FOCUS FAMILY

Interfaces handbook

Setting up and operating REST/JSON, webhooks, CMIS, XÖV, SCIM, calendar and SIEM

Version 1.3.0 · November 2026 · Software Notion

Contents

1. **About this handbook**
   1. Principles
2. **Overview of the interfaces**
   1. Addresses at a glance
   2. Terms
3. **Permissions and roles**
4. **Approval workflow (four-eyes principle)**
   1. States
   2. Procedure
   3. What triggers a new approval
   4. Switching mandatory approval off
5. **Privacy defaults**
   1. Clear names and pseudonymisation
   2. Areas
   3. Disclosure log and access requests
6. **Network and security of outbound connections**
7. **REST/JSON API**
   1. Creating an API client
   2. Requesting a token
   3. Scopes
   4. Resources
   5. Conventions
   6. Change feed
8. **Webhooks**
   1. Setting up
   2. Events
   3. Message format
   4. Verifying the signature
   5. Delivery behaviour
9. **Inbound direction**
   1. Delivering documents
   2. Changing case master data
   3. Inbox and review
   4. Withdrawing
10. **CMIS: filing in the DMS or e-file**
    1. Setting up
    2. Filing structure
    3. Properties
    4. Connection test
    5. Inbox folder (inbound)
11. **Focus Family CMIS repository**
    1. Authentication and permissions
    2. Structure
    3. Queries and limits
    4. Filing documents
12. **XÖV and xdomea**
    1. What is transferred
    2. Transports
    3. Directory
    4. SFTP
    5. XTA 2
    6. FIT-Connect
13. **User provisioning with SCIM 2.0**
    1. Setting up
    2. Behaviour
    3. Provider notes
14. **Calendar subscriptions (iCalendar)**

15. **SIEM forwarding (syslog)**
16. **Manual handover from the case view**
17. **Exporting and importing the configuration**
18. **Operation**
    1. How delivery works
    2. Transfer log and retries
    3. Pausing and emergency stop
    4. Monitoring figures
    5. Audit events
    6. Configuration parameters
19. **Troubleshooting**
    1. REST API error codes
20. **Glossary**
21. **Commissioning checklist**

About this handbook

This handbook is for the **technical administration at the provider** and for technical service partners who connect the Focus Family server to case management systems, document management, e-files, youth welfare offices, identity providers, calendars and security monitoring. For every interface it covers setup, reference, privacy defaults, operation and troubleshooting.

Installing, hardening and running the server itself is described in the **Technical handbook**, day-to-day work in the panel in the **Admin panel handbook**. Download the machine-readable description of the REST API (OpenAPI 3.1) in the panel under **Interfaces → Interface documentation**.

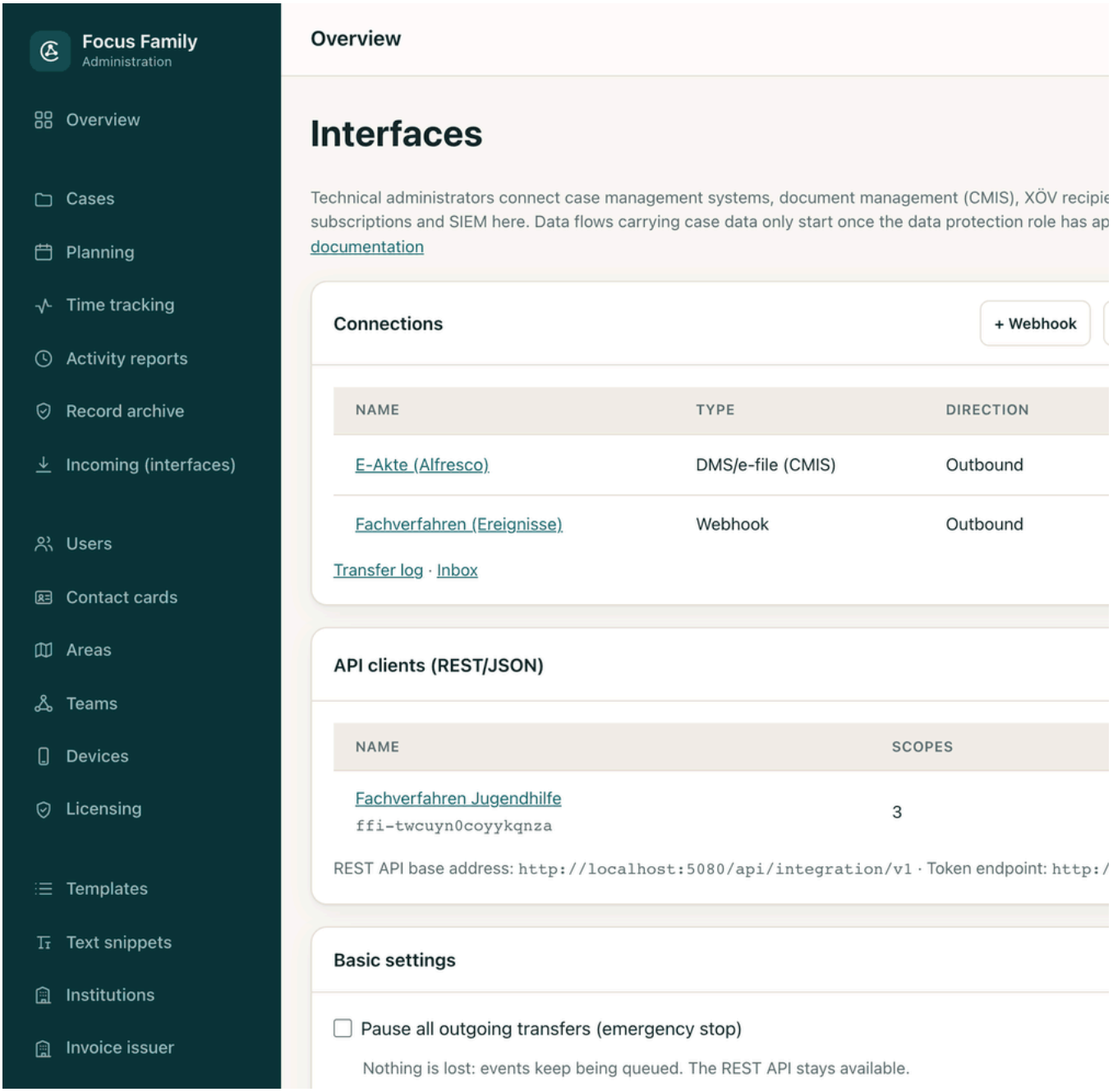
Assumed knowledge: HTTP and TLS, OAuth 2.0 (client credentials flow), JSON and XML, the basics of the respective target system (DMS, gateway, identity provider, SIEM).

Principles

- **Everything runs in the provider's server.** Apps and desktop are unchanged and fully usable offline. The interfaces only see data that has already been synchronised to the server.
- **Nothing is active by itself.** Every connection and every API client starts as a draft. Data flows carrying case data only start after approval by the data protection role (four-eyes principle).
- **Pseudonymised and minimal is the default.** Clear names, drafts, password-protected documents and confidential document types only leave the server after an explicit setting and approval.
- **Bidirectional only by configuration.** Every inbound direction is off until it is set up and approved. There are no inbound deletions.
- **Standards instead of proprietary formats:** OAuth 2.0, OpenAPI 3.1, RFC 9457, CloudEvents 1.0, Standard Webhooks, CMIS 1.1, xdomea 3.1.0 and 4.0.0, XTA 2, FIT-Connect, SCIM 2.0, iCalendar (RFC 5545), syslog (RFC 5424/5425) and CEF.
- **Private and local data stay invisible.** No interface reaches private cases and entries, cases of departed practitioners with private ownership, or deleted, deletion-pending and processing-restricted records.

Overview of the interfaces

Interface	Direction	Purpose	Approval needed
REST/JSON API	third-party system calls Focus Family (read and, if set up, write)	case management, statistics, accounting, own tools	yes, as soon as a scope with case data is chosen
Webhooks	outbound	notification about events, identifiers only	yes
CMIS (DMS/e-file)	outbound, optionally inbound	file final documents, activity reports and closed case files in the DMS; take over documents from an inbox folder	yes
CMIS repository	third-party system calls Focus Family	DMS or e-file reads the case files via CMIS 1.1 browser binding and files documents	via the API client
XÖV (xdomea)	outbound, optionally inbound	case file transfer on closure, import of incoming transfers with import confirmation; transport via directory, SFTP, XTA 2 or FIT-Connect	yes
SCIM 2.0	identity provider calls Focus Family	create, update, deactivate practitioner accounts; map groups to teams	no (no case data)
Calendar subscriptions	calendar program calls Focus Family	subscribe to own appointments in the work calendar	no; only the data protection role can choose more than "Busy"
SIEM (syslog)	outbound	audit chain events to security monitoring	no (identifiers only)



Addresses at a glance

Address	Content
`https://<server>/api/integration/v1`	base address of the REST API
`https://<server>/connect/token`	token endpoint for API clients
`https://<server>/cmis/browser`	CMIS repository (browser binding), repository ID `focusfamily`
`https://<server>/scim/v2`	SCIM base address for the identity provider
`https://<server>/feeds/calendar/<token>.ics`	personal calendar subscription
`/admin/integrations`	panel area **Interfaces**
`/admin/integrations/log`	**Transfer log**
`/admin/integrations/inbox`	**Incoming (interfaces)** for professional review
`/admin/integrations/docs`	handbook and OpenAPI description for download
`/admin/calendar-feed`	create or end the own **Calendar subscription**

The **Interfaces** page shows the REST API base address and the token endpoint of your installation for copying.

