verbinden()
- **Data backup**: backup.py passes host, port, user and TLS options from isms_db.client_argumente() to mariadb-dump and mariadb; the password still goes through MYSQL_PWD and never on the command line
- **Check**: check_config.py shows the resolved values, including dbtyp, dbport and the DB_SSL settings

Further database types:
A new type is an entry in isms_db.TYPEN plus a driver branch. It only becomes usable once the SQL of the modules has been adapted – it is written in the MariaDB dialect throughout. In Docker operation DB_MODUS=extern switches off the database container; see the Docker installation guide.

© 2012–2026 WolfInfoSec GmbH

2. User Management

User management is accessible from the admin area. Only users with the flag admin=1 can reach these functions.

2.1 User Database Structure

Column name	Meaning
id	Primary key (auto increment)
email	Email address / login name
password_hash	Hashed password (Werkzeug)
vorname / nachname	First and last name of the user
admin	Administrator rights (0/1)
readonly	Portal-only access (0/1)
master	Master permission (0/1)
assets	Asset module, access matrix and change management (0 = none, 1 = read, 2 = edit)
dok	Document module (0/1/2)
mass	Measures and projects module (0/1/2)
risk	Risk module (0/1/2)
vendor	Supplier module (0/1/2)
audit	Audit module and management review (0/1/2)
norm	Controls and legal register (0/1/2)
pers	Personnel module (0/1/2)
play	Playbooks module (0/1/2)
incidents	Incidents module (0/1/2)
schwachstellen	Vulnerability module (0/1/2); the column is created by schwachstellen_modul.py itself on load
bcm	BCM module (0/1/2)
kontakte	Contacts module (0/1/2)
ziele	Objectives module (0/1/2)
prozesslandkarte	Process map (0/1/2); isms_helpers.kartenrechte_sicherstellen() creates the column itself
netzwerkzonen	Network zones (0/1/2); isms_helpers.kartenrechte_sicherstellen() creates the column itself
ki	Use the AI assistants (0/1, default 0); isms_helpers.kartenrechte_sicherstellen() creates the column itself
datenschutz	Data protection module (0/1/2, default 0); isms_helpers.kartenrechte_sicherstellen() creates the column itself
custom	Links module (0 = no access, 1 = access)
aufgabe	Tasks module
abteilung	Department name (free text, still written for compatibility)

© 2012–2026 WolfInfoSec GmbH

Column name	Meaning
abteilung_id	Foreign key to Abteilungen – authoritative for evaluations
rolle	First assigned role (foreign key Rollen). Since v3.71 only a mirror of the assignment table user_rollen – that table is authoritative
persnr	Personnel number
lang	Interface language (de/en)
dok_read	Comma-separated IDs of the documents confirmed as read (read confirmation)
totp_secret	TOTP secret (Base32, NULL if 2FA is not configured)
totp_active	2FA active for this user (0/1)
aktiv	User account active (1) or blocked (0); blocked users cannot sign in
abwesend_von / abwesend_bis / vertreter_id	Absence with deputy; see the section Absence and Deputy

Migrations when updating an older installation:
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;
The column abteilung_id is added automatically by isms_helpers.ensure_abteilung_id_spalte() and populated from the previous free-text value.

2.2 Permissions Overview

Code	Permission	Description
ADM	Administrator	Full access including user management and system settings
RO	Portal only	Read-only access through the employee portal, no admin area
MST	Master	Extended rights across modules: system-wide task and operational plan views, role maintenance, reading obligation matrix, read status reset
KI	AI usage	Use the AI assistants; administrators always may
ASS	Assets	Asset management – includes the access matrix and change management
DOK	Documents	Document management
MAS	Measures	Measures and projects module
RSK	Risks	Risk management
VEN	Suppliers	Supplier module
AUD	Audit	Audit module – includes the management review
NRM	Controls & legal register	Controls and legal requirements
PER	Personnel	Personnel module
PLY	Playbooks	Playbooks / emergency plans

© 2012–2026 WolfInfoSec GmbH

Code	Permission	Description
INC	Incidents	Incident management
SWM	Vulnerabilities	Vulnerability module (CVSS score, sources, remediation)
KON	Contacts	Contacts module
BCM	BCM	BCM module (BIA per asset, emergency plans, exercises)
ZIE	Objectives	Objectives module (corporate objectives, security objectives, KPIs)
LNK	Links	Links module (custom.py) for external system links
PRK	Process map	Process map (management, core and support processes)
NWZ	Network zones	Network zones map (network segmentation by trust level)
DSC	Data protection	Data protection module (record of processing activities, threshold analysis, DPIA, DPA block of the suppliers)

Three modules without their own permission column:
The access matrix and change management share the assets permission, the management review shares the audit permission. No change to the user model is required for either – whoever maintains assets also maintains their target permissions, and whoever sees internal audits also sees the management review.

Objectives: Since v4.00 the module is a menu entry in the group Governance & Compliance and depends solely on the permission ziele – administrators and master users need it as well. It used to sit in the admin area and was open to them without this permission. On the first request after the update, ziele_modul.ziele_recht_sicherstellen() grants them and the oldest user level 2 once. The marker ziele_recht_uebernommen in isms_settings prevents a permission withdrawn later from coming back.

Process map and network zones: Since v4.00 menu entries in the group Organisation & Inventory, each with a permission of its own (prozesslandkarte, netzwerkzonen) that administrators and master users need as well. isms_helpers.kartenrechte_sicherstellen() creates the columns itself and grants level 2 once to those who had access so far (marker kartenrechte_uebernommen). If a column is missing, for instance after an older backup has been restored, isms._benutzer_abfragen() adds it when the user is loaded; if the database user may not run ALTER, both permissions count as not granted and user management fails until the columns have been added by hand.

AI usage: Since v4.00 a yes/no permission (ki) among the system permissions, off by default – also for existing users. Without it all AI assistants are blocked; administrators do not need it. Details in the section User Permission “AI Usage”.

Data protection: Since v4.01 a module permission (datenschutz) with no access / read / edit, default no access – also for existing users, without take-over. The first user of a new installation gets edit. Details in the chapter Data Protection.

2.3 Admin Menu Functions

Menu item	URL	Function
All users	/admin/all_users	User overview with permission badges; block or

© 2012–2026 WolfInfoSec GmbH

Menu item	URL	Function
		unblock accounts
Create user	/admin/create_user	Create a new user with permissions
Edit user	/admin/change_user	Change user data and permissions
Import users	/admin/import_users	Import user master data from a file
Roles	/admin/rollen	Create roles and assign them to users; master users too
Activities	/admin/activities	Change log of all user actions
Departments & teams	/admin/abteilungen	Maintain departments and teams with a head and parent/child structure
Company	/admin/unternehmen	Master data of your own organisation (context, size category); master users too
Plans & Registers	/plaene	Repository for network diagrams, process diagrams and the asset register
Impact table	/admin/auswirkung	Edit the risk scale (5 severity levels)
Threats	/admin/bedrohungen	Maintain the threat catalogue
Vulnerabilities	/admin/schwachstellen	Maintain the vulnerability catalogue
Asset categories	/admin/asset_kategorien	Maintain the OBASHI categories of asset management
Risk categories	/admin/risiko_kategorien	Maintain the category list of the risk module
Categories	/admin/kategorien	Categories of incidents, change management, legal register and contacts, bands of process map and network zones; tabs to the other catalogues
Supplier types	/admin/vendor_typen	Maintain the type list of the supplier module; tick cloud/SaaS (vendor_typ.cloud)
Risk parameters	/admin/risiko_parameter	Vendor weights and risk level thresholds
Master data	/admin/stammdaten	Maintain further drop-down values (status, types)
Upload logo	/admin/logo_upload	Replace the company logo (PNG)
Word template	/admin/vorlage_upload	Upload the DOCX export template
PDF letterhead	/admin/dok_briefpapier	Upload the header/footer template (PDF) for PDF exports
Licence	/admin/lizenz_upload	Upload and activate the licence file (*.lic)
API keys	/admin/api_keys	Create, enable, disable and delete REST API keys
Ticket system	/admin/ticketsystem	URL template that links the incident number to the ticket
		system
Administrator manual	/admin/handbuch	Download this documentation as a PDF
About	/about	Version, licence and licence terms; since v4.00 a tile instead of the logo at the top right
Updates	/admin/updates	Check and update the bundled JavaScript libraries (TinyMCE, bpmn-js); since v4.00 without a WolfISMS software update (disabled in container mode)
Backup & restore	/admin/backup	Create a backup or restore one

© 2012–2026 WolfInfoSec GmbH

Menu item	URL	Function
Email / SMTP	/admin/smtp	Configure the outgoing mail server and send a test mail
Single sign-on	/admin/sso	Set up Microsoft and Google sign-in, domain allowlist, discovery test
Framework import	/admin/normen_import	Import framework chapters from CSV/JSON; direct installation from the online framework catalogue
Supplier questionnaire	/admin/ vendor_fragebogen	Maintain the question catalogue based on ISO/IEC 27001 and NIS2
Data SQL	/admin/daten_sql	Export/import the master data of documents, risks and measures – individually or as a combined package
Modules	/admin/module	Switch modules on and off, order within the menu groups
Cron times	/admin/cron_zeiten	Set time and interval of the background jobs
2FA / TOTP	/admin/totp	Switch system-wide 2FA (totp_required) on or off
AI settings	/admin/ki_einstellungen	AI provider, model, API key or Entra ID (Azure), prompt addendum; locked in the free edition
Design & colours	/admin/design	Colour scheme (teal, classic, black / grey / white), dark mode as default, header bar colour, info tiles on/off

Roles in two places:
Roles are maintained at /admin/rollen and in the role catalogue of the access matrix (/zugriffsmatrix/rollen) – it is the same list, see the chapter Roles and Reading Obligation. Creating, renaming and deleting remain reserved for administrators and master users.

Layout of the Administration Page

Since v3.91 the overview /admin no longer shows its tiles as one long row but in six groups with headings. All tiles of a group have the same width.

Group	Tiles
Users & Access	All Users, Create User, Roles, 2FA / TOTP, SSO, Activities
Organisation & ISMS Context	Company, Departments & Teams, Plans & Registers
Risk & Catalogues	Risk Parameters, Impact, Categories, Threats, Vulnerabilities, Supplier Questionnaire, Framework Import
Appearance & Templates	Design & Colours, Tooltips, Logo, Word Template, PDF Letterhead
Integrations & AI	Email / SMTP, Ticket System, API Keys, AI Settings, AI Usage
System & Operations	Modules, License, Backup & Restore, Data SQL, Cron Times, Updates, Admin Manual, About

- **Visibility**: the permission checks are unchanged. A group without a visible tile is left out together with its heading – updates not in container mode, the setup wizard only once its tile is enabled (then under Organisation)
- **Master users**: the master console of a pure master user uses the same groups and shows only their tiles: roles, the organisation pages, risk parameters and AI usage

© 2012–2026 WolfInfoSec GmbH

- **New tiles**: admin() collects the tiles under a key and outputs them via _kacheln_gruppiert() according to admin_modul.ADMIN_GRUPPEN. A tile without an entry there appears in an additional group “Further” – it is not lost, but should be added
- **Way back:** Since v4.00 every admin page shows “‹ Administration” at the top (context processor inject_admin_zurueck in isms.py, CSS .adm-zurueck) – for administrators and master users under /admin/* and on the modules without a sidebar entry (organisation chart, plans). Own back buttons to /admin have been removed; on the left, admin pages show the normal main menu

2.4 Departments and Teams

The former free-text "department" field has been replaced by the master-data table Abteilungen. A row without parent_id is a department, a row with parent_id is a team or sub-department. Every department can have a head (a user).

- **Maintenance**: at /admin/abteilungen – create, rename, assign a head, choose a parent department
- **Graphical view**: the button "Graphical view" opens the org chart (/organigramm) – see the chapter Company Data and Structure Maps
- **Excel export**: /admin/abteilungen/export writes a workbook with ID, name, parent department, head including email, description, sort order and colour; the last two columns (employees, usage) are information only
- **Excel import**: /admin/abteilungen/import reads the same workbook back in. Columns are matched by their heading, not by position; a row finds its counterpart by ID, otherwise by name. The parent department is resolved by name – including against departments the same import creates first – and the head by email address, alternatively by "surname, first name". The import never deletes, never guesses and only writes after the preview with the checkbox ticked. Cycles are checked against the tree as it would look after the import
- **Used in**: users, Assets, Massnahmen, Risiko and kontakte_mitarbeiter
- **Backward compatibility**: every module writes both the new column abteilung_id and the previous text column, so existing reports and filters keep working unchanged
- **Automatic migration**: ensure_abteilungen_tabelle() creates the table on first access and imports the existing free-text names; ensure_abteilung_id_spalte() adds the ID column per table and fills it by matching names

2.5 Forgotten password (self-service)

The login page carries a “Forgot password” link. Users enter their email address and receive a link that lets them set a new password themselves – no administrator involvement needed. It reuses the token mechanism of the welcome email (table pw_reset_tokens, route /set_password), only with a much shorter validity.

Property	Value
Route	/forgot_password
Link validity	2 hours (welcome email: 72 hours)
Minimum interval per account	120 seconds between two emails
Rate limit	10 POSTs per hour and IP address
Prerequisite	Working SMTP dispatch (see /admin/smtp)
Logging	Every attempt is written to the activity table, including the unsuccessful

© 2012–2026 WolfInfoSec GmbH

Property	Value
	ones

The link on the login page appears only where the email/password form is shown as well. With SSO-only sign-in the identity provider owns the password; a reset link would set up a local password alongside it and thereby bypass the provider's policies (MFA, conditional access).

The page always answers the same way.
Whether the address is known, whether the account is disabled or an SSO account – the user always sees the same confirmation. Otherwise this page could be used to probe which addresses have an account. What actually happened is recorded in the activity log. If no mail dispatch is configured, the page says so openly and points to the administrator instead of feigning a dispatch.

2.6 Account lockout after failed attempts

Since v3.99 WolfISMS counts wrong passwords per account and temporarily locks sign-in – modelled on the 2FA code entry (section Lockout after failed attempts in the chapter Two-Factor Authentication). The IP limit of the login page alone is not enough: it does not help against distributed attempts and only counts per client behind a reverse proxy if ISMS_PROXY_HOPS is set correctly. The count lives in the database and thus applies to all worker processes together.

Property	Value
Failed attempts before lockout	10 (LOGIN_MAX_FEHLVERSUCHE in isms.py)
Lockout duration	15 minutes (LOGIN_SPERRE_MINUTEN); the counter drops to 0 when locking, afterwards the account has the full allowance again
Columns	users.login_fehlversuche and users.login_gesperrt_bis – added by the application itself at the first login after the update
During the lockout	The password is not checked, not even a correct one; the message states the remaining time in minutes
Clearing	Automatically when the time is up. Earlier directly in the database: UPDATE users SET login_fehlversuche=0, login_gesperrt_bis=NULL WHERE id=<USER_ID>;
Not affected	Sign-in via single sign-on and the 2FA code entry (own lockout)
Log	Activity table: "Fehlanmeldung (falsches Passwort, Konto gesperrt)" when the limit is reached, "Fehlanmeldung (Konto gesperrt)" for every attempt during the lockout

Two deliberate decisions: the lockout message only appears for an existing account and thus reveals after ten attempts that it exists – but a user must learn why their correct password is not accepted. And all helpers fail open: a database error while counting locks nobody out but is written to the log.

2.7 Absence and Deputy

Every user can record an absence with a deputy for a period – on their account page via "Manage absence" (/abwesenheit). During that period the deputy sees the absent user's tasks under "My

© 2012–2026 WolfInfoSec GmbH

Tasks” and receives their reminder emails. Owners, reviewers and responsibles in the records stay unchanged; once the period is over everything falls back automatically.

Administrators maintain another user’s absence on the same page (/abwesenheit?user=<id>), reachable from the user list and the user form. There is deliberately no second form: the validation rules live in a single place (isms_helpers.vertretung_pruefen), which the REST API uses as well (PUT /api/v1/users/<id>/vertretung).

Column in users	Meaning
abwesend_von / abwesend_bis	Period of the absence (DATE); both limits are included
vertreter_id	User ID of the deputy

isms_helpers.ensure_vertretung_spalten() creates the three columns on first access; no migration script is needed.

- **No period without a deputy**: only two states are allowed – all three fields empty or a complete entry. A period without a deputy is rejected, because the tasks would otherwise be left lying unnoticed.
- **Who can deputise**: active users without the portal-only flag (readonly), not the absent user themselves.
- **Chains**: if the deputy is absent as well, the tasks move on to their deputy (vertretung_empfaenger). A loop is detected and stopped.
- **Reminder emails**: aufgaben_mailer.py moves the absent user’s due tasks to the deputy. Outstanding read confirmations stay with the absent user – nobody can give them on someone else’s behalf. If the deputy receives no emails (no_mail), the tasks stay with the absent user and the log records it.
- **Log**: every change is written to the activities table – with the user edited and the user who made the change.

2.8 Ticket System Link (/admin/ticketsystem)

If a security incident is also worked on in a ticket system, the number in the incident list can link straight to it. A URL template is stored, for example https://ticket.example.com/browse/<ticketnr>; the placeholder is replaced by the number of the incident (admin_ticketsystem.py).

- **Storage**: isms_settings, key ticket_url_template, at most 255 characters.
- **Placeholder**: <ticketnr> is the documented spelling; {ticketnr} and other capitalisation are recognised as well. Without it the number is appended to the template.
- **Only http and https**: a template with another scheme is not linked – a stored javascript: address never ends up as a link in the list.
- **Without a template**: the number leads to the incident detail view as before.

2.9 Multiple Owners (objekt_besitzer)

Since v3.89, risks, assets, documents, measures, projects, controls, suppliers, incidents, vulnerabilities and changes can have several owners. Instead of the drop-down, the form shows a multi-select with a search field; the selected users are listed first. The owner column of the business table (Besitzer, Owner for incidents, besitzer for suppliers) remains the main owner –

© 2012–2026 WolfInfoSec GmbH

sorting, REST API, imports and the review default keep working with it unchanged. All further owners are stored in a shared table.

Column	Type	Meaning
tabelle	VARCHAR(64)	Name of the business table: Risiko, Assets, Dokumente, Massnahmen, Projekte, Normen, vendor, Incidents, swm_schwachstellen, chm_changes
objekt_id	INT UNSIGNED	ID of the record in that table
user_id	INT UNSIGNED	additional owner (users.id)

The primary key is (tabelle, objekt_id, user_id), plus an index on user_id. The table creates itself on first access; there is no migration script.

- **Main owner:** Stays the main owner as long as he or she is selected. If deselected, the first selected user takes over. Without a selection the user saving becomes owner – for controls an owner that was empty stays empty when editing.
- **Permissions:** besitzer_check() treats the main owner and additional owners alike. If only an additional owner is set, the creator fallback no longer applies.
- **Tasks and reminders:** /aufgaben, the task mailer and the change dates take every owner into account; for changes without an implementer the planned window appears with all owners.
- **Documents:** Additional owners belong to the individual version and are copied into every new version like attachments and the reading obligation. A rejection in the workflow goes to all owners.
- **Deletion:** Deleting a vulnerability, a change or a document version also removes its entries.
- **REST API:** List and single retrieval return weitere_besitzer, the filter ?besitzer= matches main and additional owners; they are set via POST and PATCH (incidents: POST only).

Backup and SQL export
The backup covers objekt_besitzer like any other table. The per-area SQL export (chapter Data SQL Export/Import), however, transfers only the main owner – additional owners have to be set again after an import.

© 2012–2026 WolfInfoSec GmbH

3. Company Data and Structure Maps

Four pages describe the organisation itself rather than individual records: the company data and the three graphical views org chart, process map and network zones. Company data and org chart are open to administrators and master users only and are called from the admin area. Since v4.00 the process map and the network zones are menu entries in the group Organisation & Inventory, each with a permission of its own (see the chapter User Management). All three maps are listed in isms_helpers.MODULE_REGISTRY so that they can be switched off at /admin/module, after which modul_gate() blocks their addresses.

3.1 Company Data (/admin/unternehmen)

The tile "Company" leads to the master data of the organisation the ISMS is run for (admin_unternehmen.py). It describes the context of the organisation according to ISO/IEC 27001 clause 4; number of employees and revenue class additionally determine the size category under NIS2.

Field	Content
Company name	Mandatory; the org chart draws the root of the tree from it
Address	Multi-line, one line per address line
Legal form	Text field with a suggestion list (GmbH, AG, e. V. …); foreign legal forms can be entered freely
Number of employees	Full-time equivalents as a whole number
Revenue class	Four steps following the EU SME definition (recommendation 2003/361/EC)
Sites / countries	Multi-line, e.g. "Munich (DE)"
Industry	Free text
NACE code	NACE Rev. 2 or the German WZ subclass, such as "J", "62" or "62.01.0"; free text is rejected

The only system setting master users may edit as well.
Every other tile of the admin area is reserved for administrators. The data is held in the table unternehmen with exactly one row (id=1), just like smtp_settings and sso_settings; it is created on access. Other modules read the row via admin_unternehmen.unternehmen_daten().

3.2 Org Chart (/organigramm)

The org chart is the graphical view of the table Abteilungen (organigramm_modul.py) – one box per department or team, one connection per parent_id. There is no second data store: what is drawn is what /admin/abteilungen maintains as a table. The way there is the button "Graphical view" in the department administration; "Table view" leads back.

- **In the box**: head (Abteilungen.leitung_id), the assigned users (users.abteilung_id) and the number of assets, risks and measures of the department
- **Root**: the company name from the company data. It is not a record but the bracket around the top level – and the drop target for moving a department back up

© 2012–2026 WolfInfoSec GmbH

- **Re-assigning**: a box is dragged onto another one and thereby writes parent_id. Cycles are prevented on both sides, in the browser and on the server
- **Operation**: dragging pans the sheet, the mouse wheel zooms, a click opens the side panel with name, head, parent department, description and header colour
- **Output**: PNG, SVG and print (fitted to A4 landscape)

Database: no new table. Abteilungen receives the three optional columns beschreibung, sortierung and farbe by lazy ALTER TABLE when needed (isms_helpers.ensure_abteilung_zusatzspalten). No other module reads them.

3.3 Process Map (/prozesslandkarte)

The process map shows the processes in bands – the three classic management, core and support processes are preset and can be adjusted at /admin/kategorien (prozesslandkarte_modul.py). A box is a process, a band a process category. A process with parent_id does not get a box of its own but appears as a line in the box of its main process. The map is opened through the menu entry in the group Organisation & Inventory; access is governed by the permission prozesslandkarte (1 = read, 2 = edit).

- **In the box**: process owner (users.id), responsible department and the number of sub-processes
- **Process document**: if a document is linked (Dokumente.id), the side panel offers a link to it
- **Moving**: a box is dragged into another band or to another position within its band; the server re-assigns the sort values in steps of ten

Database: the self-creating table prozesslandkarte (name, kategorie, parent_id, verantwortlich_id, abteilung_id, dokument_id, beschreibung, sortierung, farbe). There are deliberately no foreign keys – if a column points nowhere after a user, a department or a document has been deleted, the process remains and the line in the box is simply omitted.

3.4 Network Zones (/netzwerkzonen)

The network zones show the network segmentation in bands that lead from the unprotected to the most strongly protected network, top to bottom (netzwerkzonen_modul.py), preset and adjustable at /admin/kategorien: external networks, DMZ/perimeter, internal networks, OT/production and high security/management. A box is a zone, a band a trust level. A zone with parent_id is not a box of its own but a system or subnet inside a zone. The map is opened through the menu entry in the group Organisation & Inventory; access is governed by the permission netzwerkzonen (1 = read, 2 = edit).

- **In the box**: zone owner, responsible department, VLAN and address range as well as the number of systems
- **Protection requirement**: normal, high or very high – as a badge on the box
- **Network concept**: if a document is linked, the side panel offers a link to it
- **Moving**: as in the process map, dragging writes the trust level and the order

Database: the self-creating table netzwerkzonen – in addition to the columns of the process map with vlan, ip_bereich and schutzbedarf.

Bands: since v3.92 both maps read their bands from the category catalogue. The column kategorie stores the key of the band (such as kern or dmz); name, colour and order come from stamm_kategorie

© 2012–2026 WolfInfoSec GmbH

3.5 Plans & Registers (/plaene)

Alongside the drawn maps there is a repository for plans created outside WolfISMS (plaene_modul.py). Like the org chart it is open to administrators and master users only and is called via the tile “Plans & Registers” in the admin area; there is no menu entry. It can be switched off under /admin/module.

Repository	File types	Limit
Network diagrams (art = netzplan)	PNG, PDF, draw.io (.drawio, .xml)	32 MB
Processes (art = prozess)	PNG, PDF, draw.io (.drawio, .xml)	32 MB
Asset register (art = assetregister)	CSV, XLS, XLSX, ODS	16 MB

- **Content check**: what counts is the signature of the file, not its extension – an executable renamed to PNG is rejected. draw.io files must contain an <mxfile> or <mxGraphModel> element.
- **Display**: a PNG is shown directly, a PDF opens in a new tab. Deliberately without <iframe>: X-Frame-Options DENY and object-src none would leave an embedded PDF blank. draw.io files cannot be rendered (the viewer would be foreign JavaScript) – the detail page shows the page names and offers the download.
- **Asset register**: is only stored for now. The column importiert_am is reserved for a later import into the assets and stays empty until then.
- **Database and files**: table plaene_dateien (self-creating); the files are stored with a UUID prefix under <upload_dir>/plaene/<art>/ and are therefore part of the uploads backup.
- **Write protection**: with an expired licence the plans stay visible; only upload, edit and delete are locked.

3.6 Module Categories (/admin/kategorien)

Since v3.92 the categories of the modules that used to be fixed in the code are kept in one shared table stamm_kategorie (kategorien_katalog.py). They are maintained at /admin/kategorien (admin_kategorien.py) – one page with a tab per area, reached through the tile “Categories” in the group Risk & Catalogues and open to administrators only.

Area	Stored in	Kind
Incidents	Incidents.Kategorie	value list; “ohne” (none) is built in
Change Management	chm_changes.kategorie	value list
Legal Register	rechtsregister.kategorie	value list with colour
Contacts – Service Providers	kontakte_dienstleister.kategorie	value list; free text until now, initial values from the existing data
Network Zones – Bands	netzwerkzonen.kategorie	bands with key and colour, ordered from outside to inside
Process Map – Bands	prozesslandkarte.kategorie	bands with key and colour, ordered from top to bottom

- **Value lists**: the record stores the German name. Renaming updates all records via UPDATE; the English name is only what the English interface shows
- **Bands**: network zones and process map store a fixed key (e.g. kern, dmz). Name, English name, colour and order are free; renaming only changes the label. One band must remain

© 2012–2026 WolfInfoSec GmbH

- **Deleting**: is refused as long as a record uses the category
- **Initial filling**: on first access an area is filled from the former code defaults and the values already in use. stamm_kategorie_bereich remembers that this has happened – deleted defaults therefore do not come back
- **Other catalogues**: asset, risk, threat and vulnerability categories have their own tables and forms (/admin/asset_kategorien, /admin/risiko_kategorien, /admin/bedrohungen, /admin/schwachstellen); the page only links them as tabs
- **REST API**: POST /api/v1/incidents checks the category against the same catalogue – a category created here is immediately valid through the API as well

Column	Meaning
bereich	Key of the area from kategorien_katalog.BEREICHE
schluessel	Bands only: the stored key
name / name_en	German name (unique per area) and English name
farbe	Hex colour (#RRGGBB) for bands and the legal register
sortierung	Order in selection lists and maps

For developers:
The former constants are functions now: incidents.KATEGORIELISTE(), changemanagement_modul.KATEGORIEN(), netzwerkzonen_modul.KATEGORIEN() and prozesslandkarte_modul.KATEGORIEN(). A new area is an entry in kategorien_katalog.BEREICHE with table, column and defaults.

© 2012–2026 WolfInfoSec GmbH

4. Roles and Reading Obligation

A role describes the function somebody holds in the ISMS – information security officer, data protection officer, head of IT, auditor. It is not a permission: what a user may see and change is still decided solely by the permission columns of the users table (see the User Management chapter). The role answers the other question – who holds which task.

Roles are company-wide master data in the table Rollen and are needed in three places: as the columns of the access matrix, as an attribute in the personnel module (Personal.Rollen) and – since v3.71 – as an assignment to user accounts. The reading obligation matrix of the policy portal builds on that assignment.

4.1 Role Management (/admin/rollen)

The tile "Roles" in the admin area leads to the role management (admin_modul.admin_rollen). It is open to administrators and master users. The list shows for each role the short code under which it appears in the access matrix, whether it has a column there, and who holds it.

Function	Behaviour
Create	the name is mandatory and is checked against the column width by Pruefung. A name already in use is rejected – the comparison ignores upper and lower case, exactly as the role catalogue does
Rename	the same duplicate check; the catalogue entry of the access matrix follows along
Members	the users of the role are ticked (isms_helpers.set_role_users). This is the opposite direction to the multiple selection in the user form – the same table, maintained from the other side
Active in the matrix	toggles the active flag of the role catalogue. It decides only whether the role gets a column in the access matrix, not who holds it
Delete	only if the role is no longer used anywhere (see below)

Before deleting, the form checks four places: the users of the role (user_rollen), the occupied cells of the access matrix (zugriff_matrix with a level other than "n" and "–"), the personnel records that

© 2012–2026 WolfInfoSec GmbH

carry it in Personal.Rollen, and the documents that name it in the reading obligation matrix (dok_lesepflicht). If it is still in use, the form reports where – and nothing is deleted. To remove a role from the access matrix only, set it to "not active" there instead of deleting it.

A fresh installation starts with five roles.
On first access ensure_rollen_tabellen() creates information security officer, data protection officer, management board, head of IT and auditor – but only if the table is created at that moment. An existing installation brings its own list and is left untouched; a list deliberately emptied does not fill up again.

4.2 Assignment to Users (user_rollen)

Up to v3.70 the role of a user was held in the column users.rolle – which takes exactly one role id. Somebody who is both the information security officer and the data protection officer could not be represented. The assignment therefore now lives in the n:m table user_rollen.

- **User form**: /admin/create_user and /admin/change_user show a multiple selection of all roles. Roles without a column in the access matrix are marked as such but remain selectable
- **User list**: /admin/all_users carries a "Roles" column; so does the Excel export of the user list
- **Profile page**: every user sees under "My roles" which roles are assigned to them
- **Legacy column**: users.rolle remains and is written on as a mirror of the alphabetically first role (dual write, as with Abteilung/abteilung_id). Every place that only reads this column keeps working unchanged
- **Taking over existing data**: if ensure_rollen_tabellen() creates the table user_rollen, it takes over the previous single role from users.rolle once. Only what points to an existing role is taken over; orphaned references and placeholders such as "none" are dropped. The backfill is tied to the creation of the table and not to whether it is empty – otherwise deleted assignments would come back on the next page view

4.3 Role Catalogue of the Access Matrix

The access matrix keeps a catalogue entry for every role, with a short code (R01…Rnn) and an active flag. It is the same role list as under /admin/rollen, only with the additional details the matrix needs. To keep it the same list, the admin form does not touch the catalogue and matrix tables itself but calls three functions in zugriffsmatrix_modul: a role created there gets its catalogue entry right away, a deleted one takes catalogue entry and matrix cells with it, and the active flag is visible in both forms. All three work regardless of whether the access matrix module is switched on under /admin/module – the tables should match either way.

Licence prerequisite: The access matrix is part of Advanced (ADV, with and without “+”) and Enterprise (ENT); in the others its role catalogue is locked as well. Creating, renaming and deleting roles is unaffected – that happens at /admin/rollen and is available in every edition. Where a screen points at the role catalogue (the reading obligation matrix of the portal, for instance), in the other editions the link leads there instead.

4.4 Reading Obligation Matrix (/portal/lesematrix)

Until now the policy portal held that every released document of a type with a reading obligation had to be read and confirmed by every user. Anyone who only needs the visitor policy also got the cryptography policy on their reading list. Since v3.71 a matrix of document and role decides who

© 2012–2026 WolfInfoSec GmbH

has to read what (table dok_lesepflicht). It is maintained by the master user at /portal/lesematrix and is reached through the button "Reading obligation matrix" in the portal.

- **Rows**: the released documents (status 6) of the types for which a reading confirmation is intended – the positive list LESE_TYPEN in portal_modul.py
- **Columns**: the roles from the table Rollen, with the number of their users. A toggle per column sets or clears the whole column
- **A tick**: whoever holds this role must read and acknowledge the document
- **An empty row**: nobody has to read it. The document stays visible in the portal and can still be acknowledged voluntarily, but it does not count towards progress, read status and reminders
- **Logging**: every changed row goes into the activities together with the number of roles assigned

The matrix is deny by default – and there is deliberately no backfill.
A newly released document has no row and therefore no reading obligation; the same applies to existing documents after the update. The old state "everybody must read everything" is not carried over but set once, deliberately. After the update to v3.71 the matrix must therefore be filled first – otherwise WolfISMS assigns nobody a reading obligation and sends no more reminders.

So that this does not go unnoticed, the document module speaks up: if a policy is released for which no role is entered, a note points out that nobody has to read it and that no mail is triggered. Master users are additionally told the way into the matrix. When a document is deleted, its row disappears from the matrix as well.

The matrix takes effect in five places:

Place	Effect
Policy portal	the progress bar counts the mandatory reading only. Documents without a reading obligation carry the label "Optional" and appear behind the mandatory ones
Tasks	outstanding read confirmations only appear for documents the user actually has to read
Reminder mailer	only the mandatory reading is chased; a user without a role receives no reminder
Release mail	the mail sent on release goes to the users of the assigned roles instead of to everybody. If no role is assigned, no mail goes out
Personnel module	read status and Excel export count the mandatory reading per employee. The matrix in the export knows three states: read, outstanding, no obligation

4.5 Database

All three tables are created by the helpers in isms_helpers on demand (CREATE TABLE IF NOT EXISTS on first access) – no migration script is required.

Table	Structure and purpose
Rollen	id, rolle – the role list itself. It has always existed; what is new is only the initial set on a fresh installation
user_rollen	user_id, rolle_id as a composite primary

© 2012–2026 WolfInfoSec GmbH

Table	Structure and purpose
	key, plus an index on rolle_id. Assigns users and roles to each other (n:m)
dok_lesepflicht	dok_id, rolle_id as a composite primary key, plus an index on rolle_id. Defines which role must read which document

Both evaluations join via Rollen.
The queries of the reading obligation include the table Rollen. A row whose role no longer exists therefore drops out by itself and can no longer trigger a reading obligation.

© 2012–2026 WolfInfoSec GmbH

5. Data Backup

The backup module creates complete system backups (database + uploaded files) as a tar.gz archive and supports restoring them through the web interface.

5.1 Backup Content

Archive path	Content
db/isms_db.sql	Database dump (mariadb-dump or mysqldump --single-transaction --routines --triggers), stored uncompressed inside the archive; host, port and TLS come from the database connection
uploads/	A 1:1 copy of the upload directory (config.upload_dir) with all uploaded files
Not included	Application code (/var/www/isms/), configuration files (config.py, /etc/wolfisms/secret.env, isms.conf) and the licence file are NOT backed up – back these up separately (e.g. version control, configuration management)
Archive name	isms_backup_YYYYMMDD_HHMMSS.tar.gz

5.2 Creating a Backup

- Admin area → "Backup & Restore" → button "Create backup"
- Alternatively: direct link /backup/download for an immediate download

The archive is created in /tmp and deleted automatically after the download.
For automated backups: a dedicated cron job using curl/wget against /backup/download, or mysqldump + tar directly on the server. The schedules configured at /admin/cron_zeiten only control the background jobs for task reminders and the risk score history – not the backup.

5.3 Restoring a Backup (UI)

- Admin area → "Backup & Restore" → choose a backup file (isms_backup_*.tar.gz, or a plain .sql/.sql.gz dump from an older backup)
- Combined backup (.tar.gz): restores both the database (db/isms_db.sql) and the uploaded files (uploads/ → config.upload_dir); the previous upload directory is automatically preserved as a timestamped copy (*.bak_YYYYMMDD_HHMMSS)
- Plain database dump (.sql/.sql.gz): restores the database only, uploaded files remain unchanged
- An Apache restart is not required for the restore itself; afterwards systemctl restart apache2 is still recommended to clear running sessions and caches

Endpoint / behaviour	Details
POST /backup/restore	Multipart form, field file: either a complete backup (.tar.gz/.tgz) or a plain database dump (.sql/.sql.gz/.gz/.gzip); the format is detected automatically from the file extension and magic bytes
HTTP 200	Also returned on partial failure – the authoritative value is the status field ("ok"/"error") in the JSON body, not the HTTP status
HTTP 403	Not signed in as an administrator

© 2012–2026 WolfInfoSec GmbH

Endpoint / behaviour	Details
HTTP 400	No file selected or unsupported file format

5.4 Additional Checks During Restore

The restore procedure has been hardened against incomplete and manipulated archives:

- **Table scan before import**: the dump is inspected for the tables it contains beforehand; a dump without recognisable tables is rejected
- **Pre-drop**: the target tables are removed in a controlled manner before the import so that no remnants of an older schema survive
- **User snapshot**: the existing user accounts are captured before the import so that signing in remains possible after a failed restore
- **Post-import verification**: after the import the expected tables and data are checked for presence; the result appears in the restore log
- **Path traversal and symlink protection**: _safe_extract_tar() validates every archive entry; crafted archives are rejected with an error
- **Schema markers:** After a successful restore the application resets the markers of the lazy ALTER helpers (isms_helpers.einmal_alle_zuruecksetzen) so that retrofitted columns are created again if the dump carries an older schema. This only applies to the executing process – with several workers the restore log advises a restart

Caution: a restore irreversibly overwrites existing files and the entire database.
Always create a current backup of the present state before restoring. Like all admin routes the endpoint is only accessible to current_user.admin.

5.5 Manual Restore (Command Line)

```
tar -xzf isms_backup_*.tar.gz -C /tmp/restore/
# Restore the database from the dump inside the archive
mysql -u isms -p isms < /tmp/restore/db/isms_db.sql
# Restore uploaded files (target path = config.upload_dir)
cp -r /tmp/restore/uploads/. /var/www/isms/wolfisms/uploads/
chown -R www-data:www-data /var/www/isms/wolfisms/uploads/
# Application code, config.py, secret.env and the licence file are NOT part of
# this archive and may have to be restored separately
systemctl restart apache2
```

© 2012–2026 WolfInfoSec GmbH

6. Licence Management

WolfISMS validates a cryptographically signed licence file at start-up. The Flask secret key is derived directly from the licence ID – a valid licence is therefore mandatory for production use. Without a licence the system runs in demo mode with a maximum of one active user.

Component	Path / description
Validation binary	/var/www/isms/bin/license_check
Public key	/var/www/isms/license/license_public.pem
Licence file	/var/www/isms/license/customer.lic
Licence fields	Customer, Address, Tier, Expires, LicenseID
Secret key	Random and persistent: ISMS_SECRET_KEY from the environment or, if empty, the file .flask_secret_key created in the session directory on first start (0600; isms_helpers.app_geheimnis()). Not derived from the licence – a licence change keeps sessions valid
Demo mode	No licence → max. 1 active user, warning in the log
Upload via UI	/admin/lizenz_upload – upload a new .lic file and activate it immediately

After replacing a licence Apache must be restarted: systemctl restart apache2
The licence check is cached (lru_cache) – without a restart the old licence remains active. Replacing the licence changes the secret key and invalidates all existing user sessions.

6.1 Behaviour With an Expired Licence

If the licence expires, the system remains readable but write-protected. Module visibility does not change – only write operations are blocked (isms_helpers.eff_readonly()).

- **Regular users**: can only read; save and edit buttons are disabled
- **Administrators and master users**: keep write access so that a new licence can be installed and the system administered (isms.has_master_privileges())
- **Safety net**: if no user is flagged as master, the oldest user in the database is treated as one – so nobody is permanently locked out

6.2 Editions and Feature Scope

The field Tier of the customer.lic controls three things: how many users with write access may be created, which modules are available, and whether single sign-on and the REST API are unlocked. There are two editions, Essentials and Advanced, each for up to 5, 10 or 25 users with write access, plus Enterprise for more than 25 users or several installations within a group. The number of users is part of the code (ADV10 = Advanced for up to ten users); an appended “+” unlocks the add-on SSO & REST API (ADV10+). Without a valid licence file the installation counts as DEMO, the free edition.

Edition (tier)	Users with write access	Modules	SSO, REST API	AI features
DEMO (free)	max. 1	Core modules	locked	locked

© 2012–2026 WolfInfoSec GmbH

Edition (tier)	Users with write access	Modules	SSO, REST API	AI features
Essentials ESS5, ESS10, ESS25	max. 5, 10 or 25	Core modules	locked	available
Essentials Plus ESS5+, ESS10+, ESS25+	max. 5, 10 or 25	Core modules	available	available
Advanced ADV5, ADV10, ADV25	max. 5, 10 or 25	all	locked	available
Advanced Plus ADV5+, ADV10+, ADV25+	max. 5, 10 or 25	all	available	available
Enterprise ENT	unlimited	all	available	available

Only users with write access are counted, that is active accounts without the readonly flag (Portal Only). Read-only users, and with them every portal user, never count towards the limit in any edition; their number is unlimited everywhere. How and where the limit is enforced is described in the section User Limits of the Editions. In the free edition a volume limit for assets, risks, documents and suppliers applies on top – see the following section.

The core modules, and therefore every edition, comprise documents including document control, the policy portal, assets, risks, measures, change management, tasks, suppliers, incidents, controls including the SoA and data protection, plus the structure maps (org chart, process map, network zones, plans & registers) and the links. The free edition has the same modules as Essentials. The AI assistants are included in every paid edition.

Thirteen modules belong to Advanced and therefore to Enterprise as well:

- **Objectives:** /ziele – corporate and security objectives with KPIs
- **Personnel:** /personal including onboarding/offboarding and training
- **Contacts:** /kontakte – address book and emergency contacts
- **Reports:** /berichtsuebersicht – the individual reports remain available in their modules
- **Projects:** /projekte – shares the mass permission
- **Operational plan:** /op_plan – calendar of recurring mandatory tasks, master users only
- **Audit:** /audit – internal audits, checklists, findings
- **Management review:** /managementreview – shares the audit permission
- **Vulnerabilities:** /schwachstellen including the AI comparison against known CVEs and the /api/v1/schwachstellen endpoints
- **Legal register:** /rechtsregister – shares the norm permission
- **Playbooks:** /playbooks including the attachment and link routes /play_upload, /play_attachment and /play_attachment_delete
- **BCM:** /bcm – BIA per asset, contingency plans, exercises
- **Access matrix:** /zugriffsmatrix including the matrix role catalogue; roles themselves remain manageable at /admin/rollen in every edition

In Free and Essentials these menu entries are missing, and calling one of their URLs directly ends on a notice page. The measures view hides the link to the management review, the reminder emails list no operational plan dates, and the attachment and link routes of playbooks and BCM

© 2012–2026 WolfInfoSec GmbH

plans in isms.py answer with 403 (isms._require_module_access with the mapping _FLAG_MODUL). Recorded data is preserved and is there unchanged after a licence change.

The add-on “+” comprises two features:

Feature	Areas affected	Behaviour without “+”
REST API	All endpoint groups under /api/v1/ and the API key administration at /admin/api_keys (see the REST API chapter)	Every call ends with 403 and a JSON error object, before the API key is even checked; the field tier_required lists every permitted tier. Keys already issued remain stored
Single sign-on	Sign-in via Microsoft Entra ID and Google, the /admin/sso screen (see the single sign-on chapter)	Switches and credentials are neutralised as they are read: the sign-in page shows no SSO button and the callbacks fall back to password authentication. The values in sso_settings are kept

The “+” is a selection, not a threshold.
Advanced contains every module, but SSO and the REST API only with “+” – and Essentials Plus has SSO and the REST API, but not the Advanced modules. A rank comparison cannot express that. The code therefore holds the enumeration isms_helpers.PLUS_TIERS (every tier with “+” and ENT), which API_TIERS and admin_sso.SSO_TIERS both draw on.

What never depends on the edition.
The core modules are available in every edition – they are enough to run the ISMS. For certification and for demonstrating compliance to an auditor – audits, the management review, BCM, the legal register, objectives and key figures – Advanced is intended.

The tier is read by isms_helpers.lizenz_tier(). Within a request flask.g holds the result, outside of one (cron scripts) it is cached for 60 seconds – every check would otherwise start the license_check binary. There are two kinds of check: tier_ok(min_tier) compares the tier with a minimum tier from the order LIZENZ_TIER_RANK (DEMO, Essentials, Essentials Plus, Advanced, Advanced Plus, Enterprise – the number of users plays no part in it), and tier_ok_an(requirement) checks it against an enumeration of permitted tiers. tier_stufen() reduces both notations to one tuple, tier_namen() turns that into the wording for the notice page and the error object and groups the three user sizes (“Advanced (ADV5/10/25) or higher”); tier_beispiel() suggests a matching tier on the notice page that is never a step down. The field tier in isms_helpers.MODULE_REGISTRY accepts both: the minimum tier of Advanced for projects, the operational plan, audit, management review, vulnerabilities, legal register, playbooks and BCM, the enumerations NICHT_ESS_TIERS for objectives, personnel, contacts and reports and ZUGRIFF_TIERS for the access matrix. The number of users lives in USER_LIMIT_TIER.

Three kinds of gate enforce it: isms_helpers.modul_gate() – the before_request hook isms.enforce_module() – checks the edition first and the administrator switch second, centrally for every blueprint of a module; isms_helpers.tier_gate() serves as a second barrier inside the blueprint (access matrix, /admin/sso) or directly inside a single route (/admin/api_keys) and renders the notice page; isms_helpers.api_tier_gate() sits in front of the REST blueprints, answers with JSON and checks independently of the login, because the API only knows the X-API-Key. For routes outside the module blueprints isms._require_module_access() checks via modul_aktiv().

© 2012–2026 WolfInfoSec GmbH

The AI features take their own route via admin_ki_modul.ki_lizenz_ok() – see the chapter on AI settings.

6.3 Volume Limits of the Free Edition

Without a valid licence file the installation runs as DEMO. Since v3.75 it is limited not only in the number of users but also in the amount of data: ten assets, ten risks, ten documents and ten suppliers. Every paid edition has no upper limit.

Area	Table	Limit in DEMO	What is counted
Assets	Assets	10	rows of the table
Risks	Risiko	10	rows of the table
Documents	Dokumente	10	distinct document numbers
Suppliers	vendor	10	rows of the table

What is counted is the stock in the database, not the number of entries ever created: a deleted object releases its slot again. For documents the document number counts instead of the row – a further version of the same document does add a row to Dokumente but does not use up any of the quota.