Terms

- A **connection** is a target system the server contacts itself: webhook, CMIS, XÖV or SIEM. It has a direction (outbound, inbound, both directions), triggers, privacy settings and its own delivery position.
- An **API client** is a third-party system that signs in to the REST API or the CMIS repository with its own credentials. What it sees is determined by scopes, areas, IP list and privacy settings.
- **Tokens** for SCIM and calendar subscriptions are long random values that are shown only once and stored on the server only as a SHA-256 hash.

Permissions and roles

Permission in the panel	Effect
Manage interfaces (REST clients, CMIS, XÖV, webhooks, SCIM, SIEM) (`IntegrationManage`)	area **Interfaces**: create, test, submit, pause and delete API clients and connections; renew secrets; SCIM tokens; transfer log with retry; export and import the configuration. Included in system admin.
Data protection officer (`DataProtection`)	approve or reject data flows carrying case data, switch the policy **Activate data flows with case data only after approval (four-eyes principle)**, allow calendar detail beyond "Busy", take over wider privacy settings when importing.
Review and accept interface inbox items (`IntegrationInboxReview`)	page **Incoming (interfaces)**: review and reject incoming items within the own administrative scope; assigning to a case and accepting additionally require “Edit cases”. Included in the professional admin; “Edit cases” alone is not enough because the inbox can also contain other practitioners' data.
Edit cases (`CaseEdit`)	button **Hand over to target system** in the case view.
View security audit (`AuditView`)	events `integration.*` in the audit view
Configure monitoring (`MonitoringConfigure`)	access to `/metrics` with the figures `fokusfamilie_integration_*`

All accounts holding **Manage interfaces** or **Data protection officer** can read the pages of the **Interfaces** area. Accounts with the data protection permission only find the menu item **Interfaces** in their section and can review and approve there, but cannot create or change anything.

Sensitive actions require **re-authentication** (password and, if set up, TOTP, or passkey): saving an API client, renewing a client secret, saving a connection, renewing a signing secret, generating a SCIM token, importing the configuration, approving or rejecting, and changing the approval policy. Approvals require a **local account**, not an SSO account.

Approval workflow (four-eyes principle)

States

State in the panel	Meaning
Draft	created or rejected; nothing is transferred, the API client gets no token
Awaiting approval	submitted; events are not transferred until the data protection role decides
Active	data flow runs
Paused	stopped by the technical administration or automatically after repeated failures; nothing is lost
Revoked	API client permanently withdrawn; its credentials are deleted

Procedure

1. The technical administration creates the API client or connection. Purpose of the transfer and legal basis (free text, e.g. "§ 61 ff. SGB VIII, § 69 SGB X") belong to it; both appear in the approval dialog, in the privacy log and in the list of recipients.
2. For connections: run **Test connection** (see the sections of the individual interfaces).
3. **Submit for approval.** If the interface carries no case data or mandatory approval is off, the button reads **Activate** and the state changes to **Active** immediately.
4. **Another** person with the data protection permission checks recipient, purpose, legal basis, scope of data, scopes or triggers, privacy settings and areas and chooses **Approve** or **Reject** (with a reason). Whoever created the interface or changed it last (creating and every save) cannot approve it; the status card shows these persons a four-eyes notice. Example: A creates, P approves, B (technical administration and data protection) changes the endpoint; the interface is pending approval again, and neither B nor A can approve it, another person with the data protection permission can. The decision only applies to the reviewed state: the form carries a server-computed revision (SHA-256 over the approval-relevant settings and the time of the last change). If the interface was changed after the page was opened, the server refuses the decision (`itg_err_changed`), and the interface stays pending approval. Approving, editing and submitting lock the record in one transaction, so a concurrent change never activates an unreviewed state.
5. Approval writes the privacy entry `integration.recipient_approved` with name, kind, direction, purpose and legal basis.

No case data and therefore no approval is involved for: SIEM connections, API clients with only `integration:billing.read`, `integration:catalogs.read`, `integration:catalogs.write`, `integration:directory.read` and `integration:inbox.read`, SCIM and calendar subscriptions.

What triggers a new approval

If the technical administration changes the scope of data of an active, paused or waiting interface carrying case data while mandatory approval is on, the state returns to **Awaiting approval**:

- API client: scopes, areas, IP list, privacy settings, inbound direction, purpose, legal basis.

- Connection: direction, address, triggers, kind-specific settings (except the polling interval), privacy settings, inbound direction, areas, purpose, legal basis.

Token lifetime, rate limit, expiry date and secrets of API clients, and authentication, credentials, certificates, pins and the switch for unencrypted transport of connections can be changed without new approval.

Nothing is transferred until approved again. The delivery position is kept; after approval the connection catches up on the events since its last position.

Switching mandatory approval off

Under **Data protection policy for interfaces** an account with the data protection permission switches off **Activate data flows with case data only after approval (four-eyes principle)**. The technical administration then activates data flows without a second review; only the data protection role can then widen privacy settings. This is only meant for providers without their own data protection role in the panel; the change is audited as `integration.settings_changed` with `approval_off`.

Privacy defaults

Every connection and every API client has a **Data protection** card. Deviations from the default are possible but trigger approval. The data protection role may always widen (clear names, drafts, password-protected documents, unblocking sections or document types); the technical administration only while mandatory approval is on, because re-approval follows. With mandatory approval off, the server does not take over a widening by an account without the data protection permission; the previous values stay without a message.

Setting	Default	Effect
Transfer clear names	off	Unticked, the target receives case alias, file number and pseudonymised titles and file names. Ticked, social data resides in the target system and is subject to its permissions and retention periods (observe § 65 SGB VIII).
Also transfer drafts	off	default: only final or signed documents
Also transfer password-protected documents	off	The target can neither read nor index these files without the password.
Blocked case file sections	Notes	Documents in ticked sections never leave the server via this interface.
Blocked document types	progress note, conversation note, phone note, free text	Content with special confidentiality stays on the server unless unblocked one by one.
Folder template in the target (CMIS)	`/{year}/{fileNumber}/{section}`	placeholders see *CMIS*

Clear names and pseudonymisation

Without clear names (and without scope `integration:sensitive`):

- Name and contact of the youth welfare office caseworker appear as `[pseudonymisiert]`.
- Participants are called "Kind" (child) or "Person" with a short identifier; given name, family name and date of birth are missing.
- Document titles read `<document type> <date> v<version>`, file names accordingly; the author is masked.
- Activity descriptions of time entries are missing.
- Identifiers, file number, case alias, status, help type, periods, approval and responsible youth welfare office stay unchanged.

The directory resources (`integration:directory.read`) always return names and work e-mail addresses of staff in clear text, but never permissions or devices.

Areas

With the area mode switched on, the **Areas** card limits an interface to cases of practitioners in these areas. Without a selection the interface applies organisation-wide.

Disclosure log and access requests

- Every approval creates a privacy entry (`integration.recipient_approved`).
- The access package under Art. 15 GDPR lists the recipients per case: connections of all kinds including webhooks that received data of the case (first transfer, number of transfers, redeliveries counted separately) and approved API clients with scopes for case data as potential recipients.
- For connections the basis is the disclosure register: every successful delivery concerning a case or record records time, identifiers, event type and the connection's name, kind, purpose and legal basis at that time. The entries contain no case content, remain when the connection is deleted and when a case is finally erased (identifiers only), and do not change if the connection is later renamed or its purpose or legal basis changed.
- Deletions in Focus Family never delete automatically in the target system. With the trigger **Erasure of a case is requested** a webhook recipient receives the event `de.fokusfamilie.case.erasure_requested` without content; deletion in the target is the provider's responsibility.
- The final deletion of a case or of individual records in Focus Family also erases the copies held by the interfaces: inbox entries keep only technical data (ID, state, source, idempotency key) and the reason `erased`, while content, file name, case reference, external ID and file link are removed; external references are deleted; the delivery register keeps the object ID in the target system but not the storage path, which may contain file numbers or names. Because the idempotency information is kept, a repeated delivery does not recreate the deleted data.

Network and security of outbound connections

Webhooks, CMIS, XÖV (SFTP, XTA 2, FIT-Connect), OAuth token requests and SIEM use one shared, hardened client.