The check sits at every place that creates a new object:

- **Forms**: the form for a new entry cannot even be opened, and saving a new record is refused (assets_modul.py, risiken.py, dokumente_modul.py, lieferanten.py)
- **Import**: the asset import creates only as many new rows as the quota still allows and reports the skipped ones in its result
- **REST API**: POST to /api/v1/assets, /api/v1/risiken and /api/v1/lieferanten answers with HTTP 403 and the message in the field error
- **Generated documents**: risk treatment plan, statement of applicability and the NIS2 applicability assessment create evidence documents and therefore count as well; a new version of an existing document stays free

Existing data is left untouched.
The limit blocks creation only. Viewing, editing, exporting and deleting stay possible in every edition – even where an installation is above the limit after a licence has expired or been exchanged. To create again, delete an existing entry or install a paid licence.

Limits and check live in isms_helpers.py: FREE_LIMIT_ANZAHL sets the number, LIMIT_BEREICHE maps each area to its table and counting column, limit_erreicht() performs the check and limit_meldung() supplies the message in both languages. If counting fails – because the database is unreachable, for instance – the limit counts as not reached: a broken check must never block the application.

How far an installation has come is shown at /admin/lizenz_upload: in the free edition the page lists the state of each of the four areas below the licence information (7 / 10 for example), an exhausted area highlighted in red. In the paid editions the display is omitted because there is nothing to limit.

© 2012–2026 WolfInfoSec GmbH

6.4 User Limits of the Editions

Free allows one user with write access, Essentials and Advanced as many as the code says – 5, 10 or 25; Enterprise is unlimited. The “+” therefore does not raise the number of users, it unlocks SSO and the REST API: ADV10 and ADV10+ both allow ten users with write access. Counting happens in the users table with aktiv=1 and readonly=0. An account flagged "Portal Only" and a deactivated account therefore do not count – in every edition a company may let any number of employees take part in the policy portal.

The check runs at the three places that increase the number:

Path	Check
/admin/create_user	A new user without "Portal Only" is rejected once the limit is reached
/admin/change_user	Only if the user did not count as a write user so far and is about to become one. Editing a user who already counts always goes through
/admin/toggle_user_active	When reactivating an account without "Portal Only". Deactivating is never affected

Not checked are the user import at /admin/import_users and the self-registration on the first SSO login: both create read-only users only and therefore do not increase the number.

Only growth is blocked.
If an installation is above the limit after a licence change, it stays fully usable: all existing users keep signing in and can still be edited. Only the next additional write user is rejected. This is the same rule as for the volume limits of the free edition.

The numbers live in a single place: isms_helpers.USER_LIMIT_TIER. Around it are user_limit() for the limit of the current edition, user_anzahl() for the current count, user_zaehlt() for the question whether a given user already counts, user_limit_erreicht() for the decision and user_limit_meldung() for the message in the form. isms.check_lizenz() reads the same values and serves as the overall view. If counting fails – because the database cannot be reached, for instance – the limit counts as not reached: a broken check must never lock anybody out.

© 2012–2026 WolfInfoSec GmbH

7. Module Management

At /admin/module the administrator decides which modules the installation offers and in which order they appear within their menu group (admin_modulverwaltung.py). The page is reserved for administrators and is reachable through the “Modules” tile in the admin area.

7.1 Registry and Storage

Every module is listed with its menu image, target URL, label (German and English), module permission, required edition and the blueprints belonging to it in isms_helpers.MODULE_REGISTRY – that is the single place where a module is registered. The main menu, the URL lock and this management page all draw from it.

The administrator’s settings are stored in the table isms_module (modul, aktiv, sortierung). It creates itself on the first call; as everywhere in this project there is no migration script. Sort values are assigned in steps of ten, so that something can still be squeezed between two positions if needed.

Not every module in the registry has a menu entry: org chart and plans & registers are called exclusively from the admin area and appear in the registry only because of the URL lock and this management page. For all others: A menu entry appears when three conditions are met at once:

- **Switch**: the administrator has left the module enabled (field aktiv)
- **Edition**: the licence tier is sufficient (field tier). The field accepts a minimum tier or an enumeration of permitted tiers – the thirteen Advanced modules depend on it (see the Licence Management chapter, section “Editions and Feature Scope”)
- **Permission**: the signed-in user holds the module permission (field recht)

A broken management table must not empty the menu.
If the table isms_module is missing, empty or does not know a module yet, the module counts as active and keeps its position from the registry. A newly shipped module therefore appears without the administrator doing anything.

Since v3.90 the reports page (key berichte) is also in the registry with recht None. Its menu entry appears for all users with write access; which cards it shows is decided by the page per module under the same three conditions – see chapter Reports and Regulatory Reporting.

SoA: since v3.93 the Statement of Applicability has an entry of its own (key soa, target /berichte?berdetail=2, permission norm, no blueprint of its own – the route lives in bp_normen). It can be switched off at /admin/module like any module; the page checks this itself and then answers 404. menu() only shows the entry if iso27001_installiert() finds a non-deleted entry whose framework starts with “ISO/IEC 27001:” (one additional query per page view, cached); the SoA page itself still evaluates ISO/IEC 27001:2022

Several versions: since v4.00 the SoA shows every installed version of ISO/IEC 27001 (for example 2022 and 2024), by default the newest; a switch in the heading changes it, ?fassung= selects directly.

7.2 Menu Groups

Since v3.91 isms.menu() divides the sidebar into groups. They are listed with key, German and English heading and their module keys in isms_helpers.MENU_GRUPPEN; MODUL_GRUPPE maps module → group from it.

© 2012–2026 WolfInfoSec GmbH

Group	Modules (default order)
My Workspace	Policy Portal, Tasks
Governance & Compliance	Controls, SoA, Legal Register, Data Protection, Documents, Objectives
Organisation & Inventory	Assets, Access Matrix, Supplier, Personnel, Process Map, Network Zones
Risk & Treatment	Risks, Measures, Vulnerabilities
Operations & Resilience	Change Management, Contacts, Incidents, Playbooks, BCM
Planning & Review	Projects, Operational Plan, Audit, Management Review, Reports
(no heading)	Links – set apart by a line

- **Order**: modul_liste() sorts by group first, then by the stored sort value from isms_module. The groups themselves cannot be rearranged; an existing setting is kept within each group
- **Empty groups**: a group without a visible entry – switched off, not licensed or without module permission – is left out together with its heading. If only one group remains, as for a read-only user with the policy portal, the menu is drawn without any headings
- **Current entry**: modul_von_blueprint() determines the module of the request from the blueprint, or else from the active parameter of menu(). Its entry is highlighted and its group always open
- **Collapsed groups**: the browser remembers them in localStorage under wolfisms_menu_zu, per user and browser. Nothing is stored on the server
- **Menu icons**: Since v4.00 every entry starts with a coloured symbol tile carrying a white line icon instead of a PNG image. isms.MENU_ICON maps module → icon, the colour comes from the menu group (CSS class .mi-<group>), isms.menu_eintrag() renders it. The sidebar background follows the colour scheme; the active page is a pill in the accent colour, in the collapsed sidebar a ring around the tile
- **Compact view**: A button at the top of the sidebar, left of «, moves the entries closer together and shrinks symbol tiles and icons. The browser remembers the choice in localStorage like collapsing – per browser, not per account

A new module belongs in exactly one group.
If its key is missing from MENU_GRUPPEN, it ends up in the last group without a heading next to Links. It also needs an entry in isms.MENU_ICON; the icon name must exist in ui_icons._SVG, otherwise a filled square appears.

7.3 Operation

- **Switching on and off**: one checkbox per row; “Save” writes the state of all rows in one go
- **Order**: drag a row or use the arrow buttons of the row. What is saved is the order of the list, not the individual move. It only takes effect within a menu group; each row shows its group in front of the target URL, and a row dragged across a group boundary returns to its group after saving
- **Reset**: activates every module and restores the order of the registry
- **Locked rows**: modules above the current licence edition are greyed out with a padlock note. In Free and Essentials this applies to the thirteen Advanced modules; these rows can neither be switched on nor opened. The padlock note names the editions the module is part of
- **Mandatory modules**: the policy portal is marked “always on” and cannot be switched off – read-only users are redirected there, and a deactivated portal would be an endless loop. Even a direct UPDATE in isms_module is overruled when the settings are read

© 2012–2026 WolfInfoSec GmbH

7.4 What a Deactivated Module Means

- **Menu**: the entry disappears for every user, including administrators and master users
- **Addresses**: isms_helpers.modul_gate() blocks every URL of the module by blueprint name in a before_request of the app. A direct call ends with status 403 and a notice page
- **REST API**: the matching API area is blocked as well and answers with 403 and the error object {"success": false, "error": "… This module has been deactivated."} – the API key is fine, the module is not reachable
- **Data**: the tables are neither dropped nor emptied. Once the module is switched back on, the content is reachable unchanged
- **Effect**: the settings are held in flask.g per request and read fresh otherwise – a change takes effect immediately, no Apache restart needed

The admin and login areas always stay reachable.
Only the blueprints of the modules are blocked. Admin pages, login, the portal and the task overview belong to no module of the registry and therefore always pass through – an administrator cannot lock themselves out with this page.

7.5 Sidebar of the Detail Views

The detail views of the modules assemble their right-hand sidebar themselves as HTML. Since v3.94 two shared helpers in isms_helpers.py apply to it; new modules should use both.

- **Structure**: properties and metadata at the top (creator, created, last editor, changed, where applicable reviewer and review date), below that one <h4> heading per area with the count and “Manage”, and the entries as links. The separating line hangs on the top edge of the next heading and thus closes the previous section
- **sidebar_kuerzen(html, en)**: the finished sidebar is passed through once. Sections with a “Manage” link are cut to SIDEBAR_MAX = 5 entries, followed by “more … (+N)” to the same address as “Manage”. Sections without that link – read access, change log, versions, comments – stay complete. Used in documents, incidents, controls, measures, audit, projects, risks, vulnerabilities, change management, assets and suppliers
- **rueck_abschnitte_html(quellen, spalte, ziel_ids, wp, en)**: back references. Links are stored as a comma-separated ID list only on the record that creates them; the helper searches the source tables from _RUECK_QUELLEN (change, vendor, schwachstellen, projekte, risiko, since v3.95 also audit for risks and bcm for measures) for the IDs of the target and returns one section per source – only with entries and only if rueck_sichtbar() reports the module as active and readable. A new kind of link needs an entry in _RUECK_QUELLEN
- **Action bars**: since v3.95 without linking buttons. They led to the same linking views as “Manage” in the sidebar and have been removed in documents, assets, suppliers, risks, measures and incidents; the routes themselves stay. Header tables no longer repeat what the heading shows; owner, responsible persons and area/department sit in the metadata of the sidebar. The supplier questionnaire actions are in a <details> drop-down menu (ven-menu)

Target	Back references from
Assets	parent assets (field of the child assets), changes, suppliers, vulnerabilities
Risks, incidents	changes, vulnerabilities

© 2012–2026 WolfInfoSec GmbH

Target	Back references from
Measures	projects, vulnerabilities, BCM exercises
Documents	Risiko.dokumente, changes – across all versions of the chain; document↔document see chapter Document Control and Versions
BPMN	Dokumente.bpmn, as the list “Linked documents” below the diagram; one row per chain, superseded versions do not count

No migration is needed: nothing changes where links are stored. Since v3.94 the controls show documents, measures, risks and assets only in the sidebar; the tables below the description together with “Show all” and their helper functions are gone. Risks no longer have a header table; number and name are in the heading.

7.6 Interface since v4.00

With v4.00 the interface has been given a calmer, consistent look. The modules still build their HTML themselves as f-strings; consistency comes from a few central places, just as with the colour scheme:

Component	Location	Effect
Line icons	ui_icons.py, after_request hook apply_ui_icons in isms.py (runs before the colour scheme)	An emoji at the start of buttons, tabs, headings h1–h4 and symbol divs becomes <i class=wi-NAME> – a CSS mask in the text colour. New emoji: entry in _EMOJI, new shape: _SVG
Buttons and tabs	templates/template.html	Overrides the module colours: neutral grey with 8 px rounding or as a capsule. Only the main button (class btn-haupt, set automatically for the plus and floppy-disk symbols) and submit buttons of POST forms are filled in the accent colour #2E5F8A; danger buttons are red. Inline colours with !important still win and no longer belong on buttons
Lists	isms_ui.py: liste_kopf, liste_suche, liste_kacheln, liste_segment, liste_filter, liste_sortierung, liste_chips, liste_zeile, liste_rahmen, liste_zeile_form	All module and admin lists. Search filters in the browser (?search= only sets the initial value), every word must occur; paging has been dropped. Secondary actions such as export sit in the three-dot menu
Title tile	isms_ui.titel_kachel()	Main and detail views, the SoA with its individual reports and the policy portal show their heading in a white tile
Footer	templates/template.html	On the right “WolfISMS Version <version>” (isms.Version)
Light/dark	theme_modul, /design/modus	Button next to the language flag, chosen per user – see chapter Design and Colour Scheme

- **Images:** since v4.00 images/ holds only eleven files – flags, login buttons, read confirmation, About image, admin tile and the branding fallback. The field bild in MODULE_REGISTRY is gone, hover_button() expects an icon name. New symbols belong in ui_icons._SVG as an icon, not in images/ as a picture

© 2012–2026 WolfInfoSec GmbH

- **New modules:** build the list from the liste_* components; deleting in a row as a small POST form (class wl-zeile-knopf), maintenance tables with input fields as liste_zeile_form. A row of key-figure tiles belongs in theme_modul.INFOKACHEL_KLASSEN so that it can be switched off
- **Icon names:** check ui_icons._SVG before use – an unknown name yields a filled square

© 2012–2026 WolfInfoSec GmbH

8. REST API

WolfISMS offers a REST API for six areas: incidents, measures, assets, risks, suppliers and vulnerabilities. It is meant for machine integration – a SIEM reports incidents, a ticket system fetches open measures, a CMDB reconciles the asset inventory, a purchasing system reports suppliers, a vulnerability scanner delivers its findings, a reporting tool reads risks. Authentication uses an API key in an HTTP header rather than CSRF tokens.

Licence prerequisite: The REST API is part of the add-on “+” – that is Essentials Plus and Advanced Plus, recognisable by the “+” in the code – and of Enterprise (ENT). In every other edition the endpoints and the API key administration are locked; every call ends with 403 and a JSON error object before the API key is even checked. As of v3.85 the field tier_required in the error object lists every permitted tier. Keys already issued remain stored and become valid again as soon as a matching licence is installed.

This chapter covers the administrative side: which endpoints exist, how they are protected and how keys are managed. The full interface description with all fields, value lists, filters and examples is in the separate document “WolfISMS REST API”, which is written for the developers of the connecting systems.

8.1 Areas

Area	Base URL	Module	Particulars
Incidents	/api/v1/incidents	api_incidents.py	Missing columns (ticket number, sync/synctime, _en fields) are added by the module itself on load
Measures	/api/v1/massnahmen	api_massnahmen.py	Links to risks and assets are read-only
Assets	/api/v1/assets	api_assets.py	Permitted values (category, location, classification …) come from the database and can be queried via /werte
Risks	/api/v1/risiken	api_risiken.py	The rating is calculated from impact and likelihood and is not submitted. Since v3.72 the free text field chancen (column Chancen) is among the readable and writable fields; the module adds the column itself on load
Suppliers	/api/v1/lieferanten	api_lieferanten.py	Criteria are submitted as words; the score is calculated from them
Vulnerabilities	/api/v1/schwachstellen	api_schwachstellen.py	If the number is missing when a record is created, the API assigns it; the severity is derived from the CVSS score

© 2012–2026 WolfInfoSec GmbH

8.2 Endpoints

All six areas are cut the same way. <area> stands for incidents, massnahmen, assets, risiken, lieferanten or schwachstellen.

Method	URL	Function
POST	/api/v1/<area>	Create a record; the number must be unique. Only for vulnerabilities may it be missing – the API then assigns the next free number WEA-NNNNN
GET	/api/v1/<area>	List with filters, full-text search, sorting and pagination
GET	/api/v1/<area>/<number>	Retrieve a single record
PATCH / PUT	/api/v1/<area>/<number>	Write the synchronisation state (sync and synctime); PUT behaves exactly like PATCH
GET	/api/v1/<area>/werte	Permitted values from the master data – assets, risiken, lieferanten and schwachstellen only

PATCH changes a few fields only.
Any other field in the body is rejected with HTTP 400 instead of being silently ignored – a typo in the calling system surfaces immediately. For assets, risks and suppliers it stays with sync, synctime and weitere_besitzer, incidents add status, measures the ticket number and weitere_besitzer. Only vulnerabilities also accept status, score, besitzer and weitere_besitzer: there the remediation state often travels back from the reporting system. Everything else is maintained in the application.

8.3 Authentication

Every call must send the API key in the X-API-Key HTTP header. The database holds the SHA-256 hash only; the application shows the plain text exactly once, when the key is created.

Property	Details
Header name	X-API-Key
Key format	64 hex characters (32 bytes, secrets.token_hex(32))
Storage	SHA-256 hash only in table api_keys – the plain text cannot be reconstructed
Management	Through the admin UI: /admin/api_keys (create, enable, disable, delete)
Revoking	aktiv=0 takes effect immediately – the quickest way to shut down an integration

© 2012–2026 WolfInfoSec GmbH

Property	Details
CSRF	No CSRF token required – the API blueprints are CSRF-exempt
Permissions	A key carries no user role: it may read and create in all six areas

8.4 No rate limit on the API

Up to 3.51 the incidents API allowed 30 requests per hour for POST and 120 for GET. Those limits are gone: a sync client fetches and acknowledges records every few seconds and would otherwise run into HTTP 429 permanently.

```
# isms.py - the API blueprints are taken out of the count
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() rather than a high limit() value: otherwise the global defaults of 2000 requests per day and 500 per hour would apply on top. Access therefore rests entirely on the API key – one more reason to issue a separate key per connecting system and to delete keys that are no longer needed.

8.5 Synchronisation state

Each of the six areas carries the columns sync and synctime. This way it is not the connecting system that remembers what it has already transferred, but WolfISMS – a client that is rebuilt therefore does not resend old records.

Value	Meaning
sync = null	This record is not intended for synchronisation
sync = 0	Not yet synchronised – this is the client's working set
sync = 1	Synchronised
synctime	Time of the last synchronisation; set to now automatically when sync=1 is written without it

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

# 2. Acknowledge after transferring
curl -X PATCH "https://host/isms/api/v1/massnahmen/MAS-0042" \
     -H "X-API-Key: $KEY" -H "Content-Type: application/json" \
     -d '{"sync": 1}'
```

© 2012–2026 WolfInfoSec GmbH

8.6 Managing API Keys

1. Open the admin area: /admin → "API keys"

2. Enter a description (e.g. "SIEM integration") and click "Generate API key"

3. Copy the displayed clear-text key immediately – it is shown only once

4. Keys can be disabled (blocked) or deleted permanently at any time

Security note: API keys grant read and write access to all five API areas. Use a separate key for every external application and disable keys that are no longer needed without delay.

8.7 Testing without the REST console

The REST console at /admin/rest_console was removed with v4.00, as was the SQL console at /admin/sql_console – both including their code (admin_rest_console.py, admin_sql_console.py) and tile. To test the endpoints, use an ordinary HTTP client such as curl; the REST API documentation contains example calls for every area. Queries directly against the database run through the tools of the database server, no longer through the application.

© 2012–2026 WolfInfoSec GmbH

9. Data SQL Export/Import

At /admin/daten_sql a tool is available for exporting and importing master data between WolfISMS instances (admin_daten_sql.py), e.g. to transfer templates from a test instance to production. The module replaces the former admin_dokumente_sql.py and now covers three areas plus a combined package.

9.1 Areas

Area	Table	Behaviour on import
Documents	Dokumente	Fresh draft: version/subversion = 0, status = draft, creator = importing user, creation date = import date; review and approval fields are cleared. The version chain starts anew as well: stamm_id and vorgaenger_id of the source instance are discarded and rebuilt from the document number
Risks	Risiko	The assessment (status, likelihood/impact before and after measures, treatment, risk class, score) is taken over 1:1. Reset are the responsibility and state fields as well as the link columns (massnahmen, assets, bedrohungen, schwachstellen, dokumente, normen, lieferanten), because they contain foreign IDs that would point at different objects on the target instance
Measures	Massnahmen	The content (text, priority, effort, cost, end date) is taken over, but the implementation progress of the source instance is reset: status = open, erledigt = 0, archiviert = 0, ticket number cleared
All (combined package)	all three	Exports all three tables as a single SQL file and imports it area by area, each block using the rules of its own area

9.2 Further Properties

- **File formats**: export as .sql for download; import of .sql or packed as .tar.gz/.tgz
- **Import package NIS2 standard risks:** Since v4.00 import/nis2_standardrisiken.sql is included – 70 unassessed risks (German and English) on Art. 20, 21 and 23 NIS2 and §§ 30, 32 and 38 BSIG. Load it under Data SQL › Risks › import SQL file. The risks arrive with status “open” and treatment “not assessed”; owner, creator and review are taken over by the importing user
- **New IDs**: the import only appends – existing records remain unchanged
- **Duplicate numbers**: if a number from the import file already exists, the next free number in the same scheme is assigned (e.g. RSK-0042 → RSK-0087) and reported in the log
- **Department mapping**: for risks and measures the department is mapped by its plain-text name onto the local Abteilungen table; if no matching name is found, department and abteilung_id remain empty
- **Mixed files**: because the blocks stay separated by table, a combined package can also be imported on a single-area tab (which then picks out only "its" table) and vice versa
- **Not included**: attachments, histories (dok_historie, Risiko_historie, score history) and links are neither exported nor imported – only the master data itself
- **Legacy routes**: /admin/dokumente_sql and /admin/risiken_sql still redirect to the new areas

Security note: access is restricted to current_user.admin (like the framework import). Because owner, reviewer and approver fields reference user IDs that may belong to different people on the target instance, imported records should be reviewed before approval.

© 2012–2026 WolfInfoSec GmbH

10. Two-Factor Authentication (2FA / TOTP)

WolfISMS supports time-based one-time passwords per RFC 6238 (TOTP) as a second authentication factor. Compatible apps include Google Authenticator and Microsoft Authenticator. The module is implemented in totp_modul.py.