- **Address check at connect time:** the server resolves the host name and checks **every** address right before connecting. Blocked are loopback, link-local (169.254.0.0/16, fe80::/10, which includes cloud metadata services), 0.0.0.0/8, multicast, broadcast and known metadata addresses (100.100.100.200, fd00:ec2::/32). Internal networks (10/8, 172.16/12, 192.168/16) are allowed because DMS and gateways usually live in the intranet.
- **No redirects:** HTTP redirects are not followed; a 3xx counts as an error.
- **HTTPS is mandatory.** Unencrypted HTTP only with the switch **Allow unencrypted HTTP (test systems only)**, syslog over TCP only with **Allow unencrypted TCP (internal network only)**. This also applies to the **Token endpoint** (OAuth 2.0): when saving it must be an absolute `https://` address without user information, otherwise the panel reports `itg_err_https` or `itg_err_endpoint`; it may live on a different host than the connection's address (identity provider). In addition, the server refuses every outgoing request (webhooks, CMIS, XTA, FIT-Connect, token calls) without HTTPS (unless the switch is set) or with credentials in the address before anything is sent.
- **Certificates:** validation uses the container's certificate authorities, the authorities in `deploy/docker/ca/` (see Technical handbook, *Internal certificate authority*) and, per connection, **Additional trusted certificate authority (PEM)**. A certificate for another host name is always rejected.
- **Certificate pin:** optional SHA-256 of the public key (SPKI), Base64 or hex, several separated by commas. The connection is then only accepted if a certificate of the chain carries exactly this key. To determine it:

```
sh openssl s_client -connect dms.example.org:443 </dev/null 2>/dev/null \ | openssl x509 -pubkey -noout | openssl pkey -pubin -outform der \ | openssl dgst -sha256 -binary | base64
```

- **Client certificate (mTLS):** PKCS#12 file with private key and password (mandatory for XTA 2). File and password are stored protected together; the password may contain any character.
- **Proxy:** if `HTTPS_PROXY` is set, traffic goes through the proxy; the address check then happens before sending. Internal targets belong in `NO_PROXY` so that the connect-time check applies.
- **Limits:** 15 s to connect, 30 s per request (uploads to CMIS 600 s, XTA 2 and FIT-Connect 5 minutes per call), responses at most 50 MB. A target blocked by the address check counts as a permanent error.
- **Secrets** (passwords, tokens, OAuth secrets, SFTP keys, client certificates, webhook secrets) are stored in the database encrypted with ASP.NET Data Protection (keys in the volume `dpkeys`). The panel never shows them again, only "stored – leave empty to keep". They never appear in exports, logs or the audit.
- **Changed target:** if address, token endpoint, XTA service, SFTP host key, CA, pin or **Allow unencrypted HTTP** of a connection change, a stored password or secret has to be entered again or removed, and likewise a stored client certificate uploaded again or removed; otherwise the panel refuses to save. The webhook signing secret is not affected. OAuth 2.0 access tokens are cached per token endpoint, client ID, secret and scope, so a connection only ever uses a token obtained with its own credentials.

Add the target systems to your firewall rules for outbound connections.

REST/JSON API

Creating an API client

Panel → **Interfaces** → **New API client**:

1. **General:** name, description, **Purpose of the transfer**, **Legal basis**.
2. **Authentication method:** - **Signed client assertion (private_key_jwt, recommended):** store the third-party system's public key as PEM (RSA 2048 bits or more, or EC), X.509 certificate or JWKS. No secret is stored on the server. - **Client secret (client_secret_basic):** the secret is shown **once** after saving; only its hash is stored. **Generate new secret** invalidates the previous one immediately.
3. **Permissions (scopes):** only as much as necessary.
4. **Allowed source IP addresses:** one address or CIDR network per line. **Empty means no access.** `*` allows every address and is not recommended. Behind a reverse proxy the forwarded address counts; `ReverseProxy:Enabled` must be set and the proxy registered.
5. **Valid until** (optional), **Token lifetime (seconds)** 60–3600 (default 900), **Requests per minute** 1–6000 (default 120).
6. **Areas**, **Data protection** and **Inbound direction** as described in the respective sections.
7. Save (re-authentication), then **Submit for approval**.

The client ID starts with `ffi-`. **Revoke client** withdraws access permanently and deletes the credentials.

Requesting a token

Client credentials grant against `https://<server>/connect/token`. Without the `scope` parameter the client receives all assigned scopes; with `scope` (space-separated) only the requested ones. Requesting a scope that is not assigned to the client is rejected (`invalid_request`, unknown scopes `invalid_scope`).

With a client secret:

```
curl -sS -u "ffi-…:<secret>" -d grant_type=client_credentials \
  -d scope="integration:cases.read integration:documents.read" \
  https://ff.provider.example/connect/token
```

With private_key_jwt: the third-party system signs a short-lived JWT (RS256 or ES256) with `iss` and `sub` = client ID, `aud` = base address of the server (e.g. `https://ff.provider.example/`), a unique `jti` and a lifetime of a few minutes, type `client-authentication+jwt`, and sends it as `client_assertion` with `client_assertion_type=urn:ietf:params:oauth:client-assertion-type:jwt-bearer`.

The token endpoint additionally checks state **Active**, expiry, IP list and the active organisation licence. A rejection comes back as `unauthorized_client` with the error code (e.g. `integration.ip_denied`) in `error_description`. Every API request checks the same conditions again; suspending, pausing and revoking take effect within 30 seconds even for tokens already issued.

Scopes

Scope	Allows
`integration:cases.read`	read cases (master data, status, approval; pseudonymised)
`integration:sensitive`	addition: clear names and confidential fields instead of pseudonyms; counts as opening the privacy settings, so only accounts that may open them can add it
`integration:persons.read`	read persons involved in a case
`integration:documents.read`	read the case file (metadata and files)
`integration:documents.write`	deliver documents to the case file and withdraw own deliveries
`integration:cases.write`	change approved field groups of cases, set external reference
`integration:appointments.write`	create appointments for cases
`integration:time.read`	read time entries and activity reports (also in the CMIS repository)
`integration:billing.read`	read rate agreements
`integration:catalogs.read`	read catalogues (measures, activity types, institutions, organisation text snippets)
`integration:catalogs.write`	maintain measures and activity types
`integration:directory.read`	read staff and teams (without permissions and devices)
`integration:changes.read`	read the change feed (identifiers only)
`integration:inbox.read`	query the status of own deliveries

Writing scopes only take effect together with the **Inbound direction** settings.

Resources

All paths are relative to `/api/integration/v1`.

Method and path	Scope	Content
`GET /health`	valid token	connection test: client name, scopes, server time
`GET /cases`	cases.read	case list; filters `updatedSince`, `status`, `helpType`, `areaId`; `cursor`, `limit` (default 50, at most 200)
`GET /cases/{caseId}`	cases.read	one case with `ETag`
`GET /cases/{caseId}/persons`	persons.read	participants with roles
`GET /cases/{caseId}/documents`	documents.read	case file entries with files (metadata)
`GET /documents/{documentId}`	documents.read	one document with `ETag`
`GET /documents/{documentId}/files/{index}/content`	documents.read	file content with `Repr-Digest: sha-256=…`, range requests possible
`GET /cases/{caseId}/activity-reports`	time.read	archived activity reports of the case
`GET /activity-reports`	time.read	all visible activity reports
`GET /activity-reports/{reportId}/content`	time.read	PDF of a report
`GET /time-entries`	time.read	time entries; filters `from`, `to`, `caseId`, `userId`; `cursor`, `limit` (default 100, at most 200)
`GET /cases/{caseId}/rate-agreements`, `GET /rate-agreements`	billing.read	rate agreements
`GET /catalogs/{catalog}`	catalogs.read	`measures`, `activity-types`, `institutions`, `text-snippets`
`PUT /catalogs/{catalog}/{externalId}`	catalogs.write	create or change a measure or activity type (`measures`, `activity-types`)
`GET /directory/users`, `GET /directory/teams`	directory.read	staff and teams
`GET /changes`	changes.read	change feed
`POST /cases/{caseId}/documents`	documents.write	deliver a document to the case file of a case
`POST /documents`	documents.write	deliver a document with a case reference via `caseReference`
`PATCH /cases/{caseId}`	cases.write	change approved field groups
`PUT /cases/{caseId}/external-reference`	cases.write	store the system's own external ID on the case
`POST /cases/{caseId}/appointments`	appointments.write	create an appointment
`GET /inbox`, `GET /inbox/{itemId}`	inbox.read	status of own deliveries
`DELETE /inbox/{itemId}`	documents.write	withdraw an own, not yet processed delivery

Values such as `status`, `helpType`, `approval.period` and the appointment kind are enumeration names (e.g. `Active`, `SocialPedagogicalFamilySupport`, `Month`); the OpenAPI description lists them completely.