10.1 Database Migration

Before first use the following SQL migrations must be executed once (already included in the current isms.sql):

```
-- Columns in the users table
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 or isms_settings)
ALTER TABLE Stammdaten ADD COLUMN IF NOT EXISTS totp_required TINYINT(1) NOT NULL
DEFAULT 0;
```

```
-- Alternative: isms_settings table
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 Routes and Workflow

Route	Function
/2fa/setup	The user sets up 2FA: display the QR code, generate the TOTP secret (pyotp.random_base32), verify with the first code
/2fa/verify	TOTP code entry after login (called after the password check when totp_active=1); a valid code completes the login
/2fa/disable	The user disables 2FA for their own account (sets totp_active=0, totp_secret=NULL)
/admin/totp	Administrator switch: enable or disable system-wide totp_required. With totp_required=1 users without active 2FA are redirected to /2fa/setup at their next login

10.3 Security Notes

- TOTP secrets are stored in the database in clear text – secure database access accordingly
- The QR code is rendered inline as SVG in the browser – the raw data does not make a server round trip
- Pip prerequisites: pyotp and qrcode (already covered in step 1 of the installation)
- If an authenticator app is lost, the administrator resets 2FA for the account; this also clears the failure counter and the lockout. Alternatively directly in the database: UPDATE users SET totp_secret=NULL, totp_active=0 WHERE id=<USER_ID>;

© 2012–2026 WolfInfoSec GmbH

10.4 Lockout after failed attempts

Up to 3.51 a six-digit code could be guessed without limit. Since 3.52 WolfISMS counts failed attempts per account in the database and temporarily locks code entry. The count deliberately lives in the database rather than in process memory: with several WSGI workers it would otherwise apply per process and be easy to work around.

Property	Value
Failed attempts before lockout	5
Lockout duration	15 minutes
Columns	users.totp_fehlversuche and users.totp_gesperrt_bis – added by the module itself on load
Additionally per IP address	60 POSTs per 15 minutes; only POSTs are counted so that merely opening the page does not consume the allowance
Clearing	The administrator’s 2FA reset clears the counter and the lockout as well

Also since 3.52, enforced 2FA enrolment (totp_required=1) can actually be completed: /2fa/setup resolves the account from the pending sign-in and logs the user in as soon as the first valid code arrives. Previously the redirect ran into @login_required and ended back on the login page.

© 2012–2026 WolfInfoSec GmbH

11. Single Sign-On (Microsoft / Google)

WolfISMS can optionally delegate authentication to Microsoft Entra ID (Azure AD) or Google, implemented with OpenID Connect (authlib). Both methods can be enabled independently; classic sign-in with email and password remains available alongside them.

Licence prerequisite: Single sign-on is part of the add-on “+” (Essentials Plus, Advanced Plus) and of Enterprise (ENT). In the other editions the /admin/sso screen is locked and admin_sso.sso_config() neutralises switches and credentials: the sign-in page shows no SSO button, and the callbacks /auth/microsoft and /auth/google fall back to password authentication. A sign-in that was already set up therefore does not keep working after a licence change; the values in sso_settings are kept.

11.1 Activation

The credentials are held either in the environment (.env / secret.env) or in the table sso_settings maintained at /admin/sso. Which source applies is decided by a single switch at the top of the form; admin_sso.sso_config() supplies the resulting values to every sign-in path.

Switch "Use the values from this form"	Effect
off (default)	The values from .env / secret.env apply – unchanged previous behaviour. The form is saved but not used.
on	The row from sso_settings overrides the environment completely, including the on/off switches MS_AUTH and GOOGLE_AUTH.

The switch is deliberately a single one.
Only this way can an SSO enabled in the environment be switched off again through the interface. A per-field "empty = environment" could not express that and would never reveal which value currently applies.

11.2 Setup through the interface (/admin/sso)

Step	Procedure
1. Register the application	Create an application in the Entra portal or the Google Cloud Console. The redirect URIs to be entered are shown by the form in the section "Redirect URIs for the provider consoles" – they are derived from WEBPATH
2. Enter the credentials	Microsoft: directory (tenant) ID, application (client) ID and client secret. Google: client ID, client secret and optionally a different redirect URI
3. Set the domain allowlist	Permitted email domains separated by commas, e.g. example.de, example.com. Empty means nobody may sign in via SSO
4. Enable the method	Tick "Enable Microsoft sign-in" and/or "Enable Google sign-in", plus the switch at the top of the form
5. Check	"Save & check discovery" fetches the provider's OIDC discovery document and reports the issuer or the error. No restart is required – the OAuth client is built during sign-in with the values valid at that moment

© 2012–2026 WolfInfoSec GmbH

- **Secrets**: have been stored AES-256-GCM encrypted since v3.99 – the same way as the AI API key (chapter AI Settings), the key comes from the application's secret key – and are shown masked in the form. A value still Base64-encoded from before is converted the first time the table is read. An empty field keeps the stored secret, a "delete" tick removes it
- **Tenant ID**: GUID, verified domain or common / organizations / consumers. The value ends up in the discovery URL and is therefore checked strictly
- **Effective values**: the foot of the page shows which values currently apply and which source they come from
- **Incompletely configured**: if the tenant or client ID is missing, /auth/microsoft or /auth/google does not lead nowhere but back to the break-glass login with a message

11.3 Setup through the environment

Step	Procedure
1. Register the application	Create an application in the Entra portal or the Google Cloud Console and store the callback URL (https://host/auth/microsoft/callback or https://host/isms/auth/google/callback)
2. Enter the credentials	Store MS_TENANT_ID/MS_CLIENT_ID/MS_CLIENT_SECRET or GOOGLE_CLIENT_ID/GOOGLE_CLIENT_SECRET in .env
3. Set the domain allowlist	Fill SSO_ALLOWED_DOMAINS with the permitted email domains, e.g. SSO_ALLOWED_DOMAINS=example.de,example.com
4. Enable the method	Set MS_AUTH=1 and/or GOOGLE_AUTH=1
5. Restart	sudo systemctl restart apache2

11.4 Security Behaviour

- **Domain allow list**: sign-ins are only accepted if the email domain appears in SSO_ALLOWED_DOMAINS. An empty allow list admits nobody – that is the safe default, not a misconfiguration
- **Email verification**: with Google the email address must be marked verified by the provider (email_verified). Microsoft is already restricted by the tenant ID
- **Automatic creation**: without SSO_AUTO_CREATE=1 only users who already exist can sign in via SSO. With auto-create enabled unknown users are created – the allow list should then be kept especially tight
- **Blocked accounts**: users with aktiv=0 cannot sign in via SSO either
- **OIDC nonce**: a nonce is generated for every sign-in and validated against the ID token

11.5 Automatic Forwarding to Microsoft

With MS_AUTH=1 set, /login forwards to Entra by itself instead of showing the password form first. The jump is deliberately conservative so that the sign-in page can neither bounce in a loop nor swallow messages.

- **Once per session**: the session marker sso_auto_versucht prevents a failed SSO attempt from triggering the jump again straight away. After a successful sign-in it is reset
- **No jump after signing out**: logout sets sso_kein_auto. Without that lock the Microsoft session that is still open would sign the user back in silently

© 2012–2026 WolfInfoSec GmbH

- **Pending messages take precedence**: if flash messages are waiting (SSO error, rate limit, wrong password), the page stays put so that they can be read
- **Break-glass access**: the link "Sign in with email and password (emergency access)" calls /login?breakglass=1 and shows the password form – for example when Entra is unreachable or an account does not use SSO yet. A link leads back from there to the Microsoft sign-in
- **Google hidden in this view**: as long as the page jumps to Microsoft automatically, the Google button only appears on the break-glass page

Check before switching this on:
At least one administrator must be able to sign in through the emergency access. With MS_AUTH=1 set and Entra unreachable, the only remaining route is /login?breakglass=1.

© 2012–2026 WolfInfoSec GmbH

12. AI Settings

WolfISMS offers optional AI support in several places: the AI assistants of the modules (since v4.00 each on a page of its own: risks, assets, suppliers, documents, measures, vulnerabilities, changes, incidents, projects, controls, legal register, data protection, audits and playbooks), the AI translation of documents and the processing of a Word import (overview in the section AI Assistants at a Glance, operation described in the user documentation). All of them share the same provider configuration. It is maintained exclusively at /admin/ki_einstellungen, is accessible to administrators only and requires a paid licence (Essentials or above).

12.1 Licence Prerequisite: Every Paid Edition

The AI features are available in every paid edition – Essentials, Advanced and Enterprise, with and without “+”; only the free edition (DEMO) sees a notice. admin_ki_modul.ki_lizenz_ok() checks this via isms_helpers.tier_ok() against the rank of Essentials – analogous to the framework import. Without a valid customer.lic the installation counts as DEMO and is therefore locked. The result is held per request in flask.g, because every check starts the external license_check binary and a page asks several times.

The lock takes effect in four places, not only in the interface:

- **get_ki_config()** reports licensed=False and forces enabled=False
- **/admin/ki_einstellungen** shows a notice page instead of the form and does not save
- **ki_bearbeiten and ki_speichern** reject requests; so do the routes of all AI assistants
- **_call_ki_api()** aborts as a last safeguard should a call get through after all

The configuration is preserved:
The values stored in isms_settings are not touched by the lock and take effect again as soon as a sufficient licence has been installed. The admin tile marks the feature as locked instead of hiding it.

12.2 User Permission “AI Usage”

Since v4.00 every user needs the permission “AI usage” (column users.ki, 0/1) in addition to the licence and the configuration in order to use the AI assistants. It is granted in user management under System Permissions (/admin/create_user, /admin/change_user); the permission matrix and the Excel export list it as “KI”. Administrators may always use AI – as with the module permissions. The default is off, also for users who existed before the update: isms_helpers.kartenrechte_sicherstellen() creates the column with 0, the administrator grants the permission explicitly.

- **Buttons:** admin_ki_modul.ki_zugang_vorhanden() returns False without the permission – so all AI buttons and dialogues are missing, including the button “AI assistant” of the modules.
- **Routes:** a before_request hook in isms.py blocks every address with a path segment ki_… (for example /assets/ki_assistent, /dokumente/ki_bearbeiten). Pages redirect to the start page with a message, requests of the AI dialogues receive JSON with an error message. /admin/… is left out – AI settings and the usage report are administration.
- **Call:** _call_ki_api() refuses as a last safeguard; a background job (ki_job_starten) is checked when it starts and ends at once with the message if the permission is missing.

© 2012–2026 WolfInfoSec GmbH

12.3 Database Prerequisite (isms_settings)

The AI configuration and the totp_required flag are stored in the table isms_settings (key-value store). If the table does not exist yet:

```
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 Configuration Parameters

isms_settings key	Meaning / values
ki_enabled	"1" = AI active, "0" = disabled (default)
ki_provider	AI provider: anthropic | openai | mistral | google | nele | azure_openai | copilot | custom (default: anthropic). azure_openai and copilot (Azure AI Foundry) are served through the same call path. nele (since v3.93, nele.ai by GAL Digital GmbH, hosted in the EU) calls POST https://api.aieva.io/api:v1/chat-completion-sync with a bearer token; the key comes from chat.nele.ai. nele.ai assigns model IDs per account, they are entered by hand. Its built-in web search is offered as internet research. nele.ai bills in credits and reports no tokens – the usage evaluation only counts requests, without cost – and an answer cut off at the length limit is not continued automatically
ki_model	Model ID of the respective provider
ki_api_key	API key of the AI provider – since v3.99 stored AES-256-GCM encrypted in isms_settings (prefix enc1:; the key for it is derived via HKDF from the application's secret key)
	(not a hash, because the clear text is needed for the API calls). A legacy Base64-encoded value is converted on first read; shown masked in the admin UI
ki_api_base_url	Only for Azure OpenAI, Azure AI Foundry (Copilot) and custom: base URL of the endpoint. If it ends in /openai/v1, the new OpenAI-compatible Azure interface is used: the deployment then travels as the model name in the body and no API version is sent
ki_api_version	Only for Azure OpenAI / AI Foundry with a classic endpoint URL, e.g. 2024-10-21. Azure keeps introducing new versions and retiring old ones – the target URI of the deployment is what counts
ki_azure_auth	"key" (default) = sign-in with the api-key header; "entra" = token from Microsoft Entra ID instead of an API key. Only for azure_openai and copilot, since v4.00
ki_entra_tenant	Directory ID (tenant ID) of the Entra tenant
ki_entra_client_id	Application ID (client ID) of the app registration
ki_entra_secret	Client secret of the app registration – AES-256-GCM encrypted like ki_api_key. After saving, the form reads the value back as a test and reports "stored but not readable" separately from "empty"
ki_prompt_addendum	Optional additional prompt text appended to all AI requests (e.g. company-specific requirements)

© 2012–2026 WolfInfoSec GmbH

12.5 Sign-in via Entra ID (Azure)

Since v4.00 Azure OpenAI and Microsoft Copilot (Azure AI Foundry) can sign in via Microsoft Entra ID instead of an API key. This is needed for resources on which key-based authentication is disabled – often by Azure Policy (disableLocalAuth). The connection test then reports “Key based authentication is disabled for this resource” with HTTP 403.

- **Azure:** create an app registration with a client secret in the Entra admin center and grant it the role “Cognitive Services OpenAI User” on the AI resource under “Access control (IAM)”
- **WolfISMS:** at /admin/ki_einstellungen choose “Authentication: Entra ID” for the provider Azure OpenAI or Copilot and enter tenant ID, client ID and client secret; the API key field is then dropped. Base URL, model or deployment and API version stay as before
- **Flow:** admin_ki_modul._entra_token() fetches a token for https://cognitiveservices.azure.com/.default via client credentials and keeps it in the process until shortly before it expires. Only Authorization: Bearer is sent, no api-key header
- **AI configured?:** since v4.00 the modules check ki_zugang_vorhanden(cfg) instead of a stored API key – otherwise the AI buttons would be missing in Entra mode. New AI features must do the same
- **Secret not readable:** if the form shows “stored but not readable”, the process could not decrypt the value – usually because ISMS_SECRET_KEY comes from several sources (Apache envvars, /etc/wolfisms/secret.env, .flask_secret_key). Since v4.00 every process keeps the secret key fixed from start-up; enter the secret once more afterwards. The same applies to the client secrets of single sign-on

12.6 AI Assistants at a Glance

All assistants use the same connection (_call_ki_api) and the same licence gate. They are only visible when AI is switched on, access is configured (API key or Entra ID) and the user clears the hurdle of the view the assistant replaces; not in read-only mode.

Assistant	Location	Permission	Writes
AI assistant Risks	page of its own /risiko/ki_assistent, button in list and detail view	risk ≥ 2, not read-only	risks with rating and links; missing measures and suppliers
AI assistant Assets	page of its own /assets/ki_assistent, button in list and detail view	assets ≥ 2, not read-only	assets with protection needs (reason in bem_*) and links; missing suppliers, child assets and risks
AI assistant Suppliers	page of its own /vendor/ki_assistent, button in list and detail view	vendor ≥ 2, not read-only	suppliers, assessment and links; sends and evaluates the supplier questionnaire
AI assistant Documents	page of its own /dokumente/ki_assistent, button in list and detail view	dok ≥ 2, not read-only	new documents and versions, links, English version, hand-over for review; never approves
AI assistant Measures	page of its own /massnahmen/ki_assistent, button in list and	mass ≥ 2, not read-only	measures, status and evidence, links to all areas

© 2012–2026 WolfInfoSec GmbH

Assistant	Location	Permission	Writes
	detail view		
AI assistant Vulnerabilities	page of its own /schwachstellen/ki_assistent, button in list and detail view	schwachstellen ≥ 2, not read-only	vulnerabilities (free, CVE, advisory, scan or pentest report), rating, remediation evidence, links; replaces the vulnerability check
AI assistant Changes	page of its own /change/ki_assistent, button in list and detail view	assets ≥ 2, not read-only	changes and status steps under the same rules as /change/status; never votes
AI assistant Incidents	page of its own /incidents/ki_assistent, button in list and detail view	incidents ≥ 2, not read-only	incidents, classification, assessment of the reporting obligation, recorded reports, cause and lessons; never reports itself; replaces the AI analysis
AI assistant Projects	page of its own /projekte/ki_assistent, button in list and detail view	mass ≥ 2, not read-only	projects with profile, sections, roles, gates and links
AI assistant Controls	page of its own /normen/ki_assistent, button in list and detail view	norm ≥ 2, not read-only	status, implementation, evidence and related controls – only what is confirmed; never suggests an exclusion
AI assistant Legal register	page of its own /rechtsregister/ki_assistent, button in list and detail view	norm ≥ 2 or Admin, not read-only	entries (always “To review”), implementation, evidence, review dates; never “Not applicable” or “Compliant”
AI assistant Audits	page of its own /audit/ki_assistent, button in list and detail view	audit ≥ 2, not read-only	audits with checklist and assessment questions, suggested ratings, findings, audit report; questions in aud_pruefpunkte.beschreibung
AI assistant Playbooks	page of its own /playbooks/ki_assistent, button in list and detail view	play ≥ 2 or Admin, not read-only	playbooks with sections, procedure steps, roles, tests and links
AI assistant Data protection	page of its own /datenschutz/ki_assistent, button in list and detail view	datenschutz ≥ 2 or Admin, not read-only	processing activities as drafts, suggested threshold analysis, DPIA draft, TOM annex as a Word file; never “approved”, no residual values and no DPO opinion
AI processing of a Word import	edit form of a document, tick box in “Import from Word”	edit permission	nothing – the result is in the editor, saving is done with “Save”
AI translation	document, detail view, English interface only	dok ≥ 1 and owner; not approved, in review or superseded	Dokumente.text_en, on request Dokumentenname_en – as a new version
GAP assistant	audit of type gap: checklist and detail	audit ≥ 2	guides through the checklist; rating, finding and evidence per item

© 2012–2026 WolfInfoSec GmbH

Assistant	Location	Permission	Writes
	view		

- **Disclaimer**: every AI dialog shows the box from admin_ki_modul.ki_gewaehr_html() below its title – AI-generated content must be reviewed before use, no warranty is given for its correctness or completeness. A new assistant has to include it
- **Check the answer against the data**: suggested IDs are matched against the real catalogues; invented and already linked ones are dropped. The AI risk analysis takes categories, threats and vulnerabilities from the catalogues only, at most three per risk
- **Former dialogs:** Since v4.00 the risk assistant in a dialog window and “AI risk analysis” and “AI protection needs” on the asset have been merged into the two AI assistants; their buttons are gone. The routes in risiko_ki_modul.py and asset_ki_modul.py remain, the assistants use their core – the following rules still apply there
- **New risks from the AI risk analysis**: are created like in the setup assistant with status “offen”, treatment “nicht bewertet”, the department of the asset and the initial estimate as wahrvm/auswvm (values after measures 0), plus a row in Risiko_historie. The quantity limit of the edition applies, and existing risks the user may not change under the owner rule are skipped and reported
- **AI protection needs**: offers the levels of the table kritikalität without “kritisch” – the same as the edit form. Only the confirmed protection goals are written, since v3.95 with the justification in the remark column bem_<goal> (even if the level stays the same); value and inheritance are left alone, kritik_nachziehen() recalculates the criticality. If the asset is excluded from inheritance, the reason is passed to the model
- **Log**: linking, creating and changing are recorded in the activity log with the note “per KI” or “Schutzbedarf per KI”
- **AI links (dokument_ki_modul.py):** since v3.94, since v4.00 without a button of its own – the task links of the documents assistant (dokument_ki_verknuepfung.py) takes its place; routes POST /dokumente/ki_verknuepfen, …_stand and …_uebernehmen. The call runs as a background job (ki_jobs, area dokumente_link) because document text plus four catalogues can exceed Apache’s 60 seconds. At most 150 entries or 22,000 characters per catalogue are sent, presorted by keyword proximity; at most ten suggestions per area. It writes to the same places as the linking views /dokumente?link=…; documents that already refer from the other side count as linked. The blueprint belongs to the registry entry dokumente
- **Risk texts:** since v3.95 the risk assistant has the area texte (first in BEREICHE, only with risk ≥ 2): suggestions for description, business impact and opportunities (TEXTFELDER). Applying escapes the texts, converts them to HTML, checks them with Pruefung and writes to the columns of the display language (the _en columns in EN) – owner rule as for editing
- **AI translation (dokument_uebersetzung_modul.py):** since v3.95; routes POST /dokumente/ki_uebersetzen, …_stand and …_uebernehmen, blueprint dokument_uebersetzung on the registry entry dokumente. The model never receives HTML: zerlegen() splits the German text at every block tag (heading, paragraph, list item, cell, br, img) into sections, inline tags (strong, em, a, span …) become placeholders <x1>…</x1>, the content of pre stays untranslated, identical sections are sent only once. zusammensetzen() puts the translation back into the unchanged original HTML – tags, attributes and embedded images never come from the answer. If the placeholders of a section do not match, only its inline formatting is dropped; a missing section stays German; the dialog reports both. Blocks of 6,000 characters or 120 sections run as a background job with progress (area dokumente_uebersetzung); an unusable answer is halved and requested again. Locked like AI editing, saved like ki_speichern: sanitize_tinymce_html, length check against text_en, new version via _version_anlegen()

© 2012–2026 WolfInfoSec GmbH

12.7 Consultations: AI Assistants for Risks and Assets

Since v4.00 risks and assets each have an AI assistant on a page of its own: a conversation in which the AI creates, reviews and links entries together with the user. The button “AI assistant” sits in the list heading and in the detail view; from the detail view the conversation starts with that entry.

Property	Risks	Assets
Module, route	risiko_ki_assistent.py, GET/POST /risiko/ki_assistent[?risiko=<id>]; tasks in risiko_ki_neu.py, risiko_ki_verknuepfung.py, risiko_ki_stapel.py	asset_ki_assistent.py, GET/POST /assets/ki_assistent[?asset=<id>]; tasks in asset_ki_neu.py, asset_ki_verknuepfung.py, asset_ki_stapel.py
Tasks	Create a new risk (freely or starting from an asset, threat, vulnerability or incident), several risks from a list or upload, find gaps, review a risk, review a category, review the entire register	Create a new asset, several assets from a list or upload, review an asset, review an OBASHI level, review the entire inventory
Permission	risk ≥ 2	assets ≥ 2
Dialog state	table ki_dialog, area risiko	table ki_dialog, area asset

- **Layout:** chat on the left with an answer card below it, on the right the editable draft or a table of suggestions; the shared foundation is ki_dialog_bausteine.py. GET draws the page, POST (fetch from the browser) performs an action and returns the new page parts as JSON. Longer calls run as a background job (ki_jobs)
- **Writing:** nothing without confirmation. Draft and links are suggested and justified per area; only “Apply” writes. Whatever is created in another module – measures, suppliers, risks, child assets – checks that module’s permission
- **Lists:** Excel, CSV, PDF, Word or a screenshot can be uploaded; the AI reads the entries, the assistant matches them against the existing data and goes through them one by one
- **Batch review:** category, level and entire inventory start with the entries that need it most – for risks e.g. above the risk appetite without treatment or accepted without a reason, for assets without links, protection needs, owner or recovery targets
- **Conversation state:** one row per user and assistant; whoever comes back after a break can carry on there

12.8 AI Assistants of the Other Modules

Since v4.00 every module has an AI assistant following the pattern of risks and assets: the same page (chat, answer card, draft on the right), the same foundation (ki_dialog_bausteine.py), the same rule “nothing without confirmation”. The route is GET/POST <module>/ki_assistent, from the detail view with the ID of the entry; the permission is the write permission of the module (see the overview of the assistants). The state of the conversation is kept in ki_dialog, one row per user and area.