Example `GET /cases/{caseId}` without clear names:

```
{
  "id": "6f1c0e8a-4c6b-4f0e-9a51-0d2f3b8e2c11",
  "fileNumber": "51.2-2026-0142",
  "alias": "K-0142",
  "status": "Active",
  "helpType": "SocialPedagogicalFamilySupport",
  "startDate": "2026-03-01",
  "endDate": null,
  "approval": { "minutes": 480, "period": "Month", "from": "2026-03-01", "to": "2026-08-31" },
  "responsibleAuthority": "Jugendamt Musterstadt",
  "caseworker": { "name": "[pseudonymisiert]", "contact": "[pseudonymisiert]" },
  "areas": [ { "id": "…", "name": "Ambulante Hilfen Nord" } ],
  "externalReferences": { "client:…": "FV-2026-0815" },
  "revision": 57,
  "updatedAt": "2026-10-26T14:03:11+01:00"
}
```

Conventions

- **JSON:** camelCase, ISO 8601 timestamps with offset, identifiers as UUID strings.
- **Errors:** RFC 9457 (`application/problem+json`) with the member `code`, e.g. `integration.scope_missing`. All codes are listed in the chapter *Troubleshooting*.
- **Pages:** opaque `cursor`; the response contains `items` and `nextCursor`, plus a header `Link: <…>; rel="next"`. Do not build the cursor yourself.
- **ETags:** `GET /cases/{caseId}` and `GET /documents/{documentId}` return `ETag: "<revision>"`; with `If-None-Match` the server answers `304`.
- **Writing:** document deliveries require `Idempotency-Key` (otherwise `428`), at most 200 characters without control characters; the server rejects longer keys with `422` `integration.payload_invalid` instead of truncating them. Per client the same key returns the same delivery, even for concurrent retries (unique database index; the file of the duplicate upload is discarded). An `externalId` that has already been accepted yields `409` `integration.idempotency_conflict`. `PATCH /cases/{caseId}` requires `If-Match` with the revision (otherwise `428`, with an outdated revision `412`).
- **Rate limiting:** per client the configured requests per minute; beyond that `429` with `Retry-After: 60`.
- **Audit:** every request writes `integration.api_access`, every rejection `integration.api_denied`.

Change feed

`GET /changes?after=<sequence>&limit=<n>` (default 500, at most 1000) returns `items` with `sequence`, `kind` (`upserted` or `deleted`), `entityType`, `entityId`, `caseId`, `occurredAt` and `next` for the next call. It contains only types the client has read scopes for (cases, persons, documents, time entries, rate agreements), only visible cases and no content. The server checks every entry outside cases against the current state: `upserted` only if the record still exists, is not deleted, private, restricted or pending deletion, and documents pass the client's document filters as on a direct fetch; `deleted` if the record has been removed entirely (identifier only, for purging locally) or the remaining record is neither private, restricted nor pending deletion and documents pass the filters. A page may therefore contain fewer entries than `limit`; keep following `next` until it is `null`. If `after` lies before the retention start of the sync feed, the server answers `410` with `integration.cursor_expired`: reconcile once completely via the list resources, then continue with the current sequence.

Webhooks

Setting up

Panel → **Interfaces** → **+ Webhook**:

1. Name, **Address** (HTTPS), purpose, legal basis.
2. Optionally authentication at the target (Basic, Bearer, OAuth 2.0 client credentials, client certificate) and TLS settings.
3. Choose **Triggers** and, under **Events**, the event types; without a selection all triggered events go out.
4. Save: the **signing secret** (`whsec_…`) is generated and shown **once**. Store it at the recipient. **New signing secret** replaces it immediately.
5. **Test connection** sends the event `de.fokusfamilie.test`. Then submit.

Events

Type	Trigger in the panel	When
`de.fokusfamilie.case.created`	Case is created or changed	first version of a case
`de.fokusfamilie.case.updated`	Case is created or changed	every further version
`de.fokusfamilie.case.closed`	Case is closed (whole case file)	status "Completed" or "Archived"
`de.fokusfamilie.document.finalized`	Document becomes final or signed	after the connection's privacy filter
`de.fokusfamilie.activity_report.closed`	Activity report is closed and archived	new archived report
`de.fokusfamilie.case.erasure_requested`	Erasure of a case is requested	erasure request or deleted case, without content
`de.fokusfamilie.inbound.accepted`	no trigger, selectable under **Events**	an incoming item was accepted (directly or after review)
`de.fokusfamilie.inbound.rejected`	no trigger, selectable under **Events**	an incoming item was rejected on review
`de.fokusfamilie.inbound.conflict`	no trigger, selectable under **Events**	field conflict on receipt in inbox mode
`de.fokusfamilie.test`	connection test	only on **Test connection**

The `de.fokusfamilie.inbound.*` events go to every active webhook connection that subscribed to them (or made no selection) and whose areas cover the case; items without a case only reach connections without an area binding. They carry `entityType IntegrationInbox`, the item ID and the link `/api/integration/v1/inbox/{id}`, but no rejection reason; the delivering API client queries the full status via `GET /inbox/{id}`.

Message format

Every message is a CloudEvent 1.0 in structured mode (`Content-Type: application/cloudevents+json`) and contains **identifiers and links only**. The recipient fetches the data with its API client.

```
{
  "specversion": "1.0",
  "id": "3b0d2f0e-7c1a-4f7e-8d55-9a3c1e2b4f60",
  "source": "urn:fokusfamilie:organization:2c9a…",
  "type": "de.fokusfamilie.document.finalized",
  "time": "2026-10-28T09:12:44+00:00",
  "datacontenttype": "application/json",
  "subject": "8e41…",
  "data": {
    "entityType": "StoredDocument",
    "entityId": "8e41…",
    "caseId": "6f1c…",
    "links": {
      "case": "/api/integration/v1/cases/6f1c…",
      "document": "/api/integration/v1/documents/8e41…"
    }
  }
}
```

Verifying the signature

Signing follows **Standard Webhooks**: headers `webhook-id`, `webhook-timestamp` (Unix seconds) and `webhook-signature` (`v1,<Base64>`, several separated by spaces). The signed string is `<webhook-id>.<webhook-timestamp>.<raw body>` with HMAC-SHA256; the key is the Base64 part of the secret after `whsec_`.

```
import base64, hashlib, hmac, time

def verify(secret: str, headers: dict, body: bytes, tolerance: int = 300) -> bool:
    msg_id, ts = headers["webhook-id"], headers["webhook-timestamp"]
    if abs(time.time() - int(ts)) > tolerance:
        return False
    key = base64.b64decode(secret.removeprefix("whsec_"))
    mac = hmac.new(key, f"{msg_id}.{ts}.".encode() + body, hashlib.sha256).digest()
    expected = "v1," + base64.b64encode(mac).decode()
    return any(hmac.compare_digest(expected, s) for s in headers["webhook-signature"].split())
```

Reject messages with a stale timestamp (recommendation: 5 minutes) and remember processed `webhook-id` values so that an intercepted message cannot take effect twice. Retries of the same delivery carry the same `webhook-id`.

Delivery behaviour

- `2xx` counts as delivered.
- `400`, `401`, `403`, `404`, `405`, `410`, `413` and `422` count as permanent: the delivery becomes **Failed permanently** immediately.
- All other responses, timeouts and network errors are retried (see *Operation*).
- The same version of an object is scheduled only once per connection and event type.

- Immediately before every delivery attempt, including retries, the server checks again whether the case is visible within the connection's current area and the document is allowed by its privacy settings. If not, the event is not sent but marked as **Skipped** with the reason `not_visible`.
- Erasure announcements (`de.fokusfamilie.case.erasure_requested`) are still sent, but only to connections whose area covers the case and to organisation-wide connections.

Inbound direction

Every API client and every CMIS or XÖV connection has the **Inbound direction** card; webhook and SIEM connections only work outbound. Everything is off by default.

Setting	Default	Effect
Deliver documents to the case file	off	new documents as case file entries, append only
Accepting documents	inbox	**Inbox (review and accept)** or **Accept directly**
Change case master data (API client)	off	only the chosen **Allowed field groups**: File number; Responsible youth welfare office and caseworker; Approval (volume, period); Status and end of help
Accepting field changes	inbox	as for documents
Create appointments (API client)	off	appointments for visible cases; created immediately
Maintain catalogs (API client)	off	measures and activity types via `PUT /catalogs/…`
Case matching	external reference or file number	**Only via stored external reference**, **Only via file number** or both; a file number must be unique
Max. file size (MB)	25	1–2048
Max. deliveries per hour	200	1–100,000, per client or connection
Allowed file types	PDF, PNG, JPEG, DOCX	MIME types, one per line

Practitioners' free texts are never writable, there are no external deletions. Delivered files are scanned for malware like every upload, stored encrypted and reach the devices via sync.

Delivering documents

`POST /cases/{caseId}/documents` as `multipart/form-data` with the part `metadata` (JSON) and the part `file`:

```
curl -sS -H "Authorization: Bearer $TOKEN" -H "Idempotency-Key: fv-2026-0815-hpg" \
  -F 'metadata={"title":"Help plan 2026","documentType":"imported","externalId":"FV-4711"};type=application/json' \
  -F 'file=@helpplan.pdf;type=application/pdf' \
  https://ff.provider.example/api/integration/v1/cases/6f1c…/documents
```

- `title` is mandatory (at most 200 characters); `documentType`, `externalId` and `createdAt` are optional.
- Without a known case ID: `POST /documents` with `caseReference` in the metadata (external reference or file number, depending on **Case matching**). Without a unique case the document lands in the inbox in state **Assign case**; before that it passes the same checks as any other delivery (attachment policy, content against the MIME type, malware scan).
- Response `202` with the delivery status and `Location` pointing to `/inbox/{id}`; with **Accept directly** `201`.
- Accepted documents are final, carry the interface's name as author and appear in the case history with the source "Interface".

Changing case master data

`PATCH /cases/{caseId}` with `If-Match: "<revision>"` and a JSON object that may only contain `fileNumber`, `responsibleAuthority`, `caseworker` (`name`, `contact`), `approval` (`minutes`, `period`, `from`, `to`), `status` and `endDate`. Fields outside the approved groups lead to `403` `integration.field_group_denied`.

If a practitioner changed a field since the last import from this system, the practitioner wins: the change goes to the inbox as a **Conflict**. `PUT /cases/{caseId}/external-reference` stores the system's own external ID, through which later deliveries find the case.

Inbox and review

Items in state **To review**, **Conflict** or **Assign case** appear on the page **Incoming (interfaces)**. Accounts with **Review and accept interface inbox items** review them within their own administrative scope and choose **Accept**, **Assign** (choose a case) or **Reject**. The same rules as in the case view apply: items and the case picker only show cases that are not deleted, not marked for deletion and not restricted in processing. Items without a case are only visible to accounts with organisation-wide administrative scope. If an item's case or the chosen target case does not meet these rules, the server refuses accept, assign and reject; if a case is restricted in processing, for example, its items disappear from the list. The workflow from the professional side is described in the Admin panel handbook.

Accept is atomic: the target record, the entry in the sync change feed, the external reference and the state “Accepted” are written in one transaction. If a second person accepts the same item at the same time, their acceptance waits and then fails without creating a duplicate.

Withdrawing

`DELETE /inbox/{itemId}` withdraws an own delivery as long as it is unprocessed: at any time while in the inbox, after acceptance only as long as nobody has changed the document. After that only a person in Focus Family deletes.

CMIS: filing in the DMS or e-file

Focus Family files documents via **CMIS 1.1 browser binding** (JSON) in a document management system or e-file, for example Alfresco, Nuxeo or systems with a CMIS layer.

Setting up

Panel → **Interfaces** → **+ DMS/e-file (CMIS)**:

1. **Address** of the browser binding, e.g. `https://dms.example.org/alfresco/api/-default-/public/cmis/versions/1.1/browser`. All calls, including those to addresses reported by the repository (`repositoryUrl`, `rootFolderUrl`), must stay on the scheme, host and port of this address; otherwise the server aborts before any credentials are sent (`permissionDenied` "… differs from the configured address").
2. **Authentication at the target:** Basic, Bearer, OAuth 2.0 client credentials (client ID as user name, secret, **Token endpoint**, optionally **Scope**) or client certificate.
3. **CMIS repository: Repository ID** (empty: first repository), **Base folder**, **Document type** and **Folder type** (default `cmis:document` and `cmis:folder`), **Secondary types (aspects)**, **Property mapping**, **Store changed documents as a new version (check-out/check-in)**.
4. **Triggers** (all by default): Document becomes final or signed (optionally **Signed documents only** and a list of document types), Activity report is closed and archived, Case is closed (whole case file), Manually from the case view.
5. **Data protection**, **Folder template**, **Areas**, optionally **Inbox folder (inbound)**.
6. Save, **Test connection**, submit.

Filing structure

The **Folder template** determines the path below the base folder. Placeholders: `{year}` (year the help started), `{fileNumber}` (falls back to the case alias), `{alias}`, `{caseId}`, `{helpType}`, `{section}`. Case file sections are named in the target "Verträge und Einwilligungen", "Berichte und Dokumentation", "Nachweise", "Eingänge", "Notizen" and "Sonstiges". The server creates missing folders.

- **Document final:** the document is filed. If it changes later, a versionable type gets a new version, otherwise another document with a name suffix. Unchanged versions are not transferred again.
- **Case closed** and **manual handover:** all eligible documents of the case plus the case file index `fokusfamilie-akte.json` in the case folder (format `fokusfamilie-case-file/1` with document ID, title, section, version, SHA-256 and object ID in the target).
- **Activity report:** the archived PDF. The server checks new reports in the order they were received and only advances its progress marker once the whole run, including all reports with the same received time, has been queued; after an interruption the next run repeats the group. Queuing is idempotent, so nothing is lost and nothing is handed over twice.

Properties

The **Property mapping** has one line per property: `CMIS property = source`. Sources: `fileNumber`, `alias`, `caseId`, `documentId`, `documentType`, `section`, `version`, `status`, `signed`, `createdAt`, `author`, `sha256`, `helpType`, `title`. Names and author only appear in clear text with **Transfer clear names**. The properties must exist in the document type or in a registered aspect.

Connection test

Test connection shows every step separately: repository reachable, document type present, folder type present, property mapping valid, base folder ready, test folder created, test document stored, new version created, content read back, test objects removed and, where configured, inbox folder present. Missing types and properties are named in plain text next to the step.

Inbox folder (inbound)

With direction **Inbound** or **Both directions** and **Deliver documents to the case file**, the server polls the **Inbox folder** every **Polling interval (minutes)** (1–1440, default 5):

- Subfolders of the inbox folder name the case (file number or external reference). Documents directly in the inbox folder need the mapping line `inbox:caseReference = <CMIS property>`.
- The server moves taken-over documents into the **Folder for taken-over documents**. Documents without a matching case stay where they are and appear as **Assign case** in the inbox.
- Documents filed by Focus Family itself are not taken back in.
- Downloads are limited to the connection's **Max. file size** while reading and abort after 10 minutes; the connection then shows a `413` or `408` error, the polling run ends and the document stays in the folder until it is removed or replaced in the DMS. Documents refused by a check (file type, size, hourly limit, malware) stay in the folder and are checked again at the next poll. Polling errors do not count towards the automatic pause.
- Once a version of a document has been taken over, later versions of the same version series are refused (`integration.idempotency_conflict`) and stay in the inbox folder. If the idempotency key built by the server is longer than 200 characters, it uses a hash internally.

Focus Family CMIS repository

Conversely, a DMS can read the case files via `https://<server>/cmis/browser` (CMIS 1.1 browser binding, repository ID `focusfamily`).

Authentication and permissions

- An **API client** with the scopes `integration:cases.read` and `integration:documents.read`; for filing additionally `integration:documents.write` and **Deliver documents to the case file**.
- Authentication either with **Basic** (client ID and client secret, only for the client secret method) or with a **Bearer token** from the token endpoint. After `RateLimiting:AuthPermitLimit` failed Basic logins (default 100) from one IP address within a minute, further Basic requests from it receive 429 `integration.rate_limited` for the rest of the minute.
- The folder `Taetigkeitsnachweise` and its activity reports are only visible with the additional scope `integration:time.read`; without it they answer `objectNotFound`.
- State, expiry (`integration.client_expired`), IP list, licence, rate limit, areas and privacy settings of the client apply as for the REST API. Access and denials are audited as there, as `integration.api_access` and `integration.api_denied`.

Structure

```
/ (Focus Family)
├── Faelle
│   └── <file number or alias> (<short id>)          type ff:caseFolder
│       └── <case file section>
│           └── <document>                           type ff:document
└── Taetigkeitsnachweise
    └── Taetigkeitsnachweis <number>.pdf
```

Properties of the own types: `ff:caseId`, `ff:fileNumber`, `ff:alias`, `ff:documentType`, `ff:section`, `ff:status`, `ff:signed`, `ff:sha256`, `ff:source`. Supported calls: repository info, `typeChildren`, `typeDefinition`, `query`, `contentChanges` (object IDs only) and per object `object`, `content`, `children`, `parent`, `parents`, `allowableActions` and `versions`. Not supported: ACL, renditions, multi-filing, folder trees, join and ordering.

Queries and limits

`query` understands a deliberately small subset of CMIS SQL:

```
SELECT * FROM ff:document WHERE IN_FOLDER('case:…') AND cmis:lastModificationDate >= TIMESTAMP '2026-10-01T00:00:00Z'
```

- `FROM` only `cmis:document`, `ff:document`, `cmis:folder` or `ff:caseFolder`.
- `WHERE` only `IN_FOLDER('<id>')` and comparisons `=`, `>=`, `>` with a quoted value, combined with `AND`.
- At most 1000 results per query; at most 2000 cases or 5000 documents are searched; `children` returns at most 1000 entries per page.

The server rejects everything else with `invalidArgument`.

Filing documents

`createDocument` (form with `cmisaction=createDocument`, file in the part `content`) is only allowed in a case folder. Optionally `ff:documentType` and `ff:externalId` as properties and `Idempotency-Key` as a header. The document goes through the inbound direction like a REST delivery (scan, inbox or direct acceptance). `delete` only withdraws own, not yet processed deliveries.