Module	File	Tasks	Area ki_dialog
Suppliers	lieferant_ki_assistent.py (tasks in lieferant_ki_neu, _verknuepfung,	create (free, from an asset or a document), several from a list/upload, gap analysis, review, type/tier, entire register, evaluate evidence, send and evaluate the questionnaire (also CAIQ,	vendor

© 2012–2026 WolfInfoSec GmbH

Module	File	Tasks	Area ki_dialog
	_fragebogen, _stapel)	VDA ISA/TISAX)	
Documents	dokument_ki_assistent.py (dokument_ki_neu, _verknuepfung, _stapel)	create, take over from documents, revise, review, merge, translate, hand over for review, link, gap analysis (mandatory documents, controls without a document), type, entire register	dokument
Measures	massnahme_ki_assistent.py (massnahme_ki_neu, _verknuepfung, _stapel)	create (free or from a risk, control, finding, incident, vulnerability, supplier, legal register), several from a report or measure plan, gap analysis, review, status round, evidence, entire register, order of implementation	massnahme
Vulnerabilities	schwachstelle_ki_assistent.py (schwachstelle_ki_neu, _verknuepfung, _stapel)	record (free, CVE, advisory), take over a scan/pentest report, check asset versions against known CVEs, review, triage, status round, prove remediation, entire register, status report	schwachstelle
Changes	change_ki_assistent.py (change_ki_neu, _verknuepfung, _stapel)	draft, review, accompany a status step, suggest a board, post-implementation review, status round, collision check in the change calendar, status report	change
Incidents	incident_ki_assistent.py (incident_ki_neu, _melden, _verknuepfung, _stapel)	record, review, merge, assess the reporting obligation under NIS 2, GDPR and CRA, prepare and record a report, cause and lessons, status round, status report	incident
Projects	projekt_ki_assistent.py (projekt_ki_neu, _verknuepfung, _stapel)	create (idea or project order), ISMS check, derive security requirements, risk assessment, gate review, closure, status round, status report	projekt
Controls	normen_ki_assistent.py (normen_ki_neu, _stapel; mapping normen_mapping.py)	answer questions, review a control, describe the implementation, check evidence, check exclusions for indications, gap analysis, mapping between frameworks, transfer the status of related controls, status round, status report	norm
Legal register	rechtsregister_ki_assistent.py (rechtsregister_ki_neu, _stapel)	questions, create entries, missing legal bases, review, implementation, evidence, review round, legal changes, status report	recht
Audits	audit_ki_assistent.py (audit_ki_neu, _pruefung, _stapel)	plan an audit with checklist and questions, rate checklist items, prepare findings, audit report, audit programme, follow-up, audit readiness, status report	audit
Playbooks	playbook_ki_assistent.py (playbook_ki_neu, _verknuepfung, _stapel)	create (describe a scenario or upload a document), review, prepare an exercise (tabletop), record a test result, follow-up after an incident, gap analysis, status round, status report	playbook

© 2012–2026 WolfInfoSec GmbH

Module	File	Tasks	Area ki_dialog
Data protection	datenschutz_ki_assistent.py (datenschutz_ki_neu)	create a processing activity, record as processor, gaps from the assets (on request only), review an entry, threshold analysis, draft a DPIA, TOM annex under Art. 32	datenschutz

- **Limits:** ratings, status and applicability are always suggestions. The change assistant never votes in the board, the incident assistant never reports to an authority, the controls assistant never suggests an exclusion (it only reports indications that an excluded control may be applicable after all), the audit assistant decides no rating – “implemented” only with a text on the effectiveness check
- **No legal advice:** the legal register assistant always creates entries as “To review” and suggests neither “Not applicable” nor “Compliant”
- **Data protection:** the data protection assistant gives no legal advice either – legal basis, retention period and result of the threshold analysis remain suggestions, approving and completing is done by a person
- **Web search:** the task legal changes only searches the internet if provider and model support it (admin_ki_modul.web_suche_moeglich) – otherwise it works without and says so. Hits come with their source
- **Hand-overs:** new measures are created in the measures assistant – from findings, from the legal register and from the other modules the assistant switches there
- **Licence:** every paid edition; the routes check ki_zugang_vorhanden() and the licence gate themselves. The modules the assistants belong to keep their edition

12.9 Length of the answers

Up to v3.71 WolfISMS called every model with a fixed max_tokens of 4,096. That is not enough for a rewritten policy – the answer broke off mid sentence after around 7,500 characters. Since v3.72 three stages apply:

- **Limit per model**: the default is 16,384 tokens. Models with a lower ceiling are listed in a table inside the module – claude-3-5 and claude-3-7 with 8,192, claude-3, gpt-4, gpt-4-turbo and gpt-3.5 with 4,096
- **Fallback on rejection**: if the provider rejects the value with HTTP 400, the ceiling it names is read out of the error message and the call is repeated. If it expects max_completion_tokens instead of max_tokens, that is adjusted too
- **Continuation**: if the answer still ends at the limit, WolfISMS fetches the rest in up to three further rounds; the text so far goes back to the model as the starting point

One call including its continuations has a time budget of 600 seconds; a further round only starts if at least 30 seconds of it are left. The frame for this is the ingress, which keeps a response open for 300 seconds (k8s/09-ingress.yaml) – better the text written so far than a dead connection. The values are constants in admin_ki_modul.py (KI_TOKENS_STD, KI_FORTSETZUNGEN, KI_GESAMTZEIT, KI_MINDESTZEIT) and deliberately not part of isms_settings.

© 2012–2026 WolfInfoSec GmbH

The database has to hold the longer answer as well.
Dokumente.Text has always been mediumtext, whereas the English version text_en was TEXT – 65,535 bytes, which MySQL silently truncates outside strict mode. The document module therefore raises that column to mediumtext once. If the database user lacks the ALTER privilege, the smaller limit stays: a text that is too long is then reported instead of being silently cut.

12.10 Usage and Cost

No provider releases an account balance for the working key stored under AI Settings. Anthropic and OpenAI require a separate admin key for their usage and cost APIs – one that may also manage users and disable keys – and Azure does not bill through the key at all, but through the cost analysis of the subscription. Such a key does not belong in an ISMS. WolfISMS therefore records consumption itself: every answer carries its own usage in the usage field, which is added up per call and stored in the ki_verbrauch table. The module creates that table on first use.

Recording happens in a single place – in _call_ki_api() in admin_ki_modul.py, through which all AI features run. Each row holds the time, the user, the module, the provider, the model, the token counts and the number of HTTP requests:

- **Call and request**: one row is one call, such as an AI revision of a document. A call may cause several HTTP requests – the continuation rounds at the output limit described in the previous section, and the pauses of a web search. The number of requests is therefore kept as a separate figure
- **Module**: dokumente, risiken, assets, schutzbedarf, incidents, schwachstellen, audit or test – the connection test of the AI settings is counted as well
- **Aborted calls**: a call that ends in an error has already consumed tokens. The record is therefore written in every case, not only after a successful answer
- **Counting method**: Anthropic reports input without the tokens read from the cache, while the chat completions interface of OpenAI and Azure includes them. WolfISMS converts to the Anthropic reading – otherwise the same cost would be calculated differently per provider

The evaluation is at /admin/ki_verbrauch, reachable from the AI Usage tile of the administration page and from the AI settings. The page is reserved for administrators and, like every AI feature, is unlocked in every paid edition. For the selected month or the whole period it shows calls, requests, tokens and estimated cost, broken down by module, user and model, plus the trend over the last twelve months.

Price list

Tokens only become cost with a price list. Because providers change their prices, the list is not fixed in the program but maintained on the same page – one input and one output price per model per 1 million tokens, plus the currency symbol. Providers list in US dollars; to account in euros, enter euro prices and change the symbol. WolfISMS does not convert currencies.

- **Cached input**: derived from the input price rather than asking for two more prices per model. Writing to the cache counts 1.25×, reading from it 0.10× (KI_CACHE_FAKTOR_NEU, KI_CACHE_FAKTOR_LES)
- **Names match as a prefix**: the beginning of the model name is compared, longer entries first. An entry claude-haiku-4-5 therefore also prices the dated variant claude-haiku-4-5-20251001
- **As delivered**: only models whose list price is documented are included – the Claude models as of 06/2026. For OpenAI, Azure and older Claude models nothing is entered, deliberately, rather than guessing

© 2012–2026 WolfInfoSec GmbH

Calls using a model without a price are counted in the tokens but not in the cost; every table states their number separately. That way a gap in the price list does not turn into a total that is too low. A model is removed by clearing its name and saving; the button As delivered restores the shipped list.

Retention and limits

Records older than 730 days are removed as new ones are written (KI_VERBRAUCH_AUFHEBEN in admin_ki_modul.py). The page also has a button that deletes all recorded data – recording continues afterwards, the evaluation simply starts again from zero.

> **An approximation, not an invoice**
> Server-side tools such as web research are billed on top by the provider and are not fully contained in the reported usage. For the question of which module causes the cost these figures say more than the provider dashboard; for checking the invoice, the provider’s billing remains authoritative.

12.11 Jobs in the Background

Revising a longer document takes the provider minutes. A request cannot be held open that long: Apache ends a request after 60 seconds, mod_wsgi and gunicorn have limits of their own, and in front of them there is usually a proxy with its own. The connection was dropped and the result was lost.

The route therefore does not answer with the result but with a job number: it creates the job, a thread carries on working, and the browser polls for the state every few seconds. Every single request thus stays short.

Table ki_jobs

The state is held in the database and not in process memory – with several workers the poll would otherwise reach the wrong process and not find the job. The table is created on first access; as everywhere else in this project there are no migration scripts.

Column	Meaning
id	Job number, a 32 character hex string from uuid4. No other format is accepted
bereich	Where the job comes from: dokumente, assets, schutzbedarf or audit
nutzer	User ID of the requester
status	laeuft, fertig or fehler (running, done, failed)
ergebnis	The answer of the work function as JSON (MEDIUMTEXT)
angelegt / geaendert	Timestamps for cleanup and timeout

- **KI_JOB_MAXDAUER (1800 seconds)**: how long a job may sit on “laeuft”. After that it counts as interrupted – because the worker was restarted in between, for instance. The user is told to start the editing again.
- **KI_JOB_AUFHEBEN (86400 seconds)**: older rows are deleted when the next job is created. Without that the table would grow without limit; a cron run of its own is not needed.
- **Only your own job**: the status poll compares the user ID of the row with the logged-in user and does not hand out somebody else’s job – the job number alone is not enough.

© 2012–2026 WolfInfoSec GmbH

The thread lives in the worker process.
Restarting Apache during a run discards it; the job stays on “laeuft” until KI_JOB_MAXDAUER has passed and is then reported as interrupted. Usage and cost are only booked after the call – a discarded run may nevertheless have been charged by the provider.

AI editing of a document, the processing of a Word import, the calls of all AI assistants (ki_dialog_bausteine.job_starten), the AI links and the AI translation of a document and the gap assessment assistant go this way – the latter with progress reporting (ki_job_starten(…, mit_fortschritt=True)), which ki_job_stand() passes to the browser as fortschritt. Since v4.00 no AI call waits for the answer directly past Apache any more.

12.12 Security Notes

- Since v3.99 the API key is stored AES-256-GCM encrypted in isms_settings. The key for it is derived from the application’s secret key (ISMS_SECRET_KEY or .flask_secret_key in the session directory) and therefore lives outside the database: a database dump or a backup no longer reveal the key. Up to v3.98 it was merely Base64-encoded; such values are converted on first read. The database user should still have restricted privileges
- In the admin UI the API key is only shown masked (first and last four characters); the full key can no longer be read out. If the application’s secret key is replaced (different ISMS_SECRET_KEY, deleted session directory), the stored key can no longer be decrypted: the form then shows no key and the log a warning – re-enter the key under /admin/ki_einstellungen
- Document content is transmitted to the external provider during AI requests – review data protection and confidentiality requirements before enabling the function
- **Error messages**: if a call fails, the provider’s own message is shown verbatim instead of a generic failure. An empty model name is rejected before the call, because providers only answer it with an uninformative HTTP 400

© 2012–2026 WolfInfoSec GmbH

13. Design and Colour Scheme

The colour scheme of the interface is configured at /admin/design (theme_modul.py). The menu entry is reserved for administrators.

13.1 How It Works

The interface is not styled through stylesheets: the modules assemble their HTML as f-strings and write the colours in as hex literals – roughly 2,500 of them across more than 40 files. Some of those literals also appear in Excel and PDF exports, where they are not CSS but library parameters. A search and replace across the modules would therefore be neither complete nor safe.

Instead, an after_request hook in isms.py recolours every HTML response that is delivered to the configured palette (theme_modul.response_einfaerben):

- **The mapping**: the hue is pulled towards the base hue of the theme, saturation is damped, lightness is kept. Contrast, light/dark relationships and therefore legibility stay as they were designed – only in one colour family
- **HTML only**: the hook touches text/html exclusively. Exports and downloads (xlsx, pdf, docx) are unchanged
- **Template adjusted**: in templates/template.html the link hover colour has been #0000ff instead of the keyword blue since v3.51, so that it is covered by the mapping as well
- **Fixed colours:** since v3.94 recolor_html() leaves alone whatever sits between the markers FARBE_FEST_AN/FARBE_FEST_AUS (HTML comments) or, inside <style>, between FARBE_FEST_CSS_AN/FARBE_FEST_CSS_AUS (CSS comments /*farbe-fest*/ … /*/farbe-fest*/). An HTML comment in a stylesheet would be invalid CSS and break the following rule. Protected are the risk matrix with its legend, the risk level tiles and the score gauges on the start page and in risiken.py and lieferanten.py – green to red is the message there. New elements whose colours carry meaning belong inside as well

13.2 Available Colour Schemes

Key	Name	Character
petrol	Teal / Petrol	Cool teal, a little friendlier than pure blue. Default
klassisch	Classic (colourful)	Original WolfISMS colours; no remapping at all
graustufen	Black / grey / white	Since v4.00. Plain without colour: saturation 0, signal colours become fixed shades of grey, strong colours darken towards black; header bar #1c1c1e by default

The former schemes blue/grey, graphite, navy and slate and the custom fine tuning with sliders for base hue and saturation are gone; the selection shows stripes with the remapped signal colours for every scheme. Since v4.00 there is no separate background colour for the main menu any more – the sidebar follows the scheme by itself.

13.3 Light and Dark, Header Bar and Info Tiles (isms_settings)

The settings are stored as key-value pairs in the same isms_settings table as the AI configuration. If the table does not exist, it is created automatically on first access.

© 2012–2026 WolfInfoSec GmbH

isms_settings key	Values	Meaning
theme_name	petrol | klassisch | graustufen	Active scheme; default petrol
theme_dunkel	1 | 0	Dark mode as default for everyone who has not chosen themselves
modus_u<id>	dunkel | hell	A user's own choice via the button next to the language flag (POST /design/modus); without sign-in the session applies
theme_kopf_bg	#rrggbb or empty	Background of the header bar; empty = default of the scheme (#0c519b, for black / grey / white #1c1c1e). The colour is not remapped; on a light background the text turns dark
theme_infokacheln	1 | 0	Show statistics and category tiles above the module lists; default 0 = hidden
theme_menu_bg	–	Formerly the custom menu colour; since v4.00 without effect, saving clears the value

- **Dark mode:** remaps every colour according to its role – text, border or area, recognised from the CSS property in front of the colour – so that the menu and dark buttons stay dark. Areas between the FARBE_FEST markers stay unchanged there too. Write new colours as hex with the property name in front, otherwise they count as an area
- **Info tiles:** hidden on delivery via CSS using the classes in theme_modul.INFOKACHEL_KLASSEN (18 modules). Dashboard, administration, history and PDF reports are not affected, nor are the filter tiles of the lists
- **Reset:** restores teal without a dark default, the scheme's header bar and hidden info tiles; the users' light/dark choice stays. Nothing is saved in read-only mode. With several server processes a change applies everywhere within 20 seconds

13.4 Tooltips

Since v3.97, form fields and table columns in all modules carry an ⓘ symbol with a short explanation. It appears when the mouse hovers over it, when the symbol receives focus via Tab, or when it is tapped; Esc closes it. The texts explain technical terms and the effect of a field – for example gross and residual risk, RTO, RPO and MTPD, reporting deadlines or the justification in the SoA – and exist in German and English. Buttons show their text directly, without an ⓘ, when the mouse rests on them briefly – symbol buttons and actions such as Review or Propagate O→I as well as Save or Back. The modules themselves were not changed for this.

- **Delivery:** A context processor in isms.py passes each page, via tooltips_modul.tooltips_js(), only the texts that belong to it. The script in templates/template.html finds fields by their name attribute and columns by their visible heading.
- **Saving forms only:** On fields the symbol only appears in forms with method="post". Search and filter bars stay clear.
- **Position of the symbol:** It is attached to the field's label – label for, form group, table cell before it or heading above it. If the script finds no label, it is placed after the field. There is one symbol per form and field name.
- **Buttons:** Matched by the visible label without leading or trailing symbols – a leading symbol is dropped –, symbol-only buttons by the symbol (e.g. ✏ for edit, ✕ for remove). The text appears after just under half a second of hovering, on keyboard focus or on a long press on touch devices –

© 2012–2026 WolfInfoSec GmbH

a long press does not trigger the button. An existing title attribute is shown in the same style; a catalogue text takes precedence. Only buttons in the content area count, including those a script inserts later (AI assistants).

- **Sidebar:** Every menu entry and menu group has a short text that appears to the right of the entry. It is matched by the module key from MODULE_REGISTRY (data-modul, set by isms.menu()) or gruppe:<key> from MENU_GRUPPEN. In the collapsed sidebar the module name comes first; the tooltip replaces the previous browser bubble of the icon there.

The default texts are in tooltips_katalog.py on three levels; the more specific one wins:

Level	Key	Applies to
General	*	every page, e.g. besitzer, abteilung_id, link
Module	blueprint name, e.g. risiken; pages without a blueprint: isms	all pages of the module
Page	endpoint, e.g. bcm.bcm_bia_edit	this page only – when a field name means different things within one module

A key of the form name@other only applies in forms that also contain a field other, and takes precedence over the plain name. This gives the link form (bezeichnung@link) a different text from the new-measure form (bezeichnung), and a comment field called text gets no symbol.

Switching on and off

- **Tile:** The Tooltips tile under Administration shows the state (“on” or “off”) and switches it for all users with one click (isms_settings, key tooltips_aktiv, default on).
- **Effect:** Each process caches the state and the texts for one minute. After that the change applies to everyone; no restart is needed.

For new fields the text belongs in tooltips_katalog.py, not in the module. If a field name means different things on different forms of the same module, use a page level or a name@other key.

© 2012–2026 WolfInfoSec GmbH

14. Supplier questionnaire

The supplier questionnaire collects, in writing, how a supplier implements information security. It has three parts: a question catalogue maintained by the administrator, dispatch to the supplier, and the evaluation attached to the supplier record. It is implemented in lieferanten_fragebogen.py.

Day-to-day use – sending, chasing, reading answers – is described in the user documentation. This chapter covers the administrative prerequisites.

14.1 Question catalogue

The catalogue is maintained under /admin/vendor_fragebogen. On first use WolfISMS seeds a template based on ISO/IEC 27001:2022 and NIS2 (Art. 20, 21 and 23) once: 49 questions in 12 categories, each with a framework reference and a German as well as an English wording.

Field	Meaning
Order	Position in the questionnaire; the template uses steps of ten so that questions can be inserted in between
Category	Section heading in the form (German and English)
Question	Question text (German and English)
Framework reference	Free text, e.g. “A.5.19” or “NIS2 Art. 21”
Answer type	ja_nein (Yes / Partly / No / N/A), text, langtext or datum
Mandatory	Mandatory questions must be answered in the form
Active	Only active questions go into newly sent questionnaires; those already sent remain unchanged

The template is seeded exactly once.
If the catalogue is empty and always has been (AUTO_INCREMENT is 1), WolfISMS seeds the template. If an administrator later deletes all questions, the catalogue stays empty – a deliberately emptied catalogue does not refill itself. The button in the catalogue re-seeds the template at any time, replacing the existing catalogue.

14.2 Dispatch

Questionnaires are sent from the supplier’s detail view. The supplier receives an email with a link to a public form (/lieferantenfragebogen) – no sign-in, protected by a one-time token in the URL.

Prerequisite	Details
SMTP	Configured mail dispatch (/admin/smtp); without it only the PDF or EML route is available
ISMS_BASE_URL	Determines the address in the link. Without this environment variable a relative path is produced that the recipient cannot open – the task mailer uses the same variable
External domain	Optional. If the ISMS is reachable for suppliers under a different address than for your own staff, it is stored at /admin/vendor_fragebogen (isms_settings key

© 2012–2026 WolfInfoSec GmbH

Prerequisite	Details
	vendor_fragebogen_domain) and takes precedence over ISMS_BASE_URL. Without a path WolfISMS appends the web path of the installation
Supplier data	Email address and contact person must be stored on the supplier; if either is missing, dispatch is refused with a reason
Validity	30 days from dispatch (LINK_GUELTIGKEIT_TAGE). After that the form shows the questionnaire as expired; sending again issues a new token

Besides the link there are two routes without a form: send the questionnaire as a PDF, or generate an EML file and send it from your own mail client. Answers returning this way are entered manually by a member of staff; the questionnaire then carries the source “manual” instead of “link” and does not expire.

14.3 Database

The module creates its three tables on first load – no manual schema update is required. Existing installations are extended with the fields for manual entry via ALTER TABLE.

Table	Content
vendor_frage_katalog	The question catalogue: order, category, question, framework reference, answer type, mandatory, active – each in German and English
vendor_fragebogen	A sent or recorded questionnaire: supplier, token, recipient, language, status, expiry, source, attachment
vendor_fragebogen_antwort	The answers. Question, framework reference and answer type are copied along so that an old questionnaire stays readable if the catalogue changes later

14.4 Operation and security

- **Public route:** /lieferantenfragebogen is deliberately reachable without sign-in. Access rests on the token alone; without a valid token the page shows nothing.
- **No guessing room:** The token is 64 characters from secrets and unique per questionnaire (UNIQUE index).
- **Expiry:** Expired tokens no longer accept answers. The record remains – a supplier who did not answer is a result that belongs in the supplier assessment.
- **Effect on the assessment:** When answers arrive, WolfISMS sets the supplier’s “questionnaire” field to “Yes” and recalculates the risk score.
- **Deletion:** A questionnaire can be deleted together with its answers; the activity log keeps the event.

© 2012–2026 WolfInfoSec GmbH

15. Suppliers: Import and Cloud Exit Strategy

Since v3.93 suppliers can be read in from Excel or CSV, and suppliers of cloud and SaaS types get a cloud exit strategy. Both live in lieferanten.py; the user documentation describes how to use them.

15.1 Import (/vendor/import)

- **Access**: users.vendor ≥ 2 and not read-only; other people's suppliers can only be changed where the owner rule allows it
- **Files**: .xlsx or .csv (comma or semicolon), MAX_IMPORT_BYTES = 8 MB. Column headings are recognised in both languages, the assessment criteria under their label from Risk Parameters (plus the default texts in case the workbook predates a renaming)
- **Matching**: known supplier number → that supplier; no number → the supplier with exactly that name; otherwise a new one if the option is set. Nothing is guessed: unclear rows are skipped with a reason
- **Two passes**: first a preview with "old value → new value"; writing only happens when the file is uploaded again with "Apply"
- **Read but not applied**: risk, risk score and cloud exit strategy. The import recalculates risk and score with the same function as the form and extends the score history
- **Log and limits**: every creation and change goes into the activity log; the quantity limit of the free edition applies

The Excel export has gained additional columns for this: email, the seven assessment criteria as plain text of the short rating and the cloud exit strategy as plain text ("(fehlt)" for cloud suppliers without a strategy). An export can be read back unchanged and results in "no differences". Analyses that rely on the previous number of columns need adjusting.

15.2 Cloud Exit Strategy

Column	Meaning
vendor_typ.cloud	TINYINT; 1 = the type counts as cloud/SaaS. Maintained with a tick at /admin/vendor_typen. When the column is first created, types with "Cloud" or "SaaS" in their name are preset – in the delivered data "Cloud Anbieter", but not "Softwareanbieter"
vendor.cloud_exit	MEDIUMTEXT; the strategy as TinyMCE HTML, sanitised on saving; an empty editor counts as empty

The application creates both columns itself on first use – through the interface as well as through the REST API. The list shows the number of cloud suppliers without a strategy together with a filter, the Excel export has a column of its own, and the REST API offers the fields cloud, cloud_exit and cloud_exit_fehlt (see the REST API documentation, section Suppliers).

15.3 Description and Detail View

Since v4.00 every supplier has a free-text field Description (column vendor.beschreibung), edited with the same editor as the cloud exit strategy. The application creates the column itself on first use. The field appears in the edit form and the detail view, in the Excel export and in the REST API (GET, list, POST, PATCH; section Suppliers of the REST API documentation).

© 2012–2026 WolfInfoSec GmbH

Since v4.00 the “Risk assessment” block with the seven assessment criteria only appears in the edit form; the detail view shows the description and the cloud exit strategy without it.

© 2012–2026 WolfInfoSec GmbH

16. Audit Management

The audit module (audit.py) manages audits and their findings. Its operation – creating an audit, recording findings, deriving measures – is described in the user documentation. This chapter covers the changeover from the former question catalogues, the data model, the framework-based checklist and the storage of the attachments.