XÖV and xdomea

What is transferred

- **Outbound:** on case closure (or manual handover) the server builds an **xdomea transfer** (message `0401`, version **3.1.0** or **4.0.0**) as ZIP package `<process ID>_Abgabe.Abgabe.0401.zip`. The package contains the message and the primary documents of the eligible case file entries with SHA-256 checksum, structured as one process per case file section or as a single process (**Case file structure**). Retention period (years, default 10) and **Disposition** (A – worthy of archiving, B – to be appraised, V – destroy; default B) go into the metadata. Every message is validated against the official xdomea schema before sending; an invalid package is not sent.
- **Confirmation:** an incoming import confirmation (`0402`) marks the transfer in the transfer log as **Acknowledged by recipient**; failures of single objects appear in the error text.
- **Inbound:** incoming transfers (`0401`) are validated, their documents go through the inbound direction (case matching via the file's identifier). If the sender asks for an import confirmation, the server sends back a `0402` with success or error per object.

Enter sender and recipient by name and optionally with a DVDV identifier (**Sender (organization name)**, **Sender ID (DVDV, optional)**, **Recipient (e.g. youth welfare office)**, **Recipient ID (DVDV, optional)**).

Transports

Transport	Address	Direction	Notes
Directory	empty	both	hand-over to an existing gateway through a directory in the container
SFTP	`sftp://host:port`	both	host key fingerprint mandatory; password or private key
XTA 2	URL of the XTA web service	both	XTA 2 version 4 (KoSIT), client certificate mandatory
FIT-Connect	base URL of the Submission API	outbound only	OAuth 2.0 client credentials, end-to-end encryption

Incoming packages may be at most 200 MB. The server polls the inbox at the connection's polling interval; for directory and SFTP it processes up to 50 packages per poll.

Directory

Outbox and inbox lie below the XÖV root directory `/var/lib/fokusfamilie/xoev` (setting `Integrations__Xoev__FolderRoot`), default `outbox` and `inbox`. The server rejects paths outside the root directory. Packages are first written as `.partial` and then renamed; only `*.zip` files are read and afterwards moved to `inbox/processed` or `inbox/failed`.

The root directory is **not a volume** in the standard stack. Mount a host directory that your gateway can also reach and give it to the container user (UID 1654):

```
services:
  app:
    volumes:
      - /srv/fokusfamilie/xoev:/var/lib/fokusfamilie/xoev
```

```
mkdir -p /srv/fokusfamilie/xoev/outbox /srv/fokusfamilie/xoev/inbox
chown -R 1654:1654 /srv/fokusfamilie/xoev
docker compose up -d --force-recreate app
```

SFTP

- **SFTP host key (SHA256 fingerprint)** is mandatory; without a matching fingerprint the server does not connect, a mismatch is reported with the presented fingerprint. To determine it: `ssh-keyscan host | ssh-keygen -lf -`.
- Authentication with **Username** and password or a private key (PEM) in the **Secret** field.
- **Outbox folder** and **Inbox folder** relative to the SFTP account (default `outbox`, `inbox`).
- Every SFTP operation is time-limited: connect, list, rename, delete, move and connection test to 60 seconds (`Integrations__Xoev__SftpOperationTimeoutSeconds`), uploading and downloading a package to 600 seconds (`Integrations__Xoev__SftpTransferTimeoutSeconds`). A timeout counts as a transient error: a send is retried with backoff and counts towards the automatic pause, a failed inbound poll is repeated at the next poll.
- The size limit for incoming packages applies to the bytes actually received. An oversized package is aborted with "Incoming package exceeds the size limit." and moved to `inbox/failed`. If a download exceeds the time limit, the package stays in the inbox and is retried at the next poll.

XTA 2

- **Service (service URI from the DVDV)**, **Identifier type** (default `xoev`), **Business scenario**, optionally **Mark as test message (TestMsg)**.
- Sender and recipient identifiers are the author and reader of the transport order and mandatory.
- The client certificate (PKCS#12) of the XTA account goes under **TLS, certificates and network**.
- If the gateway has separate ports, `{port}` in the address stands for `management`, `send` or `msgbox`.
- Sending uses `createMessageID` and `sendMessage` (MTOM), receiving `getStatusList`, `getMessage` and `close`; the connection test calls `checkAccountActive`.
- The implementation follows the specification and is tested against a specification mock. Test with your gateway and its test account before production use.

FIT-Connect

- Authentication via **OAuth 2.0 client credentials** (client ID, secret and **Token endpoint** of the FIT-Connect authorisation service).
- **Destination ID** as UUID, **Public service (service key)**, **Name of the service**, optionally **Schema of the submission data**.
- The xdomea message is sent as submission data and the primary documents as attachments, each encrypted with the destination's public key (JWE, RSA-OAEP-256 and A256GCM). The key must be RSA with at least 4096 bits and allow `wrapKey`; the destination must offer the service and the schema. Before encrypting, the server checks the key's certificate chain (`x5c`): the key must match the leaf certificate, the leaf must not be a CA and must allow key encipherment, and the chain must lead to a root of the German administrative PKI (embedded PCA-1-Verwaltung-17, -20, -23 and -26; additional roots via `Integrations__Xoev__FitConnectTrustedRootsPem` or `…File`). Revocation is checked online (`Integrations__Xoev__FitConnectRevocationMode`, default `Online`). If only the revocation status cannot be determined, the transfer is retried with backoff; revoked or untrusted certificates end as **Failed permanently**.
- FIT-Connect is a sending-only route. The implementation is tested against a specification mock; test in the FIT-Connect test environment before production use.

User provisioning with SCIM 2.0

The identity provider (e.g. Microsoft Entra ID, Keycloak, Okta) creates, updates and deactivates practitioner accounts.

Setting up

1. Panel → **Interfaces** → **User provisioning (SCIM 2.0)**: switch on **Allow SCIM provisioning** and save.
2. **SCIM access tokens** → **Generate new token** (label, re-authentication). The token (`ffscim_…`) is shown **once**.
3. In the identity provider enter `https://<server>/scim/v2` as SCIM base address and the token as bearer token ("secret token").
4. Optionally **Groups** → **teams**: one line per mapping, `group name = team name`.
5. Choose **Rights of new SCIM accounts**: **None – the administration grants rights** (default, recommended) or **Role “practitioner”**. Only someone with **Manage users** may switch to **Role “practitioner”**.

Behaviour

- New accounts are active practitioners **without a licence seat**; `userName` must be an e-mail address. With SSO active they are created as SSO accounts. Their permissions follow **Rights of new SCIM accounts**: with **None**, app sign-in stays blocked until the administration grants permissions under **Users** (badge “App login blocked”); with **Role “practitioner”** they immediately receive the permissions for cases, appointments, tasks, times and own text snippets, but no administrative permissions. The administration assigns the licence seat in the panel.
- Flow: SCIM creates the account → the practitioner signs in via SSO; on the first sign-in Focus Family links the SSO identity to the account via the e-mail address verified by the provider → app sign-in succeeds as soon as the account has operational permissions.
- `active=false`, `PATCH` with `active` or `DELETE` **deactivate** the account and end its sessions; nothing is deleted.
- Only practitioner accounts whose permissions lie within the “team lead” role (“practitioner” role plus team-lead permissions) or are app-only permissions are visible and changeable via SCIM. Any other permission (e.g. system, data protection or operations-admin permissions, managing users or teams, managing catalogues) makes the account invisible to SCIM (`404`) and removes it from the count; permissions can never be changed via SCIM.
- Groups only act through the mapping: members of a mapped group become members of the team (role practitioner). SCIM records a separate grant per group and team membership; if several groups are mapped to the same team, their grants count independently, and a membership only ends when no group grants it any more. When adding and reactivating, the area check applies as in the panel: a team with an area only accepts staff of that area. The server records skipped members as `integration.scim_change` with `members_skipped=N` (result “Denied”).
- If a membership already existed before SCIM first grants it, the server records its origin: an active manually assigned membership stays after the last SCIM grant is removed; a previously inactive one is active only while a group grants it and is set inactive again afterwards. Memberships created by SCIM are deleted.
- `replace` on `members` and `PUT` synchronise fully (an empty list removes all of the group's grants); `remove` and `DELETE` of the group only remove its own grants. `GET` on a group lists all members to whom it grants an active membership in the currently mapped team, including memberships originally assigned manually. Other groups have no effect.
- Concurrent changes to the same team membership are serialised (database row lock within the transaction). If the identity provider removes a person through several groups of the same team at the same time, no active membership remains without a granting group.
- When a group is renamed (`PATCH` on `displayName` or `PUT`) or you change the **Groups → teams** mapping and save, the server reconciles: grants in a team that no longer matches are removed, and the members are granted a membership in the newly mapped team (with the area check). If the group is no longer mapped to any team, its grants are only removed. Saving in the panel records `integration.scim_change` with `group_map_reconciled skipped=N`.
- Supported: `/ServiceProviderConfig`, `/ResourceTypes`, `/Schemas`, `/Users` and `/Groups` with GET, POST, PUT, PATCH, DELETE; filters only `eq` on `userName`, `emails.value`, `externalId`, `id` (groups: `displayName`, `id`); at most 200 entries per page. Not supported: bulk, sorting, ETags, password change.
- Every change writes `integration.scim_change`. **Revoke** ends provisioning through a token immediately.