16.1 Changeover from the question catalogues

Up to v3.71 the module was a standards self-check: chapter checklists for ISO 27001, NIS 2, Cyber Essentials, VdS 10000, GDPR and TISAX on the tables Audit, AuditNIS2, AuditCE, AuditVdS, AuditDSGVO and AuditTISAX. With v3.72 it has become an audit management. The checklists were dropped without replacement, and with them the report and export routes per catalogue and the audit charts.

The compliance level per framework is no longer maintained in two places: it lives in the Controls module, per requirement with applicability, implementation status and justification, condensed into the chapter overview and the conformity report.

The old tables stay in place.
Neither the application nor a migration script touches Audit, AuditNIS2, AuditCE, AuditVdS, AuditDSGVO or AuditTISAX. The former assessments remain in the database in full but are no longer read by any route. Whoever still needs them fetches them directly from the database or from a backup – nothing is deleted automatically, and whoever drops the tables does so knowingly.

16.2 Data model

Four tables, created by _ensure_tables() with CREATE TABLE IF NOT EXISTS the first time the module is opened – no migration script required.

Table	Structure and purpose
aud_audits	nummer (AUD-00001, consecutive), titel, beschreibung, datum_von / datum_bis, status (planned | confirmed | completed), art (internal | external | gap), auditor, norm, plus Ersteller/erstellt and Aenderer/geaendert
aud_abweichungen	audit_id, nummer (ABW-00001), titel, beschreibung, typ (recommendation | minor | major non-conformity), status (open | in progress | implemented | not implemented), verantwortlich (users.id), faellig, norm, norm_punkt, massnahmen, risiken (since v3.95)
aud_anhaenge	audit_id, original_name, stored_name, rel_path, mime_type, file_size, beschreibung, ersteller, erstellt – the same set of columns as mass_anhaenge so that the helpers from isms.py apply unchanged
aud_pruefpunkte	audit_id, norm_id (reference to Normen.id), norm, kapitel, nummer, titel – the last four copied from Normen –, bewertung (not assessed | fulfilled | largely fulfilled | partly fulfilled | not fulfilled | not applicable), feststellung, nachweis, beschreibung (assessment questions, from the AI or by hand), abweichung_id, sortierung

© 2012–2026 WolfInfoSec GmbH

Measures are linked, not copied.
The column massnahmen holds the IDs from the table Massnahmen as a comma list – the same pattern as in the vulnerability and incident modules. /audit/mass_neu creates a measure and links it straight away; for that the user additionally needs the mass editing permission.
Risks: since v3.95 the column risiken holds the IDs of linked risks. /audit/risiko_neu creates a risk from the finding (audit editing permission and risk ≥ 2, quantity limit of the edition) and links it together with the measures of the finding; a new measure on the finding is added to Risiko.massnahmen of every linked risk the user may change under the owner rule. The risk shows the finding via _RUECK_QUELLEN['audit'], the Excel export has the column “Risks”.

16.3 Checklists and Gap Assessments

Since v3.76 an audit can be conducted against a framework: the requirements of a framework held in the table Normen are taken into the audit as individual check points and assessed there one by one. A gap assessment is therefore not a separate type of object but an audit of the type gap – the same checklist, the same evaluation, the same findings.

The checklist copies, it does not reference.
Framework, chapter, clause number and title are written into aud_pruefpunkte when the checklist is built. An audit report has to show the wording that was assessed at the time – even where the entry in the Controls module is later changed or deleted. The column norm_id is kept as well so that the checklist can link to the current entry; that link is shown only to users with read permission for the Controls module.

The table Normen is read only in this process. An audit never changes the implementation status of a requirement: the state of the ISMS lives in the Controls module, the result of one particular audit lives in the audit. The two may diverge, and that is precisely what an audit is for.

The rating has five levels:

Rating	Conformity	Meaning
fulfilled	100 %	requirement implemented and evidenced
largely fulfilled	75 %	implemented, with minor gaps in evidence or execution
partly fulfilled	50 %	started, essential parts are missing
not fulfilled	0 %	not implemented
not applicable	none	scoping decision – the point is left out of the calculation
not assessed	none	the state before the assessment, no rating yet

The conformity level is the average across the assessed points. Points still open and those marked as not applicable are left out – otherwise a half-completed checklist would paint a wrongly bad picture. It is shown in the checklist, per chapter in the detail view of the audit, as a column in the audit overview and in both Excel exports. The levels are defined in audit.py in BEWERTUNG and BEW_PROZENT.

The routes involved:

- **/audit/pruefliste_setup**: choose the framework and the chapters and take the requirements over. Optionally only those marked as applicable (Normen.anwendbar = 1), and optionally replacing the

© 2012–2026 WolfInfoSec GmbH

existing checklist. Without replacing, only requirements that are still missing are added – ratings already given are therefore kept
- **/audit/pruefliste**: the assessment itself, with filters for chapter, rating and free text. The whole form on display is saved in one go (/audit/pruefliste_save)
- **/audit/pruefliste_reset**: clears the checklist of an audit. The findings created from it remain – they are findings of the audit and do not belong to the assessment grid
- **/audit/pruef_export**: the checklist of one audit as an Excel workbook, with the conformity level in the title row

The order of the check points is the order of the framework: sortierung takes over the id of the entry in Normen. Chapters added later therefore appear in their proper place instead of at the end of the list.

A finding can be created straight from a check point. Clause number and requirement become its designation, the finding text together with the evidence becomes the description, and the reference to the framework is carried over; the type is proposed from the rating (not fulfilled → major non-conformity, partly fulfilled → minor non-conformity, otherwise recommendation) and can be changed afterwards. Responsible person and due date stay empty, which is why the route leads straight into the form of the new finding. Check point and finding stay connected through abweichung_id: if the finding is deleted only the reference drops, and a new one can be created at the check point.

16.4 AI Assistant and GAP Assistant

The check points only carry framework, chapter, number and title of the requirement. Two aids have been added, both only for audits of type gap with a checklist and for users with audit ≥ 2 outside read-only mode.

- **AI assistant (audit_ki_modul.py)**: since v4.00 a task of the audit AI assistant (/audit/ki_assistent), which uses this core; phrases assessment questions per requirement – based on the title and, where available, the framework text from Normen.Beschreibung – and writes them without an apply step into the column aud_pruefpunkte.beschreibung, which audit.py adds by ALTER TABLE if needed. It works in blocks of 20 points (at most 500 per run) in the background through ki_jobs with progress reporting; every block is saved at once. Existing descriptions are kept unless “Overwrite” is ticked. Routes: /audit/ki_beschreibung and /audit/ki_stand, area audit
- **GAP assistant (gap_assistent.py)**: /audit/gap?audit=<id> shows the overview, /audit/gap/schritt one requirement with rating, finding and evidence. It writes to the same fields of aud_pruefpunkte as the checklist; it keeps no answers of its own
- **Progress**: table aud_gap_fortschritt (audit_id, kapitel as SHA-1 of the chapter name, status begonnen | erledigt | uebersprungen, position = check point to continue with, benutzer). Using the name instead of the position keeps the state valid when requirements are added later. “Reset progress” only empties this table
- **Missing questions**: if AI is usable, the page fetches them via /audit/ki_beschreibung for the requirement shown and the next nine empty ones in one call; existing ones are never overwritten there

16.5 Attachments and links

- **Files**: stored under UPLOAD_BASE/audit/<audit_id>/ with a UUID prefix in the file name. validate_pdf() checks the extension, the magic bytes and the size (16 MB); delivery is always as ATTACHMENT_MIME, the content type reported by the browser is never used

© 2012–2026 WolfInfoSec GmbH

- **Links**: kept in the central table Links with bereich='audit' – as for controls and measures. safe_url() defuses anything that is not an ordinary web scheme when it is displayed
- **Deleting an audit**: removes findings, the checklist, attachments (including the files on disk) and links. The linked measures remain – they belong to the measures module

16.6 Permission and menu

What counts is the column users.audit (0 = no access, 1 = read, 2 = edit); the management review carries the same permission. Menu entry, module permission and module gate come from isms_helpers.MODULE_REGISTRY (key audit) – the module can therefore be switched off at /admin/module without any data being lost.

Since v3.90 the detail view produces an audit report as PDF (/audit/<id>/pdf); its data also feeds the management review – see chapter Reports and Regulatory Reporting.

© 2012–2026 WolfInfoSec GmbH

17. Document Control and Versions

Every change to a document creates a new row in the Dokumente table; the previous state is left untouched. What that means for the users is described in the user documentation – this chapter describes what happens in the database.

17.1 Data model

Four columns hold the versions of a document together. They were added later on and are created on first access; every query addresses them by name so that their position in the table does not matter.

Column	Meaning
stamm_id	ID of the first version. Every version of the chain carries it, the first one points to itself
vorgaenger_id	The version this one grew out of
ist_aktuell	Exactly one version per chain: the one the links from the other areas point to – the most recently approved one, and while there is none, the latest
historisch	Superseded intermediate states. They are hidden from every list, evaluation and task run and are only reachable through the version list of the document

At most two versions per chain stay visible: the most recently approved one and, while it has not been approved, the latest. Everything in between is historical. The version number itself still lives in Version and Subversion – Version counts up on approval, Subversion with every save in between.

17.2 Migrating the existing data

The first time the document module is opened after the update, WolfISMS creates the four columns and assigns the existing rows to version chains. The document number is what identifies them: it was already what tied related states together before – the earlier New Version function created its record under the same number. The oldest row of a number becomes the root, rows without a number stay on their own. Comments move to the chain so that a discussion thread does not break off on the next save. A marker in isms_settings (dok_versionierung_migriert) makes sure this runs exactly once.

17.3 Links

There are two kinds of links, and they behave differently when the version changes:

- **Attached to the individual record** are attachments (dok_anhaenge), the reading obligation (dok_lesepflicht) and links (Links). They are copied along with every new version so that an old version is complete when it is opened again. The attachment file itself is not duplicated – several versions point to the same file through rel_path and stored_name; it is only deleted once no version holds it any more.
- **Attached to the version chain** are Normen.dokumente, Risiko.dokumente, Dokumente.dok_verknuepfung, prozesslandkarte.dokument_id and netzwerkzonen.dokument_id. They point to the approved version and stay there while a new one is being worked on.

On approval the references of the second group are moved to the new version in one go, the predecessors are set to status 7 and taken out of the process link. A record that already pointed to both versions keeps the reference exactly once. If a version is deleted, the remaining one moves

© 2012–2026 WolfInfoSec GmbH

up and takes over the references; if it was the last one of the chain, they are removed rather than left pointing at nothing.

Document↔document: Dokumente.dok_verknuepfung is still stored only on the document where the link was created. Since v3.94 _dok_verknuepft_ids() also reads the back references of the other side; sidebar, linking view and AI links show both directions without duplicates. A reference from a superseded version does not count, one from a draft appears as the valid version. Unlinking from the other side removes the entry on the creating document; superseded versions stay unchanged as evidence. Storing it on both sides is deliberately avoided: a draft takes over the list of its predecessor, so a link entered on the other side in the meantime would be missing after approval

Old IDs remain valid addresses.
A bookmark or a link to a superseded version still leads to exactly that version – it is the record of a past state. The detail view says so and takes you to the valid version. A superseded version is not written to: editing, linking, attachments and AI editing are blocked, including against a POST sent directly.

17.4 Effect on lists and evaluations

Every query that means the valid stock hides historical versions. That is what dok_aktuell_bedingung() in isms_helpers is for: the helper returns the WHERE condition and creates the columns if needed. Affected are the start page and the status charts, tasks and task mails, the policy portal, personnel, risks, controls, the process map, network zones, the full-text search and the document context of the AI. Queries for a single, known ID do not need the condition – a historical version is meant to stay reachable through its ID.

17.5 Volume limit, backup and import

- **The volume limit** of the free edition counts document numbers, not rows. Versions therefore use up no quota – work can continue even with ten documents.
- **Backup**: the Dokumente table grows noticeably with versioning, because every saved change is a row holding the full text. The backup covers it unchanged; this should be taken into account when sizing.
- **Data import**: on import through Data SQL export/import, stamm_id and vorgaenger_id are discarded – they point to IDs of the source instance. On the target instance the assignment is rebuilt from the document number.

17.6 Word import

Above the editing form of a document sits a box that accepts a .docx file (POST /dokumente/word_import). The conversion to HTML is done by word_import.py using python-docx; the result goes back into the form, not into the database. It is stored only by the regular save – through the same path with validation, sanitiser and log as any manual entry.

Only elements allowed by isms.ALLOWED_TAGS are produced; afterwards the same sanitiser as on save runs over the result. The preview in the editor therefore shows exactly what will end up in the column. Images go into the text as data: URLs – that is how the editor keeps them, and the Word export turns them back into image files.

© 2012–2026 WolfInfoSec GmbH

Limits

Limit	Value and place
File size	16 MB (MAX_DOCX_MB in dokumente_modul.py). MAX_CONTENT_LENGTH is no safeguard here: the value is very high because of the backup restore, so the size is measured explicitly
Single image	2 MB (BILD_MAX_BYTES in word_import.py)
All images	8 MB (BILD_BUDGET). The inflation caused by base64 counts against both limits with a factor of 4/3
Converted content	16 MB – the column Dokumente.Text is a MEDIUMTEXT
File type	Only .docx. The extension and the ZIP signature PK\x03\x04 are checked; a renamed .doc file is rejected

Permissions and locks

- **Write permission**: the import requires write permission on documents (dok ≥ 2) and is blocked for read-only accounts.
- **The same rules as for editing**: a superseded version (historisch = 1), a write-protected document and a missing owner or creator right are rejected – against a POST sent directly as well.
- **Volume limit**: when a new document is created, the volume limit of the free edition applies just as it does on every other path.

Completeness and operation

- **What stays out**: headers and footers (the letterhead comes from branding.vorlage_path), the table of contents of the Word file, text in text boxes and images in formats a browser cannot display (EMF/WMF). Skipped images are counted and reported.
- **Completeness check**: after the conversion _fehlender_anteil() compares the text of all w:t nodes with the visible text of the generated HTML. If more than 10 per cent is missing, a note appears above the form – an import that came out too short would otherwise look like a complete one.
- **python-docx** is mandatory (requirements.txt). Without it the module reports “Word import not available”; the documents page stays usable.
- **Backup**: embedded images make document texts noticeably larger. Together with versioning – every version carries the full text – this has to be taken into account when sizing the backup.

© 2012–2026 WolfInfoSec GmbH

18. Onboarding and Offboarding

Since v3.72 the personnel module (personal.py) has a second area: /personal/onboarding tracks people joining and leaving. Its operation is described in the user documentation; this chapter covers the data model and the effect on user accounts.

18.1 Data model

These tables create themselves on first access as well (_ensure_onb_tables, CREATE TABLE IF NOT EXISTS).

Table	Structure and purpose
pers_onboarding	The process: art (on | off), vorname, nachname, email, firmenemail, abteilung and abteilung_id, aufgabe, datum (joining or leaving date), user_id (the linked account from users), the three fixed items s_vertrag, s_mail and s_isms, bemerkung, abgeschlossen
pers_onb_zugaenge	The system accounts: vorgang_id, bezeichnung, benutzername (account), erledigt and erledigt_am, bemerkung, sortierung
pers_onb_equipment	The equipment – same structure; benutzername here carries the inventory or serial number

The three fixed items know three values: 0 = open, 1 = done, 2 = not required (does not count towards the progress). s_vertrag only counts for an onboarding. The item "ISMS user" has no column of its own but is derived: for an onboarding it counts as done as soon as user_id is set, for an offboarding only once users.aktiv of the linked account is 0.

18.2 User accounts

The linked account can be deactivated from within a process (users.aktiv = 0). Accounts are still created exclusively in the user management – the process merely refers to one and, when the link is made, takes over the email address of the account as long as the process carries no company email yet.

The same safeguards as in the user management.
Blocking follows the rules of /admin/toggle_user_active: your own account and the last remaining active administrator respectively master user stay active – otherwise the installation would lock itself out.

18.3 Permission and log

What counts is the column users.pers (1 = read, 2 = edit); readonly users do not write even with level 2. A completed process is locked against changes until it is reopened. Creating, changing, completing, reopening and deleting are recorded in the activities table (see the chapter Activities and Logging).

© 2012–2026 WolfInfoSec GmbH

19. Training (Personnel)

The third tab of the personnel module (since v3.93, /personal/schulungen) manages training sessions and their participants. Its use is described in the user documentation. personal.py creates both tables itself on first use (CREATE TABLE IF NOT EXISTS).

Table	Structure and purpose
pers_schulungen	The session: titel (mandatory), datum (mandatory), ort, referent, beschreibung, ersteller/erstellt_am, aenderer/geaendert_am
pers_schulung_teilnehmer	The assignment: schulung_id, user_id, besucht (0/1), besucht_am, zugewiesen_am; unique per session and user

- **Permission**: users.pers ≥ 1 to view, ≥ 2 to create, assign and record attendance; read-only users do not write
- **Assigning**: “Assign all users” only takes active accounts (users.aktiv = 1); people already assigned keep their status
- **Attendance**: besucht_am is the date of the session. Open assignments to a past session count as missed
- **Display**: the staff list (column “Open training”), the tiles and the detail page of a person read the same tables
- **Log**: changes appear in the activities as “Schulung-<id>”; every input goes through Pruefung first

© 2012–2026 WolfInfoSec GmbH

20. Change Management

The Change Management module (changemanagement_modul.py, /change) controls changes to systems and configurations according to ISO/IEC 27001 A.8.32: request, approval by the board, implementation, post-implementation review. It is available in every edition and can be switched off under /admin/module (registry key change).

20.1 Permission

- **Permission assets**: the module has no column of its own in users. A change is always a change to assets or configurations – whoever may read the asset register also reads the changes (1), whoever maintains it maintains them (2).
- **Owner**: editing, status changes, board, post-implementation review and deletion require write permission and ownership – with several owners, any one of them suffices; master users may always. As with incidents, the creator does not count automatically.
- **Approvers**: whoever is named on the board casts their vote even with read permission only. Portal-only accounts may do none of this.

20.2 Data Model

The module creates its three tables on first call; no migration script is needed.

Table	Content
chm_changes	The change request: number CHG-NNNNN, title, description, justification, type, category, status, priority, risk and impact of the change, planned window and downtime, implementation, test and rollback plan, owner and implementer, a timestamp per step and the fields of the post-implementation review (pir_*). Links to assets, risks, measures, incidents and documents are held as a comma list of IDs in the record
chm_freigaben	The board: one row per approver with stage (1–3), role, vote (offen, zugestimmt, abgelehnt – pending, approved, rejected), comment and time. A user appears only once per change
chm_historie	Every status change and every vote: who, when, from which status to which, with a remark

20.3 Types and Workflow

Type	Approval
Standard	Pre-approved routine change – no board, approved on submission
Normal / Major	Board required; without a named approver the change cannot be submitted
Emergency	May be implemented before approval, straight from the draft; the board votes retroactively. A later no does not reset the change but stands as an objection in the history

Step	From	To
Submit	Draft	Submitted (Standard: Approved)
Vote	Submitted	Approved once everybody has approved;

© 2012–2026 WolfInfoSec GmbH

Step	From	To
		Rejected on the first rejection
Start implementation	Approved (Emergency also Draft, Submitted)	Implementing
Implementation done	Implementing	Implemented
Roll back	Implementing, Implemented	Rolled back
Close	Implemented, Rolled back	Closed – only with a recorded post-implementation review
Withdraw	Draft, Submitted	Withdrawn
Revise	Rejected, Withdrawn	Draft – all votes are reset to pending

- **Status only through steps**: the status is changed exclusively via /change/status, never through the edit form. Which step is offered depends on the current status.
- **Stages**: a stage only votes once the stage below has approved completely. The owner maintains the board under /change/cab.
- **Post-implementation review (PIR)**: with “Implementation done” or “Roll back” it falls due 14 days later (PIR_FRIST_TAGE). Date, result (successful, with restrictions, failed …) and a remark are recorded. Without it a change cannot be closed – this is the evidence an audit wants to see.
- **Delete**: removes the board and the history of the change as well.

20.4 Tasks and Calendar

Three dates of a change appear in /aufgaben and in the reminder email: the planned window with the implementer (or every owner), a pending vote whose stage is due with the approver – due at the latest when the window starts – and the due post-implementation review with every owner. aufgaben_mailer.py deliberately keeps its own copy of the query so that it works without the UI modules. The change calendar (/change/kalender) shows every change with a planned start in a monthly grid.

© 2012–2026 WolfInfoSec GmbH

21. Projects

Since v4.00 the project module (projekte.py, /projekte) has an ISMS frame according to ISO/IEC 27001 A.5.8 “Information security in project management”: every project carries classification, security responsibility, security requirements, risk assessment, checkpoints along the way and an orderly hand-over with lessons learned. Frame and traffic light live in projekte_isms.py. The module shares the permission mass and belongs to the Advanced edition and to Enterprise.

Property	Value
New columns in Projekte	auftraggeber, sicherheitsverantwortung (user IDs), klassifizierung, umfang, ablage, sicherheitsanforderungen, risikobewertung (+ _datum), ergebnis, uebergabe, lessons_learned, restrisiko (+ _datum), risiken, assets, dokumente, lieferanten – by ALTER TABLE on first call; “Goal and benefit” stays in Bezeichnung (with _en)
projekt_rollen	projekt_id, rolle, user_id or person (free text for externals), reihenfolge; suggested client, project lead, security responsibility
projekt_gates	projekt_id, name, termin, kriterium, stand (open, passed, passed with condition, failed), auflage, nachweis, reihenfolge; suggestion G0 initiation to G3 hand-over & closure
Phases	planning (open, in planning), implementation (approved, in implementation), closure (implemented, archived) – the traffic light asks for different sections per phase
ISMS traffic light	pruefpunkte(): what is missing in the current phase – a hint only, no lock
Permissions	read with mass ≥ 1, write with mass ≥ 2 and not read-only; changes by owners or master

© 2012–2026 WolfInfoSec GmbH

22. Business Continuity Management

The BCM module (bcm_modul.py, /bcm, permission bcm) holds the business impact analysis (BIA), the contingency plans and the BCM exercises. Since v3.84 all three are attached to an asset. The former separate level “critical processes” has been dropped: the recovery objectives belong to the asset, and a parallel process list in BCM had to be maintained twice. Business processes are kept as primary assets.

22.1 Data Model

Table / column	Meaning
Assets.rto_stunden / rpo_stunden / maz_stunden	Recovery objectives in whole hours, NULL = not defined. Fixed steps 1, 4, 8, 24, 72, 168, 720; RPO additionally 0 (no data loss). The RTO must not exceed the MTPD
bcm_bia.asset_id	The asset assessed
bcm_bia.nummer	BIA number in the format BIA-NNNNN
bcm_notfallplaene.asset_id, bcm_uebungen.asset_id	The asset the plan applies to or that was exercised
bcm_uebungen.massnahmen	since v3.95: linked measures as a comma list of IDs like aud_abweichungen.massnahmen. Maintained via /bcm/uebung/massnahmen (bcm ≥ 2), a new measure additionally with mass ≥ 2; the measure shows the exercise via _RUECK_QUELLEN['bcm']
bcm_prozesse, prozess_id	Legacy data – stays readable but is no longer maintained

assets_modul._ensure_spalten() creates the columns on the asset, the _ensure_* helpers in bcm_modul.py those on the BCM tables – both idempotent and without a migration script. The BCM module calls the asset check itself, because it may be opened before the asset module.

22.2 Taking Over Legacy Data

- **Once per server process**: _ensure_en_columns_bia() adds asset_id and nummer, makes prozess_id nullable and assigns missing BIA numbers in id order. migration_nummern.py knows the prefix BIA as well.
- **No automatic assignment**: no asset is guessed from a former process. A BIA without an asset shows “formerly process: …”; users with edit permission see above the list how many are still to be assigned.
- **Record RTO/RPO/MTPD again**: the values of the former processes are not taken over – they belong to the respective asset and are entered there.
- **Old addresses**: /bcm/prozess, /bcm/prozess/neu and /bcm/prozess/edit redirect to the BIA overview; /bcm opens the BIA directly.
- **Do not delete bcm_prozesse**: the table holds the names that keep legacy data traceable.

22.3 Evaluation and Interfaces

- **BIA score**: mean of all 20 ratings (four dimensions financial, operational, regulatory, reputation across the time steps 1 h, 4 h, 24 h, 72 h, 1 week; scale 1–4). From 3.5 critical, from 2.5 high, from 1.5 medium.

© 2012–2026 WolfInfoSec GmbH

- **Primary assets without a BIA**: are counted as a key figure. Secondary assets inherit their requirements from the primary ones and need no BIA of their own.
- **Asset export and import**: the workbook carries RTO, RPO and MTPD as columns of their own (“4h”, “3d”); the import reads the same notation and checks RTO ≤ MTPD.
- **REST API**: the assets area returns and accepts rto_stunden, rpo_stunden and maz_stunden, filters by rto_max/rpo_max/maz_max and sorts by them – see the document “WolfISMS REST API”.

© 2012–2026 WolfInfoSec GmbH

23. Playbooks

The playbook module (playbooks_modul.py, /playbooks) was rebuilt with v4.00 – following the frame of ISO/IEC 27035 and NIST SP 800-61, ISO/IEC 27001 A.5.24–A.5.28, NIS 2 Art. 23 and GDPR Art. 33/34. A playbook describes how to respond to a scenario (ransomware, phishing, failure of a service provider, reportable incident). Data model, traffic light and display components live in playbooks_isms.py; the AI assistant is described in the section AI Assistants of the Other Modules. The module belongs to the Advanced edition and to Enterprise.

Property	Value
Routes	/playbooks[?pbdetail=<id>] (list or detail), /playbooks/bearbeiten[?id=] (new and edit), /playbooks/link?id=&view= (links), /playbooks/zuordnen (assign a file or link), /playbooks/loeschen (POST); /playbooks/ansicht redirects to the detail view
Tables	playbook_texte (profile and sections, extended by ALTER TABLE), playbook_schritte, playbook_rollen, playbook_tests – created by the module on first call. Files stay in playbooks (new playbook_id), links in Links (bereich playbooks, bereich_id); play_upload and play_attachment remain in isms.py
Profile	nummer (suggested PB-001), art (incident response, emergency and recovery, reporting, operations), description as scenario, schweregrad (from low/medium/high/critical), besitzer, stellvertretung, freigabestand (draft, in review, approved, under revision), version, freigabe_datum, freigegeben_von, naechste_pruefung, archiviert
Sections	zweck, ausloeser, erstbewertung, eindaemmung, beseitigung, wiederherstellung, kommunikation, nachbereitung (MEDIUMTEXT, each with a hint); the former free text inhalt remains as “Further content”
Approval	on the change to “approved” freigabe_werte() sets date and approver, the next review to twelve months (if empty or past) and the version to 1.0 (if empty)
Traffic light	pruefpunkte(): responsible person and deputy, type and scenario, all eight sections, at least three procedure steps, each with a role, roles staffed, approved, next review, test within the last twelve months. Red for an overdue review or a failed last test – a hint only, no lock
Links	ID lists in playbook_texte: assets, risiken, incidents, dokumente, lieferanten, behoerden, dienstleister (contacts), massnahmen
Permissions	read with play ≥ 1 or admin, write with play ≥ 2 and not read-only; an existing playbook is changed and deleted by the responsible person, the deputy, the creator (as long as nobody is responsible) or a master
Delete	removes steps, roles and tests; assigned files and links remain and appear unassigned in the list again
Input	sections are cleaned like documents (sanitize); isms_helpers.Pruefung checks required fields and lengths before the SQL

© 2012–2026 WolfInfoSec GmbH

24. Further Data Model Changes (v4.00)

The modules create the following changes themselves on first call; no intervention is needed. They are listed here so that a look into the database makes clear where the columns come from.

Area	Change
Legal register	fixed set of ten fields (number, legal basis, category, requirement, applicability, implementation with linked measures/documents/controls, responsible, status, open items with deadline, last/next review). New columns include fundstelle, anwendbarkeit (+ _en), massnahmen_ids, dokumente_ids, normen_ids, verantwortlich_id, frist, naechste_pruefung. The old columns relevanz, bereiche, verantwortlicher, dokumentenreferenz and faelligkeit are taken over exactly once by the process that creates the column anwendbarkeit; afterwards they are no longer written
Related controls	Normen.verwandte_normen (comma-separated IDs), always maintained on both sides (normen_mapping.py); detail, link view and coverage per framework show them
ISO/IEC 27001 versions	report, SoA, “Save SoA as document” and the status chart work with every installed version (framework ISO/IEC 27001:*), by default with the newest; ?fassung= selects another. If several are installed, a switch in the heading changes between them. The package normen_ISO27001_2024.tar.gz contains DIN EN ISO/IEC 27001:2024-01 with numbers and titles, without the text of the standard
Measures ↔ assets	Massnahmen.assets (ID list); an asset shows its measures directly and via risks, vulnerabilities, changes and incidents
Operational plan	status per open instance open → in progress → done; only “done” archives the instance as evidence and creates the next date – for badly overdue tasks counted from today
Vulnerabilities	swm_fristen – remediation deadlines per severity, see Risk Parameters
Supplier assessment	“Previous security incident at the supplier?” was weighted the wrong way round, and the tier did not affect the score. Both are corrected once per installation, then level and score of all suppliers are recalculated; weights you maintained yourself stay untouched
AI dialogues	ki_dialog holds one row per user and assistant (area), ki_jobs the background jobs of all assistants

© 2012–2026 WolfInfoSec GmbH

25. Data Protection

Since v4.01 WolfISMS has a data protection module (datenschutz_modul.py, menu item “Data Protection” in the group Governance & Compliance, /datenschutz). It keeps the record of processing activities under Art. 30 GDPR, the threshold analysis and the data protection impact assessment (DPIA) under Art. 35, and the details on processing on behalf under Art. 28 in the supplier file. Data model and rules live in datenschutz_isms.py, the DPIA in datenschutz_dsfa.py, the DPA block in datenschutz_avv.py. The module is one of the core modules and available in every edition, including the free one.

25.1 Permission

New module permission datenschutz (column users.datenschutz, 0 = no access, 1 = read, 2 = edit, abbreviation DSC in the permission matrix and the Excel export). isms_helpers.kartenrechte_sicherstellen() creates the column with 0 – also for existing users; there is no take-over as with the process map and network zones. The first user of a new installation gets level 2. Administrators may always read and edit. As in the other modules, owners and master users may change and delete an entry; only administrators maintain the DPIA list.

25.2 Data Model

Table	Content
ds_vvt	one row per processing activity: nummer (VVT-…), typ (verantwortlicher | auftragsverarbeiter), titel, beschreibung, zweck, rechtsgrundlagen (keys 6a…9j separated by “|”), interesse, betroffene, datenkategorien, besondere, retention (loesch_zahl, loesch_einheit, loesch_ausloeser), joint controllership (+ partner), abteilung_id; as processor auftraggeber_id (customer contact), kategorien_av, avv_datum, avv_dokument; links assets, controls, massnahmen, unterauftrag as ID lists; status, review; threshold analysis swa_muss, swa_kriterien, swa_ergebnis, swa_begruendung, swa_von, swa_am; Besitzer, Ersteller, Aenderer
ds_vvt_empfaenger	recipients per processing activity: art (intern, lieferant, behoerde, sonstige), lieferant_id or abteilung_id or name, land (ISO code), garantie (adequacy decision, Data Privacy Framework, standard contractual clauses, BCR, Art. 49, none). A country outside the EU and the EEA counts as a third country
ds_mussliste	DPIA list under Art. 35(4): nr, text, beispiel, quelle, aktiv. Filled with the 17 entries of the DSK list (version 1.1 of 17 October 2018) when created, maintained by the installation itself afterwards
ds_dsfa	one row per DPIA: nummer (DSFA-…), vvt_id, titel, status, the sections beschreibung, notwendigkeit, abhilfe, stellungnahme, standpunkt (MEDIUMTEXT), ergebnis (tragbar | konsultation), ergebnis_text, massnahmen, controls, stellungnahme_von/_am, abgeschlossen_am, review
ds_dsfa_risiken	risks from the data subjects’ point of view: quelle, schaden, w and s (1–4 each), rest_w, rest_s, massnahmen, risiko_id (entry in the risk register after the hand-over)
ds_avv	DPA details per supplier (lieferant_id = vendor.id): avv, datum, dokument_id, drittland, land, garantie, tia_dokument, bemerkung
ds_unterauftrag	sub-processors per supplier: name, leistung, land, ul_lieferant_id (if it is in the supplier register itself)

© 2012–2026 WolfInfoSec GmbH

The modules create all tables themselves on first call (once per server process); no intervention is needed. They are part of the normal backup.

25.3 Routes

Route	Purpose
/datenschutz	list as controller; ?typ=auftragsverarbeiter, ?typ=dsfa switch, ?vvtdetail=<id> shows an entry
/datenschutz/edit	create and edit (?id=, ?typ=)
/datenschutz/swa	threshold analysis of a processing activity (?id=)
/datenschutz/status, /empfaenger, /link, /delete/<id>	set status, recipients, links, delete (POST)
/datenschutz/dsfa…	DPIA: new (from a processing activity as controller), detail, edit, risiko, status, link, delete
/datenschutz/avv	save the DPA block of the supplier file (POST)
/datenschutz/mussliste	maintain the DPIA list – add entries, switch them off and on; administrators only
/datenschutz/export	Excel export per record (?typ=)
/datenschutz/ki_assistent, /ki_tom_download	AI assistant and download of the TOM annex

25.4 Rules

- **Status of a processing activity:** draft → under review → approved. As controller an entry can only be approved once purpose, at least one legal basis and a threshold analysis with result and justification are in place; as processor it needs a client. If no review date is set on approval, it is set to today + 12 months
- **Threshold analysis:** regel_ergebnis() suggests “DPIA required” as soon as an entry of the DPIA list applies or at least two of the nine criteria of WP 248 rev. 01 are met. A person sets the result; if it differs from the suggestion, the page shows that
- **Hints:** data categories under Art. 9/10 without a legal basis from Art. 9 and recipients in a third country without a safeguard are flagged
- **Status of a DPIA:** draft → in progress → DPO opinion → completed. Score = likelihood × severity (1–4 each, following DSK short paper no. 18), always calculated. The residual risk only counts implemented measures. Completing requires the opinion, a result and residual values per risk; anyone with datenschutz = 2 completes the opinion
- **Review:** if the processing activity changes after the DPIA has been completed, the DPIA shows that it has to be reviewed (Art. 35(11))
- **Hand-overs:** a DPIA risk goes to the risk register via /risiko/ki_assistent?ausgang=dsfa:<id> (risiko_id then points there), the measures assistant creates mitigating measures with origin “DPIA” and links them
- **DPA block:** appears in the detail view of a supplier as soon as the criterion “personal data” is Yes or Partly or DPA data already exist. Anyone who sees the supplier file may read it; it is maintained with datenschutz = 2, not with the supplier permission – so the data protection officer maintains the DPA without being allowed to change the other supplier data

© 2012–2026 WolfInfoSec GmbH

25.5 AI Assistant for Data Protection

datenschutz_ki_assistent.py (tasks in datenschutz_ki_neu.py), area datenschutz in ki_dialog and in the usage report. Permission: datenschutz ≥ 2 or administrator, not read-only, plus “AI usage”. Tasks: create a processing activity, record as processor, gaps from the assets (assets with personal data without a processing activity – on request only), review an entry, suggest a threshold analysis, draft a DPIA, TOM annex under Art. 32(1) from controls and measures as a Word file.

No legal advice
Legal basis, retention period and result of the threshold analysis remain suggestions. The assistant never sets “approved”, never completes a DPIA, never writes the DPO’s opinion and creates neither assets nor suppliers.

© 2012–2026 WolfInfoSec GmbH

26. Reports and Regulatory Reporting

v3.90 added several reports, exports and the reporting assistant for security incidents. This chapter describes structure, tables and prerequisites; operation is covered in the user documentation, chapters Reports and Incidents.

26.1 Reports Page (/berichtsuebersicht)

berichte_modul.py provides the page. It is registered in isms_helpers.MODULE_REGISTRY with the key berichte and recht None and therefore has a menu entry for all users with write access; portal-only users do not see it. Every card belongs to a module of the registry and appears only if modul_aktiv() is true for that module and the user holds its module permission. The selection of individual audits, management reviews and exercises reads the most recent entries directly from aud_audits, mgmt_review and bcm_uebungen; if a table is missing, only the selection is left out. The former overview /berichte in the Controls module redirects here; its individual reports (?berdetail=1/2/3) remain.

26.2 PDF Reports

Report	Address	File
Audit report	/audit/<id>/pdf, short version ?pruefliste=0	audit.py
BCM exercise report	/bcm/uebung/<id>/pdf	bcm_modul.py
KPI report	/ziele/bericht?von=YYYY-MM-DD&bis=YYYY-MM-DD	ziele_modul.py
Awareness report	/personal/awareness	personal.py

All four are produced via isms_helpers.pdf_briefpapier() and pdf_erzeugen(): HTML is rendered with WeasyPrint, a letterhead configured in the settings (isms_settings pdf_briefpapier_path and the four margin values pdf_briefpapier_margin_*) is laid under every page, and emoji are removed. If only the letterhead fails, the report is delivered without it. WeasyPrint must be installed on the server – the same prerequisite as for the management review. An audit report export is recorded in the activity log.

26.3 Excel Exports

Address	Sheets	Permission
/schwachstellen/export	Patch status, Vulnerabilities	schwachstellen
/change/export	Key figures, Changes	assets
/bcm/export	Coverage, BIA, Contingency plans, Exercises	bcm
/ziele/export?von=…&bis=…	Overview, Values	ziele

They are built with isms_helpers.xlsx_blatt() and xlsx_antwort() (openpyxl): title row in the module colour, frozen header rows, auto filter, dates as real date cells. Exports from module lists take over their filter parameters. The older exports (incidents, assets …) are unchanged.

© 2012–2026 WolfInfoSec GmbH

26.4 Reporting Assistant (meldungen_modul.py)

The blueprint meldungen belongs to the registry entry incidents and is switched on and off together with the incident module. Which frameworks apply is still determined by the columns DSGVO, NIS2 and CRA of the table Incidents; Incidents itself is unchanged. Two new tables create themselves on first access:

Column	Type	Meaning
incident_id, regelwerk	INT, VARCHAR(10)	Incident and framework (NIS2, DSGVO, CRA); unique per pair
kenntnis	DATETIME	Time of awareness from which the deadlines run
entscheidung	VARCHAR(30)	offen, meldepflichtig or nicht meldepflichtig
begruendung	TEXT	Mandatory for “nicht meldepflichtig”
Aenderer, geaendert	INT, DATETIME	last change

Table inc_meldepflicht – the assessment per incident and framework.

Column	Type	Meaning
incident_id, regelwerk, stufe	INT, VARCHAR	Incident, framework and reporting stage (fruehwarnung, meldung, zwischen, abschluss, betroffene)
gemeldet_am	DATETIME	Time of the report
stelle, aktenzeichen	VARCHAR(200), VARCHAR(100)	Recipient and reference of the authority
inhalt	MEDIUMTEXT	Text submitted
Ersteller, erstellt	INT, DATETIME	who recorded the report

Table inc_meldungen – one row per report submitted.

- **Deadlines:** hours from awareness (24 h or 72 h); the final report counts one calendar month from the actual 72-hour notification, or from its deadline if none was recorded. Without an assessment, midnight of the detection day applies. The stages are defined in REGELWERKE at the top of the file.
- **Done:** a stage as soon as a report of that stage is recorded; if it lies after the deadline, it counts as late.
- **Updating:** recording a report sets Meldungwann, Meldungwem, Änderer and geändert of the incident so that list, export and REST API agree; an open assessment becomes meldepflichtig.
- **Overview:** offene_fristen() returns all open mandatory stages, archived incidents excluded – for the list /incidents/meldefristen, the bar on the incidents start page and the reports page.
- **Tasks:** since v3.94 aufgaben_meldefristen(tage) returns the open and overdue mandatory stages whose deadline ends within the coming days – one item per owner of the incident (main owner and objekt_besitzer), without an owner only in the “All” view of master users. /aufgaben shows them in a section of their own with a tile if the incident module is active and users.incidents ≥ 1
- **Permissions:** read with users.incidents ≥ 1; assessing, recording and deleting require write access and besitzer_check() on the incident. Every action is logged in the activity log under the incident number.

© 2012–2026 WolfInfoSec GmbH

26.5 Read Confirmation Log (dok_lesebestaetigung)

users.dok_read only holds which documents a user has confirmed. Since v3.90 a confirmation in the policy portal also writes a row (user_id, dok_id, gelesen_am) to the table dok_lesebestaetigung; the awareness report reads the most recent confirmation from it (isms_helpers.lesebestaetigungen_map()). Earlier confirmations have no entry and appear without a date. Resetting the read status deliberately does not delete the log – it is the history, the current state remains dok_read. A failure while logging never prevents the confirmation.

26.6 Audit Data in the Management Review

audit.mr_auswertung(von, bis) supplies the management review with two lists: the audits whose period overlaps the review period (without an end date the start counts), and the nonconformities – major and minor findings that are either open or come from an audit in the period. Recommendations only count as a number with the audit. Like the other live sections, the data is recalculated on every opening and export. If the audit module is switched off or not licensed, both sections are left out and the free texts remain.

Backup and SQL export
The backup covers inc_meldepflicht, inc_meldungen and dok_lesebestaetigung like any other table. The per-area SQL export (chapter Data SQL Export/Import) does not transfer them.

© 2012–2026 WolfInfoSec GmbH

27. Risk Parameters

Administrators configure the automatic risk calculation at /admin/risiko_parameter. The module manages vendor weights (supplier assessment) and risk level thresholds (all risk types).

27.1 Vendor Weights

For every supplier, questions are evaluated in two groups. Each answer (Yes / Partly / No) yields points. The average per group produces the value 1–5 for likelihood and impact respectively.

Group	Default question	Yes	Partly	No
Likelihood	Certification in place? (ISO 27001, SOC 2 etc.)	0	2	5
Likelihood	Security questionnaire completed?	0	2	5
Likelihood	Contractual agreement / DPA in place?	0	2	5
Likelihood	Previous security incident at the supplier?	5	2	0
Impact	Processing of personal data (GDPR)?	5	2	0
Impact	High protection requirement of the systems / data?	5	2	0
Impact	Tier criticality (1 = uncritical, 3 = critical)	2	4	5
Impact	Legal / compliance requirements violated?	5	2	0

The table vendor_gewichte is created on first access to /admin/risiko_parameter if it does not exist yet. Changes take effect immediately on newly calculated supplier risks. Every recalculation is also appended to vendor_historie, so the development of a supplier risk can be traced.

Weight of the tier: Since v4.01 the tier counts double in the impact – (points of the other three impact questions + 2 × tier points) / 5, rounded – and sets a floor of tier points − 1: mission-critical or tier 1 at least 4, business-critical at least 3, tier 2 at least 2. The tier points come from the label in vendor_tier, not from the row tier criticality: a number in the name applies directly (tier 1 = 5, tier 2 = 3, tier 3 = 0), labels without a number count by their position (with four levels 0, 2, 4, 5). Before, a mission-critical supplier without personal data almost always ended up “low”. The same rule applies in the REST API and the risk history; level and score of all suppliers are recalculated once per installation (flag vendor_bewertung_2026_11_tier in isms_settings).

27.2 Risk Level Thresholds

Risk value = likelihood × impact (each 1–5, total 1–25). The thresholds apply to all risk types.

Risk level	Default threshold (< X)	Meaning
low	5	Risk value < 5 → low
medium	10	Risk value < 10 → medium
high	20	Risk value < 20 → high
critical	≥ 20	Automatically critical (no separate threshold)

© 2012–2026 WolfInfoSec GmbH

27.3 Remediation Deadlines for Vulnerabilities

Since v4.00 the section “Deadlines” sets how many days a vulnerability may stay open per severity (ISO/IEC 27001 A.8.8, NIS 2 Art. 21(2)(e)). The days are stored in the table swm_fristen (schwachstellen_fristen.py). Due date = date recorded + deadline of the current severity; it is never stored but always calculated afresh – a changed deadline or score takes effect at once. An empty field means “no deadline”; remediated, accepted and rejected vulnerabilities are never due.

Severity	CVSS	Default
Critical	9.0 – 10.0	7 days
High	7.0 – 8.9	30 days
Medium	4.0 – 6.9	90 days
Low	0.1 – 3.9	180 days
Informational	0.0	no deadline
No score	–	30 days

List and detail view of the vulnerabilities show the due date, the tile “Overdue” counts the exceeded ones; the vulnerability assistant uses the same calculation.

© 2012–2026 WolfInfoSec GmbH

28. Risk Score History and Cron Times

WolfISMS records the trend of three average scores daily (historie_modul.py / risiko_historie.py): ⌀ open risks, ⌀ all risks, ⌀ active suppliers. This chapter covers the recording and the control of the background jobs; the evaluation at /risiko/historie is described in the user documentation.

28.1 Snapshot Mechanism

- ensure_today(db) checks on every call to /risiko/historie whether a snapshot already exists for the current day and creates one if needed
- On the page itself a snapshot can be forced at any time via action=snapshot (POST /risiko/historie)

28.2 Cron Time Management (/admin/cron_zeiten)

admin_cron_zeiten.py manages time, interval and active state of the daily background jobs in the table cron_settings. Currently wired up: the task reminder email (aufgaben_mailer.py) and the risk score snapshot (risiko_historie.py).

- Selectable interval: every day, every 2/3/14/30 days, weekly
- Using "Run now" each job can be started immediately, regardless of time, interval and active state

Important prerequisite:
For a time stored at /admin/cron_zeiten to actually take effect, the server crontab must call the scripts more frequently (recommended: every 5 minutes), e.g.
*/5 * * * * /usr/bin/python3 /var/www/isms/aufgaben_mailer.py
Each run checks for itself whether the configured time has been reached and performs the actual work at most once per day (or per interval). This schedule does not control the backup – see the chapter Data Backup.

28.3 Task Reminders and Read Confirmations

The task mailer sends the reminder for due tasks. Besides measures (for each of their owners), reviews and document workflows, the recorded tasks also include outstanding read confirmations: training and policy documents a user has to read according to the reading obligation matrix and has not yet confirmed as read in the employee portal (users.dok_read). Which document types know a read confirmation at all is defined by the positive list LESE_TYPEN in portal_modul.py; who has to read which of them is decided by the matrix dok_lesepflicht × user_rollen – see the chapter Roles and Reading Obligation. A document without a role assignment and a user without a role trigger no reminder.

Reporting deadlines: since v3.94 the mailer also includes the reporting deadlines of incidents, with the usual lead time VORLAUF_TAGE = 14 and redirection to deputies. It imports meldungen_modul for this instead of rebuilding the deadline calculation; if the import fails, a warning goes to the log and the rest is sent as usual. The mail goes out once a day – a 24-hour deadline can expire before the first reminder; the page remains authoritative.