Provider notes

- **Microsoft Entra ID:** enterprise application → provisioning → automatic; "Tenant URL" = SCIM base address, "Secret token" = SCIM token. Map `userPrincipalName` or `mail` to `userName`.
- **Keycloak:** SCIM extension with target URL and bearer token; group synchronisation only needed for the mapped groups.
- **Okta:** app with SCIM 2.0, authentication "HTTP Header" with the token; "Push Groups" for the mapped groups.

Calendar subscriptions (iCalendar)

Practitioners subscribe to their appointments in their work calendar (Outlook, Thunderbird, iOS).

- **Switching on:** panel → **Interfaces** → **Calendar subscriptions (iCalendar)** → **Allow calendar subscriptions**.
- **Level of detail** for all subscriptions: **Only "Busy" (recommended)**, **Appointment type and case alias** or **Appointment type, case alias and location**. Only an account with the data protection permission can choose more than "Busy", because subscriptions are stored on third-party servers (e.g. Microsoft 365); the choice sits on the same page and additionally requires **Manage interfaces**. Private appointments and appointments of private cases always appear only as "Busy", names never.
- **Subscription address:** every person in the panel under **Calendar subscription** → **Create subscription address**. The address `https://<server>/feeds/calendar/<token>.ics` is shown only once, is valid for one year and is personal and secret. **Create new address** invalidates the previous one, **End subscription** revokes it.
- **Content:** own appointments from 30 days back to 180 days ahead, time zone from `Organization__TimeZone` (default `Europe/Berlin`); calendar programs may fetch at most every 5 minutes (`Cache-Control: private, max-age=300`).
- The server answers `404` for invalid, expired or revoked addresses, switched-off subscriptions, deactivated accounts and organisations without a licence.

SIEM forwarding (syslog)

Audit chain events go to a SIEM or security monitoring. Only identifiers are transmitted, no case data; no approval is needed.

- **Address:** `tls://host:port` (recommended, RFC 5425, usually port 6514) or `tcp://host:port` with **Allow unencrypted TCP (internal network only)**. Certificates, pin and client certificate as for the other connections.
- **Message format:** RFC 5424 with structured data or CEF. Framing by octet counting.
- **Event filter:** audit action prefixes, one per line (e.g. `user.`, `sso.`, `integration.`, `privacy.`); empty means all.
- Forwarding starts at the time of activation (no back-filling of older events) and runs in blocks of up to 500 events. On errors the server retries from the same position.

- **Time limit:** connecting, the TLS handshake, writing and flushing of a block are limited to 30 seconds in total. A receiver that does not respond yields the error “The syslog endpoint did not complete the TLS handshake or the transfer within 30 seconds.”; the attempt counts as a failure (automatic pause after 10 in a row), the position in the audit chain is kept, and the other interfaces keep running. **Test connection** uses the same limit.

Example RFC 5424 (facility authpriv, severity 6 on success, 4 on denial, 3 on failure):

```
<86>1 2026-10-28T09:15:02.123Z ffhost focusfamily – integration.approved [ff@32473 seq="18231" action="integration.approved" result="Success" actor="…" targetType="Integration" targetId="…" correlation="…" hash="…"] integration.approved Success
```

CEF: `CEF:0|SoftwareNotion|FocusFamily|<version>|<action>|<action>|<severity>|rt=… act=… outcome=… suid=… cs1Label=targetType cs1=… cs2Label=targetId cs2=… cs3Label=correlation cs3=… cn1Label=sequence cn1=…` ; CR and LF in header and extension fields become spaces.

Manual handover from the case view

If an active, outbound CMIS or XÖV connection has the trigger **Manually from the case view**, the case view shows accounts with **Edit cases** the button **Hand over to target system**. After choosing the **Target system** and confirming, the server transfers the case file with the released documents as on case closure (CMIS: documents and case file index; XÖV: xdomea transfer). The connection's privacy settings apply. The handover appears in the transfer log and in the audit (`integration.manual_handover`).

New connections start "from now on": the server does not transfer existing documents and closed cases retroactively. Use the manual handover for existing case files.

Exporting and importing the configuration

Under **Transfer configuration**, **Export configuration** exports API clients and connections as a JSON file (`focusfamily-schnittstellen-<date>.json`), for example for a test or second server.

- Secrets, client secrets and client certificates are never exported. Public keys of private_key_jwt clients, CA certificates and pins are included; areas are referenced by name.
- **Import configuration** (file at most 2 MB, re-authentication) creates everything as a **Draft**. Approvals are not taken over, API clients get new client IDs.
- Entries with an existing name are skipped. Areas are matched by name; if an entry lists areas that are missing or ambiguous on the target server, it is not imported, because it would otherwise reach further than in the export. Create the areas or give them unique names and import again. Entries without area restriction are imported organisation-wide. Wider privacy settings and the scope `integration:sensitive` are only taken over when an account with the data protection permission imports; otherwise the defaults apply and the scope is dropped with a warning. Inbound limits outside the permitted ranges (max. file size 1 KB to 2 GB, deliveries per hour 1 to 100,000) prevent the import of the entry.
- After the import the panel lists everything to check: new client secrets, secrets and certificates to enter again, privacy settings reset to the defaults.

Operation

How delivery works

A background service follows the server's change feed every 5 seconds, schedules matching events per active connection and delivers them. Delivery only goes to active, outbound connections of licensed organisations whose outgoing transfers are not paused; each run is split evenly across these connections, so a paused or heavily loaded connection does not hold up the others. With several app instances only one works at a time (PostgreSQL lock). It polls inbox folders and incoming packages at each connection's polling interval and forwards SIEM events in the same cycle.

An organisation counts as licensed when it has sponsored seats, or purchased seats with a subscription that is active, in its grace period or has a billing issue. Without a licence, deliveries and polling rest; the token endpoint, the REST API, the CMIS repository, SCIM and calendar subscriptions refuse (`integration.license_required`, `403` or `404`). SIEM forwarding continues. Once the licence check passes, work continues without data loss. The server caches the licence state for five minutes; a rejection by the licensing server (subscription inactive, organisation invalid) counts as “not licensed”. Only while the licensing server cannot be reached does a previously confirmed licence stay valid for up to 24 hours.

Transfer log and retries

Panel → **Interfaces** → **Transfer log**, filterable by connection and status:

Status	Meaning
Pending	scheduled
Delivered	delivered; the object ID in the target is shown under **Object**
Failed, will retry	next attempt after 1 min, 5 min, 30 min, 2 h, 6 h, 24 h
Failed permanently	permanent error or after the seventh attempt
Skipped	not applicable, e.g. event type not subscribed or case or document no longer visible to the connection at the time of delivery (`not_visible`, `case_not_visible`)
Acknowledged by recipient	xdomea import confirmation received

Retry reschedules one delivery, **Retry all failed** up to 1000 at once; the attempt counter starts again. The overview shows per connection **Consecutive failures**, **Last transfer** and **Queue**; the last error text is shown in the connection.

Pausing and emergency stop

- **Pause** stops a connection or API client, **Resume** resets the failure counter; pending transfers go out in the next cycle, transfers waiting for a retry keep their scheduled time, and permanently failed ones only go out again via **Retry**.
- **Automatic pause:** after 10 consecutive failures (deliveries or SIEM forwarding; setting `Integrations__Dispatch__FailingThreshold`) an active connection switches to **Paused** with the reason `auto:consecutive_failures`, audited as `integration.paused`. The connection's status card explains this. Fix the cause, **Test connection**, then **Resume**.
- **Pause all outgoing transfers (emergency stop)** under **Basic settings** stops every outbound delivery and SIEM forwarding. Events stay in the change feed and are caught up after resuming. The REST API stays available.
- **Delete connection** removes the connection's queue and delivery log. Data already transferred to the target remains there. The disclosure register for access requests under Art. 15 GDPR is kept.
- A paused connection catches up as long as the change feed still contains the events (sync retention, see Technical handbook). After longer pauses, hand over missing case files manually.

Monitoring figures

`/metrics` (see Technical handbook, *Monitoring*) contains:

Figure	Meaning
`fokusfamilie_integration_clients_active`	active API clients
`fokusfamilie_integration_connections_active`	active connections
`fokusfamilie_integration_connections_paused`	paused connections
`fokusfamilie_integration_connections_pending_approval`	connections and API clients awaiting approval
`fokusfamilie_integration_outbox_pending`	pending and retrying transfers
`fokusfamilie_integration_outbox_retrying`	transfers that failed at least once
`fokusfamilie_integration_outbox_dead_lettered`	permanently failed transfers – should be 0
`fokusfamilie_integration_outbox_oldest_pending_age_seconds`	age of the oldest open transfer
`fokusfamilie_integration_deliveries_24h`	transfers of the last 24 hours
`fokusfamilie_integration_inbox_pending_review`	incoming items awaiting professional review

Useful alerts: `outbox_dead_lettered > 0`, `outbox_oldest_pending_age_seconds > 86400`, `connections_pending_approval > 0` over several days, `inbox_pending_review` rising permanently.

Audit events

Event	Occasion
`integration.client_created`, `integration.client_changed`, `integration.client_secret_rotated`, `integration.client_revoked`	API clients
`integration.connection_created`, `integration.connection_changed`, `integration.connection_deleted`, `integration.connection_tested`	connections
`integration.submitted`, `integration.approved`, `integration.rejected`, `integration.paused`, `integration.resumed`	approval and operation
`integration.settings_changed`	basic settings, approval policy
`integration.delivered`, `integration.delivery_failed`, `integration.delivery_retried`	deliveries
`integration.api_access`, `integration.api_denied`, `integration.token_issued`, `integration.token_revoked`	access and tokens
`integration.inbound_received`, `integration.inbound_accepted`, `integration.inbound_rejected`	inbound direction
`integration.scim_change`, `integration.manual_handover`, `integration.config_exported`, `integration.config_imported`	SCIM, handover, configuration

Configuration parameters

The defaults suit most installations. Enter deviations in a `docker-compose.override.yml` under `services.app.environment`.

Setting	Default	Meaning
`Integrations__Dispatch__Enabled`	`true`	background service on or off
`Integrations__Dispatch__IntervalSeconds`	`5`	cycle of the background service
`Integrations__Dispatch__ScanBatch`	`500`	changes per connection and cycle
`Integrations__Dispatch__DeliveryBatch`	`25`	deliveries per cycle
`Integrations__Dispatch__FailingThreshold`	`10`	consecutive failures until the automatic pause
`Integrations__Network__TimeoutSeconds`	`30`	time limit per request
`Integrations__Network__UploadTimeoutSeconds`	`600`	time limit per CMIS upload
`Integrations__Network__MaxResponseBytes`	`52428800`	largest response (50 MB)
`Integrations__Network__AllowLoopback`	`false`	allow loopback targets (test environments only)
`Integrations__Network__AllowLinkLocal`	`false`	allow link-local targets (test environments only)
`Integrations__Xoev__FolderRoot`	`/var/lib/fokusfamilie/xoev`	XÖV root directory
`Integrations__Xoev__MaxIncomingBytes`	`209715200`	largest incoming package (200 MB)
`Integrations__Xoev__SftpOperationTimeoutSeconds`	`60`	time limit per SFTP operation (connect, list, rename, delete, move, connection test)
`Integrations__Xoev__SftpTransferTimeoutSeconds`	`600`	time limit per SFTP upload or download of a package
`Integrations__Xoev__FitConnectTrustedRootsPem`, `Integrations__Xoev__FitConnectTrustedRootsFile`	empty	additional root certificates for the FIT-Connect key check (e.g. test environment)
`Integrations__Xoev__FitConnectRevocationMode`	`Online`	revocation check of the FIT-Connect key
`RateLimiting__AuthPermitLimit`	`100`	failed Basic logins per IP address and minute at the CMIS repository
`Organization__TimeZone`	`Europe/Berlin`	time zone of calendar subscriptions

Troubleshooting

Symptom	Cause	Remedy
token request `unauthorized_client`, `integration.client_inactive`	client not active (draft, waiting, paused, revoked)	submit, have it approved or resume
`integration.ip_denied`	source IP not in the list or list empty	extend the IP list; behind a proxy check `ReverseProxy`
`integration.scope_missing`	scope not assigned or not requested in the token request	add the scope (triggers approval) or request without `scope`
connection stays "Awaiting approval"	the data protection role has not decided yet	approval by another person with the data protection permission
"Approval must be given by someone other than …"	four-eyes principle: you created or last changed the interface	person with the data protection permission who neither created nor last changed it
"The configuration was changed after the page was opened. Please reload and review the current state again."	the interface was changed after the page was opened (or the request carried no revision); the approval only applies to the reviewed state	reload the page, review the current state and decide again
import: ": not imported, area(s) … not found or ambiguous …"	areas of the entry are missing or ambiguous on the target server	create the areas or give them unique names, then import again
"Approvals require a local account …"	signed in via SSO	approve with a local account
connection test: address not allowed	target resolves to loopback, link-local or a metadata address	enter the correct address; check DNS
TLS error	internal certificate authority unknown, pin does not match, wrong host name	enter the CA under TLS or place it in `deploy/docker/ca/`; determine the pin again
webhook "Failed permanently" with HTTP 401/403	recipient rejects authentication or signature	check the signing secret at the recipient, regenerate if needed; then **Retry**
"redirects are not followed"	recipient answers with a redirect	enter the target address directly
CMIS: document type or property missing	type or aspect not created in the DMS	create the type or adjust the mapping; the connection test shows the names
xdomea validation …	package violates the schema (e.g. missing mandatory data)	check sender and recipient, retention and file number; the error text names line and element
SFTP "Host key mismatch"	host key changed or entered incorrectly	determine the fingerprint again and compare it with the operator
XÖV directory not writable	root directory not mounted or wrong owner	set up the bind mount, `chown 1654:1654`
SCIM `401`	token wrong, expired or revoked	generate a new token
SCIM `403`	SCIM provisioning not allowed or no licence	**Allow SCIM provisioning**
SCIM `404` for a known person	the account has permissions outside the “team lead” role and the app-only permissions	maintain the account in the panel
calendar subscription `404`	subscription ended, expired, subscriptions switched off, account deactivated or no licence	create a new subscription address; check the licence
SIEM receives nothing	filter too narrow, emergency stop active, TLS error	clear the filter, **Test connection**
connection suddenly **Paused** with reason `auto:consecutive_failures`	target unreachable or refusing ten times in a row	read the connection's error text, fix the cause, **Resume**
widening of privacy settings is not saved	mandatory approval off, account without the data protection permission	have the data protection role make the change

REST API error codes

Code	HTTP	Meaning
`integration.scope_missing`	403	scope of the resource missing
`integration.area_denied`	–	reserved; the server answers objects outside the areas with `integration.not_found`
`integration.ip_denied`	403	source IP not allowed
`integration.client_inactive`	403	client not active
`integration.client_expired`	403	validity expired (also at the CMIS repository)
`integration.license_required`	403	no active organisation licence
`integration.not_found`	404	object unknown or not visible
`integration.cursor_invalid`	400	cursor damaged
`integration.cursor_expired`	410	the change feed no longer holds the position
`integration.rate_limited`	429	rate limit
`integration.inbound_disabled`	403	inbound direction off for this kind
`integration.field_group_denied`	403	field group not approved
`integration.payload_invalid`	400/422	request invalid
`integration.file_too_large`	413	file above the limit
`integration.mime_type_denied`	415	file type not allowed or content does not match the type
`integration.malware_detected`	422	malware scan found something
`integration.precondition_required`	428	`If-Match` missing
`integration.precondition_failed`	412	revision outdated
`integration.idempotency_key_required`	428	`Idempotency-Key` missing
`integration.idempotency_conflict`	409	key reused with different content
`integration.inbound_conflict`	–	field conflict; the item waits in state **Conflict** in the inbox
`integration.inbound_quota_exceeded`	429	deliveries per hour or storage quota exhausted
`integration.withdraw_denied`	403	delivery already processed
`integration.case_not_writable`	–	case deleted, restricted or private in the meantime; recorded as reason on the item

Glossary

Term	Explanation
API client	third-party system with its own credentials for the REST API and the CMIS repository
Change feed	consecutively numbered list of all changes on the server; basis of sync and interfaces
CloudEvents	CNCF standard for event messages
CMIS	Content Management Interoperability Services (OASIS), standard for accessing document management
DVDV	German administration services directory, provides identifiers and services for XTA
External reference	identifier of a third-party system stored on a case or document
FIT-Connect	FITKO transmission infrastructure for applications and messages to public administrations
Idempotency	repeating the same request takes effect only once
mTLS	TLS with a client certificate
Inbox	page **Incoming (interfaces)** for the professional review of incoming content
private_key_jwt	client authentication with a signed JWT instead of a secret (RFC 7523)
Pseudonymisation	replacing names with alias, identifier or `[pseudonymisiert]`
SCIM	System for Cross-domain Identity Management (RFC 7643/7644)
Scope	permission of an API client
SPKI pin	SHA-256 of a server certificate's public key
SSRF	server-side request forgery: abusing the server to reach internal addresses
Standard Webhooks	open specification for signing webhooks
xdomea	XÖV standard for exchanging and disposing of records
XÖV	XML in public administration, KoSIT framework
XTA 2	XÖV transport adapter, web service interface to OSCI intermediaries

Commissioning checklist

1. Define accounts: technical administration with **Manage interfaces**, at least one **other** person with the data protection permission and a local account.
2. Outbound firewall rules for the target systems; internal CAs in `deploy/docker/ca/` or per connection.
3. For XÖV via directory: set up the bind mount for `/var/lib/fokusfamilie/xoev`.
4. Create the connection or API client, enter purpose and legal basis, keep privacy settings minimal.
5. **Test connection** until all steps are green.
6. Submit and have it approved.
7. Hand over a first real case manually or trace the first API call in the audit.
8. Set up monitoring alerts for `fokusfamilie_integration_*`.
9. Export the configuration and file it with the operating documentation.

Focus Family · Interfaces handbook · Version 1.3.0 · October 2026