Since v3.48 the readers are additionally notified the moment a document is released, instead of waiting for the next run of the task mailer (dokumente_modul._notify_lesepflicht_freigabe):

- **Trigger**: the release of a document whose type appears in LESE_TYPEN. All other types are merely displayed in the portal and send no mail
- **Recipients**: the users of the roles that have to read the document according to the reading obligation matrix – provided they are active, have an email address and have not unsubscribed

© 2012–2026 WolfInfoSec GmbH

from mail (users.aktiv, users.no_mail). If those columns are missing in an older installation, the respective filter is skipped. If no role is assigned, no mail goes out; the document module points this out to whoever released the document

- **Language**: per recipient according to users.lang; the mail carries number, name, version, who released it and a direct link into the employee portal
- **Delivery**: all mails go out over a single SMTP connection (aufgaben_mailer.send_email_bulk). A failure for one recipient does not abort delivery to the rest
- **Fault tolerance**: a missing or disabled SMTP configuration is logged but never passed back to the request – the release itself works without mail delivery too

Since v3.90 every confirmation is also logged with a timestamp in dok_lesebestaetigung – see chapter Reports and Regulatory Reporting.

© 2012–2026 WolfInfoSec GmbH

29. Diagnostic and Maintenance Scripts

Besides the web blueprints, the installation directory contains a number of standalone command-line scripts for diagnostics and one-off migrations. They are executed manually with sudo python3 <script>.py in the application directory and are not reachable through the web server.

- **check_config.py**: shows all values resolved by config.py (including the env file used) so they can be compared with the previous configuration before a change; dbpass is printed masked only
- **debug_mailer.py**: checks whether secret.env can be found and read, and compares the database credentials loaded from it with the environment variables that are set – useful when the task mailer fails to start
- **diag_lese.py <email>**: shows, for a single user, the exact read status value (dok_read) and the associated training documents, exactly as aufgaben_mailer.py computes it
- **migration_nummern.py**: normalises existing object numbers (documents, risks, measures, assets, suppliers, projects, incidents, BIA) chronologically to the format PREFIX-NNNNN; the default is a dry run, only --commit writes to the database
- **read_version.py [file]**: reads the Version variable from isms.py (or a specified file), e.g. for scripts or CI/CD pipelines. If the .py sources have been removed after compilation, the version is read from the adjacent .pyc without executing the module

Note: these scripts are maintenance tools for administrators with server access, not part of the web interface. With --commit migration_nummern.py permanently changes historical object numbers – run it only after a backup and in agreement with the users.

© 2012–2026 WolfInfoSec GmbH

30. Activities and Logging

WolfISMS logs all security-relevant user actions in the database table activities and login events in a separate log file.

30.1 Activity Table (Database)

Column	Content
id	Primary key
datumuhrzeit	Timestamp of the action
user	User ID
nummer	Record number / ID of the affected object
bereich	Module area (e.g. risks, measures, assets …)
was	Description of the action (created, changed, deleted)

30.2 Further Logs

Log	Location	Content
Login log	/var/www/isms/login.log	Successful and failed sign-ins in the format YYYY-MM-DD HH:MM:SS – [message]

Reviewing the login log is recommended with: tail -f /var/www/isms/login.log

With v4.00 the logs of the SQL and REST consoles and of the software update are gone. The tables sql_console_log and rest_console_log stay in existing databases with their content, but no route writes or reads them any more – whoever no longer needs them drops them knowingly with DROP TABLE. An existing update_log.txt comes from the former software update.

© 2012–2026 WolfInfoSec GmbH

31. Security Notes

31.1 Access Control

- The admin area is protected by the flag admin=1 in the database
- All routes use @login_required and additionally check the permission they require
- **Deleting controls:** since v3.95 administrators only – the delete route in normen.py rejects everyone else, even for requests sent directly; the ¨ button in the chapter list and detail view only appears for them. A whole framework was already an admin matter
- CSRF protection is active for all POST forms (Flask-WTF CSRFProtect); the five REST API blueprints are explicitly CSRF-exempt and authenticate by API key instead. The public questionnaire form keeps CSRF protection and is additionally guarded by its one-time token
- **Writing links via POST only:** since v3.93 every route that changes data through an ordinary link is listed in _NUR_POST (isms.py) – linking and unlinking in seven modules, the supplier actions, deleting contacts, the read confirmation in the policy portal and the treatment plan. A GET to them is rejected. The links in the modules stay; a global script in template.html sends them as POST with the CSRF token, including links added later. They are recognised by their path: since v3.95 only the path part of WEBPATH is put before the route – in Docker installations WEBPATH is a full URL, so until then no entry matched there and every link was rejected as GET. Consequences: opened in a new tab, such an action does not run; with an expired licence they are blocked for normal users just like forms. New modules register writing links there
- Rate limiting: 2000 req/day and 500 req/hour globally, HTTP 429 on exceeding the limit. The REST API is exempt (limiter.exempt) so that sync clients are not throttled – there the API key is the only protection
- Passwords are hashed with Werkzeug – never stored in clear text
- API keys are stored only as SHA-256 hashes – the clear text cannot be recovered after creation
- Blocked user accounts (aktiv=0) cannot sign in, neither by password nor via SSO

31.2 Uploads and Attachments

- **Request limit**: MAX_CONTENT_LENGTH is 512 MB. The value is deliberately high because restoring a backup accepts a complete archive of SQL dump and upload directory – it is therefore not a safeguard for the attachment routes
- **Attachment limit**: individual attachments are subject to the much tighter limit of 16 MB, enforced by validate_pdf()
- **File type check**: attachments must be genuine PDF files: the extension, the magic bytes (%PDF) and the file name are all validated
- **Content type on download**: the content type reported by the client is neither stored nor used for the download. Attachments are always served as application/pdf so that an HTML file disguised as a PDF cannot execute in the origin of the application
- **Word import**: the .docx file is not stored; it is converted in memory and discarded again. The extension, the ZIP signature and a size limit of its own of 16 MB are checked; the generated HTML passes through the same sanitiser as any entry made in the editor

31.3 Input and Output Safety

- Rich-text fields are sanitised server-side with bleach
- **Data in script blocks:** values that end up in a <script> block go through js_daten() (isms_helpers.py), which escapes <, > and & – json.dumps alone lets a </script> through. URL parameters that mean an ID (such as /bpmn?id=, /risiko/mass_neu?riskdetail=) are only accepted as a number

© 2012–2026 WolfInfoSec GmbH

- Free text that ends up in HTML attributes or text areas is escaped consistently – this applies in particular to the parameters of the document workflow routes (/pruefen, /zupruefen, /freigabe), the change log and the user names in selection lists. The helper user_label() renders user names consistently as “surname, first name” everywhere – in the same order the selection lists are sorted by
- Stored links are validated by the helper safe_url() (isms_helpers.py) against a scheme allow list so that javascript: URLs cannot be executed; url_param() escapes values that go into URL parameters
- The login reports unknown users and wrong passwords with the same message so that valid accounts cannot be enumerated; the distinction is preserved in the activity log

31.4 Input validation before the database

Up to 3.51 a form value that did not fit its column only surfaced at INSERT time: MySQL reported “Data too long for column” or “Incorrect date value”, the user saw an error page and their input was lost. Every write path now checks values against the schema beforehand – centrally in isms_helpers.py (class Pruefung, function spalten_laenge).

- The limits come from information_schema rather than from constants kept in the code: a changed column takes effect without a code change
- VARCHAR is counted in characters, TEXT in bytes – an umlaut takes two bytes in utf8mb4, an emoji four
- The message names the label from the form, not the column name, and collects all objections of a form instead of stopping at the first error
- The form attributes maxlength and type="date" remain in place but are only a suggestion to the browser – a directly submitted POST bypasses them
- Attachment file names are shortened to the column width (the extension is kept) so that an INSERT cannot fail after the file is already on disk
- Comments are limited to 4000 characters; overlong texts are rejected with a notice instead of being silently truncated

31.5 Configuration Security

- The database password is not in config.py but in /etc/wolfisms/secret.env
- secret.env: chmod 640 and chown root:www-data – no web access possible
- Flask secret key: ISMS_SECRET_KEY from secret.env or, if empty, the file .flask_secret_key created in the session directory on first start (0600). It signs sessions and CSRF tokens and, since v3.99, protects the AI API key – changing it signs out all users and makes the stored AI key unreadable
- **ISMS_PROXY_HOPS:** Number of reverse proxies in front of the application (default 1, set in the Docker image). Only that many hops of the headers X-Forwarded-For/-Proto/-Host/-Port are evaluated – client IP for the rate limit, HTTPS detection, host name for redirects and external links. If the application is attached directly to the web server (Apache with mod_wsgi), ISMS_PROXY_HOPS=0 belongs in secret.env; otherwise any client can set the headers itself and forge its IP or the host name. Up to v3.98 Proto and Host were always on and For was not evaluated at all: behind a proxy the rate limit only saw the proxy’s IP, and all users shared one login quota
- config.py contains no fallback value for DB_PASSWORD – if the variable is missing the app will not start
- Never commit secret.env to Git repositories (add it to .gitignore)
- SSO credentials (MS_*, GOOGLE_*) also belong in secret.env – not in isms.conf
- Flask debug mode is off by default and must not remain enabled in production

© 2012–2026 WolfInfoSec GmbH

31.6 Authentication and sessions

- Sessions are stored server-side in the file system (Flask-Session, type: filesystem). Up to 3.51 this was configured but never switched on – the call to Session(app) was missing, so Flask kept using the signed client cookie. Session contents were therefore readable in the browser and could not be revoked server-side. Since 3.52 the call is in place; at startup the application logs a warning should the cookie interface be active again
- Session cookies: Secure=True, HttpOnly=True, SameSite=Lax, lifetime = 1 day (SESSION_PERMANENT=True)
- **Remember me:** since v3.93 the remember cookie also carries Secure, HttpOnly and SameSite=Lax. Anyone signed in with “remember me” before keeps the old cookie until it expires or they sign in again
- Max. 500 concurrent sessions (SESSION_FILE_THRESHOLD)
- The session directory is set to 0700 at startup and the session files to 0600 – only the account the application runs under may read them (SESSION_DIR, default /var/www/isms/wolfisms/sessions)
- If compromised: delete all session files (rm -rf /var/www/isms/wolfisms/sessions/*)
- Rate limit storage: the default memory:// counts per WSGI process – with several workers every limit is multiplied accordingly. For such installations set RATELIMIT_STORAGE_URI to shared storage (e.g. redis://…). In the container the application warns at start-up as long as memory:// applies – the image starts three gunicorn workers
- **Account lockout:** Since v3.99 WolfISMS locks an account for 15 minutes after ten wrong passwords – counted in the database, independent of the IP limit (section Account lockout after failed attempts). For an unknown email address a dummy hash is checked so that the response time does not reveal whether an account exists
- When a limit is hit, WolfISMS answers to suit the caller: JSON clients get the API error shape including Retry-After, browsers a message that matches the limit that was exceeded – no longer a blanket “too many login attempts”

31.7 Delivery: Images, Compression and Headers

- **Images at their own addresses**: since v3.93 the logo, the flags and the remaining images come from /bild/<file>?v=<modification time> and /branding-bild/… instead of being embedded as base64 in every page. The browser keeps them for a year; when a file changes, so does its address. Both routes work without signing in (the login page shows the logo), are exempt from the rate limit and only serve image formats – ../ paths and other files end in 404
- **Line icons:** Since v4.00 menu and button symbols are no longer image files but SVG masks in the stylesheet (ui_icons.py). They take the text colour, in dark mode too, and need no request of their own
- **Compression**: HTML and JSON are sent gzip-compressed – important in Docker, where gunicorn serves without a front-end server. Switch it off with the environment variable ISMS_GZIP=0, for instance if a reverse proxy already compresses
- **Headers**: Cache-Control: no-store for HTML and JSON as a real HTTP header, Strict-Transport-Security (HTTPS only) and Cross-Origin-Opener-Policy: same-origin
- **Templates**: correct meta tags, <html lang> following the user’s language, alt texts on all images

31.8 Recommended Hardening Measures

- Install a valid licence file – prevents demo mode and lifts the quantity limits
- Enforce HTTPS (configure SSL/TLS in the Apache vhost) – session cookies are already Secure=True

© 2012–2026 WolfInfoSec GmbH

- Schedule regular backups (at least daily via cron job) and test the restore annually
- Monitor the login log regularly for failed attempts and HTTP 429 entries
- Create at least a second admin account (emergency access)
- Grant master rights very sparingly – they allow approvals, the view of all tasks and role maintenance
- Disable user accounts that are no longer needed and delete unused API keys
- Use a separate API key for every external application
- Grant the database user only the privileges it needs (least privilege)

© 2012–2026 WolfInfoSec GmbH

32. Administrator Checklist

32.1 Routine Maintenance

Interval	Prio	Task
Daily	H	Create a backup and store it off-site
Daily	H	Review the login log for failed attempts
Weekly	M	Check for new WolfISMS versions and install them (Docker: new image); JavaScript libraries at /admin/updates
Weekly	M	Review user accounts and permissions; disable accounts no longer needed
Monthly	L	Evaluate the activity log
Monthly	L	Check the licence expiry date
Quarterly	M	Recertify the access matrix – review permissions that are due, in particular privileged ones (levels A and N)
Annually	H	Test a backup restore
Annually	H	Perform and document the management review per ISO 27001 clause 9.3

© 2012–2026 WolfInfoSec GmbH

33. Support and Contact

WolfISMS – Wolf's Information Security Management System
© 2012–2026 WolfInfoSec GmbH
Update server: https://update.wolfisms.de
Version: 4.01 | As of: November 2026

Revision History of This Document

Revision	Version	Main changes
Rev. 4	v2.13	REST API, API key management, licence-derived secret key, Azure AD environment variables, session cookie flags, rate limiting
Rev. 5	v2.30	Comment function, cron time management, SQL console, risk score history, role/threat/vulnerability catalogues, branding, direct installation of the standards catalogue, container mode, corrected description of backup and restore
Rev. 6	v3.28	Objectives module, Operational Plan module, document SQL export/import, corrected blueprint and database column overview, new permission codes ZIE and LNK
Rev. 7	v3.47	New chapters on the management review, access matrix, REST console and single sign-on; data SQL export/import replaces the document-only export; departments and teams as master data; role maintenance moved into the role catalogue; extended incidents API (GET list with filters); hardened backup restore; debug mode off by default; reworked security chapter (upload limits, attachment MIME type, XSS and enumeration protection); updated configuration, package and permission overviews
Rev. 8	v3.51	Module chapters 19–23 reduced to the administrative facts – how the modules are used is described in the user documentation; new chapters on the AI assistants for risks and incidents and on design and colour scheme; AI consistently tied to the Enterprise licence (ENT); provider list extended by Azure AI Foundry (Copilot) with notes on base URL and API version; automatic forwarding to Microsoft with break-glass access; notification of the readers when a document is released; updated blueprint and admin menu overviews
Rev. 9	v3.63	Five REST API areas instead of incidents only, with synchronisation state and without a rate limit; new chapter on the supplier questionnaire; forgotten-password self-service; lockout after failed TOTP attempts and server-side sessions actually enabled; central input validation before the database; user names shown consistently as “surname, first name”
Rev. 10	v3.68	Editions and feature scope in the licence management chapter (access matrix, management review and operational plan from Enterprise upwards); new chapter on module management – modules can be switched on and off at /admin/module and ordered in the menu; new vulnerability module with its own permission column (SWM) and a sixth REST API area; three chapter references corrected
Rev. 11	v3.70	New chapter Company Data and Structure Maps (company data, org chart, process map, network zones); editions rewritten – no module depends on the edition any more, AI features from Standard (STD); SSO can be set up in the interface at /admin/sso; Excel export and import of departments
Rev. 12	v3.71	New chapter Roles and Reading Obligation: role management at /admin/rollen, multiple assignment through user_rollen, reading obligation matrix of the policy portal (dok_lesepflicht); task reminders and the release mail follow the matrix
Rev. 13	v3.72	New chapter Audit Management – the audit module replaces the question catalogues with audits and findings (aud_audits, aud_abweichungen,

Revision	Version	Main changes
		aud_anhaenge), the old tables stay untouched; new chapter Onboarding and Offboarding (pers_onboarding); AI answer length per model with continuation rounds; external domain of the questionnaire link
Rev. 14	v3.75	New section Volume Limits of the Free Edition in the licence management chapter: DEMO is limited to ten assets, risks, documents and suppliers each, unlimited from Standard upwards; counting method, the checked write paths including REST API and import, and the usage display at /admin/lizenz_upload
Rev. 15	v3.76	New section Checklists and Gap Assessments in the audit management chapter: an audit takes the requirements of a standard as check points (aud_pruefpunkte), rates them on five levels and derives the conformity level from them; gap assessment as a new type of audit; finding created from a check point
Rev. 16	v3.77	Edition model recast: the REST API (including API keys and the REST console), single sign-on and the access matrix belong to the Enterprise edition, while all modules stay part of every edition; Enterprise without a user cap, Standard 5 and Professional 20 users with write access; new section User Limits of the Editions with the counting method and the three checked write paths
Rev. 17	v3.78	New section Usage and Cost in the AI settings chapter: WolfISMS records the token consumption of every AI call from the providers' own answers (ki_verbrauch), because the working key yields no account balance; evaluation by module, user and model at /admin/ki_verbrauch, a maintained price list per 1 million tokens, retention 730 days
Rev. 18	v3.79	New chapter Document Control and Versions: every change to a document creates a version of its own (stamm_id, vorgaenger_id, ist_aktuell, historisch), and the existing rows are grouped into version chains by document number once; attachments, reading obligation and links travel with the version, while standards, risks, the process map, network zones and document links are moved over on approval
Rev. 19	v3.80	New section Jobs in the Background in the AI settings chapter: long AI runs are kept as a job in the ki_jobs table, because the web server does not hold the request open that long. New section Word import in the chapter Document Control and Versions: conversion of a .docx file into the document text including limits, permission checks and the completeness check
Rev. 20	v3.84	New chapters Change Management (change requests, approval board in up to three stages, post-implementation review) and Business Continuity Management – BIA, contingency plans and exercises are now attached to assets, RTO/RPO/MTPD are held on the asset, the level critical processes has been dropped. New sections Absence and Deputy, Ticket System Link, Plans & Registers and Addenda per AI Request
Rev. 21	v3.85	Edition model recast: six tiers including the Plus editions STD+ and PRO+. Eight modules now belong to Professional (projects, operational plan, audit, management review, vulnerabilities, legal register, playbooks, BCM), while the access matrix, single sign-on and the REST API belong to the Plus tiers and to Enterprise – not to Standard and not to Professional. The field tier therefore also accepts an enumeration of permitted tiers (tier_ok_an), and the edition is checked centrally in modul_gate(); tier_required of the REST API is now a list
Rev. 22	v3.87	The Standards module is now called Frameworks throughout the interface: menu entry, framework import, framework reference in audits and the supplier questionnaire, and the permission overview are renamed. The table Normen, the permission norm and the address /normen are unchanged.
Rev. 23	v3.89	Multiple owners per object: new sub-chapter on the table objekt_besitzer (main owner, permissions, tasks, document versions, REST API, SQL export); REST API PATCH fields, change management permission and dates, and the task mailer updated.
Rev. 24	v3.90	Reports and regulatory reporting: new chapter on the reports page, PDF reports

© 2012–2026 WolfInfoSec GmbH

Revision	Version	Main changes
		(audit, BCM exercise, KPIs, awareness), Excel exports (vulnerabilities, changes, BCM, KPIs), the reporting assistant for NIS-2, GDPR and CRA (inc_meldepflicht, inc_meldungen), the read confirmation log (dok_lesebestaetigung) and audit data in the management review.
Rev. 25	v3.91	Menu groups: the sidebar arranges the modules in six groups (MENU_GRUPPEN), the order set at /admin/module applies within a group; the browser remembers collapsed groups, the group of the current page stays open; own menu images for six modules. The administration page arranges its tiles in six groups (ADMIN_GRUPPEN).
Rev. 26	v3.92	AI assistants for assets (AI risk analysis creates or links risks, AI protection needs suggests CIAA), AI assistant and GAP assistant for gap assessments, disclaimer in all AI dialogs, overview of the AI assistants; module categories maintained centrally at /admin/kategorien (stamm_kategorie); database connection with DB_TYPE, DB_PORT and TLS (isms_db.py).
Rev. 27	v3.93	Area Frameworks renamed to Controls; SoA as a menu entry of its own when ISO/IEC 27001 is installed; supplier import (/vendor/import) and cloud exit strategy (vendor_typ.cloud, vendor.cloud_exit); training in the personnel module (pers_schulungen); AI provider nele.ai; writing links via POST only, js_daten(), image routes, gzip (ISMS_GZIP) and response headers.
Rev. 28	v3.94	Sidebar of the detail views with five entries at most per section (sidebar_kuerzen) and back references from other modules (rueck_abschnitte_html); document links in both directions; AI links for documents (dokument_ki_modul.py); reporting deadlines in tasks and the task email; risk matrix, tiles and gauges keep the traffic-light colours in the colour scheme (FARBE_FEST).
Rev. 29	v3.95	AI translation for documents (dokument_uebersetzung_modul.py), risk assistant with description, business impact and opportunities; risks from audit findings (aud_abweichungen.risiken), measures on BCM exercises (bcm_uebungen.massnahmen); remark per protection goal (bem_*), criticality by the maximum principle; action bars without linking buttons; deleting controls admin only; _NUR_POST with WEBPATH as a full URL.
Rev. 30	v3.96	New licence model: Essentials (ESS) and Advanced (ADV) for up to 5, 10 or 25 users with write access, each also with “+” for single sign-on and the REST API, plus Enterprise (ENT). Objectives, personnel, contacts, reports and the access matrix now belong to Advanced; Free has the same modules as Essentials; AI in every paid edition. Section Editions and Feature Scope rewritten, user limits, module management, REST API, REST console, SSO and AI adjusted.
Rev. 31	v3.97	Tooltips: form fields and table columns in all modules carry an ⓘ with an explanation (German and English); buttons and sidebar entries show their text on hover. New section Tooltips with how it works and the levels; new tile in the group Appearance & Templates for switching on and off.
Rev. 32	v3.99	Security: AI API key stored AES-256-GCM encrypted instead of Base64 (key derived from the application secret key, legacy values converted on first read); secret key random and persistent instead of licence-derived (correction); ISMS_PROXY_HOPS controls ProxyFix, X-Forwarded-For counts for the per-client rate limit; start-up warning for memory:// limiter in the container; account lockout after ten wrong passwords (new section 2.6); SSO client secrets encrypted as well; restore resets the schema markers.
Rev. 33	v4.00	New interface (line icons, list components with live search and no paging, title tiles, symbol tiles and compact sidebar, way back to the administration, version in the footer); SQL and REST consoles and the software update removed; AI sign-in via Entra ID; AI assistants for risks and assets as a consultation on a page of their own (replacing the risk dialog, AI risk analysis and AI protection needs); design rewritten for teal, classic and black/grey/white with light/dark per user and switchable info tiles; supplier description; new chapter Playbooks (editor); import package NIS2 standard risks; addenda per AI request removed; objectives, process map and network zones in the main menu with their own

© 2012–2026 WolfInfoSec GmbH

Revision	Version	Main changes
		user permissions; contacts under Operations & Resilience; AI assistants of all modules (suppliers, documents, measures, vulnerabilities, changes, incidents, projects, controls, legal register, audits, playbooks); playbook module described anew according to ISO/IEC 27035; new chapters Projects (ISO/IEC 27001 A.5.8) and Further Data Model Changes; remediation deadlines for vulnerabilities; weight “previous security incident” corrected; ISO/IEC 27001 in several versions; copyright WolfInfoSec GmbH.
Rev. 34	v4.01	New chapter Data Protection (record of processing activities as controller and as processor, threshold analysis with DPIA list, data protection impact assessment with risk table, DPA block in the supplier file, AI assistant for data protection); user permission “Data protection”; tier weighted more strongly in the supplier assessment.

© 2012–2026 WolfInfoSec GmbH