bimover

Filum, technical: what IT wants to know before rollout

Permissions, data paths, archive format and verifiability. A sheet for the IT of an architecture or planning office.

As at 19.09.2026. Every statement here comes from the code as it stands in the tree today, not from a brochure. Where a source in the repository is more precise, it is named. Not legal advice.

Filum is not published yet. The macOS version is in preparation for the App Store, the Windows version is on hold. This sheet describes the state of the program, not a delivery date.

1. What Filum does

An architecture or planning office runs its project correspondence in Microsoft 365. Each person files their project mail into an Outlook folder. Filum reads that folder through Microsoft Graph and puts the contents into a folder on the office NAS: the originals as `.eml`, the attachments unpacked, the conversations as HTML, plus an overview with search that works straight off the disk. Every run writes a record covering the entire holding, chained to the previous one and timestamped by a third party.

Three properties determine everything else:

- Filum does not run a service. There is no bimover server holding mail and no database holding an archive. The archive is a folder on a medium the office chooses itself.
- Filum works with the sign-in of the person starting the run. There is no service account and no tenant-wide read access.
- The archive is readable without Filum and verifiable without Filum. Neither is a statement of intent; both sit as files in the folder.

2. Permissions

Delegated, not application-level

Filum signs in through the multi-tenant registration «Filum Mailarchiv», a public client with no secret on the workstation. On Windows a device code runs, on macOS a sign-in window. The app then sees exactly the mailboxes and folders the signed-in person may open anyway, including shared and delegated ones.

The alternative would be application-level access. It would need `Mail.Read` for the whole tenant, that is read access to every mailbox in the office, and a secret stored somewhere. Filum has neither, and that is why rollout needs no project.

Permission	What for
`Mail.ReadWrite`	read the project folder, fetch raw bytes, move messages into the project folder while collecting
`Mail.ReadWrite.Shared`	the same for shared and delegated mailboxes
`User.Read`	display name and address of the signed-in person, and from it the own domain
`offline_access`	keep the sign-in beyond a program restart

Write permission is on the list because collecting moves messages. Filum deletes nothing in any mailbox, and it writes no message.

Source: `Kern/crates/filum-graph/src/auth.rs` (permissions), `Kern/crates/filum-kern/src/zugang.rs` (registration).

When the tenant blocks third-party apps

If an organisation blocks user consent, the first sign-in fails. Then, and only then, Filum shows the prompt with the address for the one-time approval by whoever administers the accounts. Afterwards the app stands as an enterprise application in the organisation's own tenant and applies to everyone in the office.

There is deliberately no separate registration per customer. It would not narrow the access; it would only push a form onto the customer.

3. Data paths

Filum talks to four counterparts, and to nothing else.

Destination	What goes there	When
`login.microsoftonline.com`	sign-in, token renewal	when signing in and afterwards in the background
`graph.microsoft.com`	folder and message queries carrying the access token	during a run
`www.bimover.ch`	licence evidence, request for free seats, report after the run	before and after a run
Timestamp service, default `tsa.swisssign.net`	the hash of one record, nothing else	once per run

What goes to bimover are hashes and numbers: one SHA-256 per archived mailbox instead of the address, the number of runs, the number of messages, an archive identifier without the project name. Subject lines, addresses, file names and contents are not provided for, and a fixed list of forbidden field names in the tests holds that in place (`filum_kern::meldung`).

The report has a second purpose that may interest IT: everything proving an archive is unchanged otherwise sits in the same folder that whoever wanted to change it can write to. The server remembers, per archive, the highest number of runs and messages it has seen. If the same archive later reports fewer, Filum says so.

A diagnostic report is created only on request, passes through a dedicated gate in the core (`filum_kern::austritt`) and names neither subject lines nor addresses; of the storage path it has contained only the last folder since 19.09.2026. The house-wide basis for this is in `PRIVACY.md` one level up.

Requests to Microsoft go exclusively to `graph.microsoft.com`. Even the continuation address in a response, which Filum uses to page through results, is checked against that first; if it points elsewhere, Filum does not follow it and says so in the run report (`filum_kern::zugang::zeigt_auf_graph`).

---

4. Where the archive sits

A separate share on the NAS, for example `Mailarchiv`. Projects are subfolders inside it. There are no permissions per project, and that is a decision rather than an omission: an office of six does not maintain a permission matrix across forty projects, and what nobody maintains is wrong after the second staff change.

The four rules of the setup:

1. A separate share for the archive, not inside the project storage. The archive only grows, so its snapshots cost little space and may be kept for a long time.
2. Everyone writes with their own account; the staff group gets read and write. No shared archive account: Windows does not allow two simultaneous connections to the same server with different accounts.
3. Snapshots run automatically and belong to the NAS administration. For the staff they are invisible, and nobody on the staff administers the NAS.
4. Deleting happens only through the app, because the delete function leaves a record instead of a gap.

This holds because the check names every missing or altered file and the snapshot brings it back. That is exactly how the law treats changeable media. An earlier model with letterbox permissions (create yes, delete no) was abandoned on 14.09.2026: laborious to set up, on some devices enforceable only over SMB, and whoever had created a file could delete it anyway.

During a run a lock file sits in the archive so that two runs do not write at once. If a run breaks off hard, the lock counts as orphaned after two hours and the next run takes over.

Step-by-step instructions for Synology and QNAP, including access logging: `docs/backup-vorgabeblatt.md`.

5. What sits in the folder

```
2019-0234_neubau-sonnenhof/
  Mails ansehen.html     overview with search and filters, the main entrance
  Mails/                 one archive file per message (PDF/A-2u), chronological
  .filum/                hidden, everything else:
    originale/           the originals as .eml, untouched, flat
    verlaeufe/           one readable HTML per conversation
    anhaenge/            unpacked, deduplicated by content
    verzeichnis.csv      one row per message, for a spreadsheet or text editor
    belege/              holding records, timestamps (.tsr), anchors
    pruefen.py           verifies the archive without Filum
    verfahren.txt        procedural documentation
    laeufe/, pruefungen/ logs
```

The distinction everything hangs on: the evidence is the `.eml` alone. It is the most original form available of a message, with the attachments inside it. Everything else is a reading copy and can be regenerated from the `.eml` at any time: the archive file in `Mails/`, the unpacked attachments, the reading view, the index. A deviation in an original is a hard finding, a regenerated archive file is not.

File names look like this:

```
2019-04-02 0914 Eingang – Anna Muster – Baugesuch Sonnenhof a5d21a19.eml
```

Date and time first, so that any file manager sorts chronologically; local time, because people search for the time that stood in Outlook; plain ASCII, because macOS decomposes file names and Windows composes them and the behaviour over SMB with mixed forms is undefined. The real subject with umlauts sits in the HTML and in the CSV. At the end of the name stands a short identifier derived from the message identity.

As an option, Filum files new messages without the payload of their attachments. The original is then reassembled byte for byte from the skeleton and the parts under `.filum/anhaenge`, and check, extraction and the verification script all compute over the reassembled original. For an office where the same set of drawings sits in twelve mailboxes, that is the difference between twelve copies and one. A separate run converts already filed messages in either direction, without identifiers or records changing.

6. Evidence and verification

Every message has two identifiers. Its identity is a hash over the canonicalised form, not over the bytes: Exchange keeps MAPI and reassembles the MIME on every fetch, so a byte hash would invent new messages at every quarterly run. Alongside it sits the byte hash of the stored file, and that is what the check recomputes.

Every run writes a holding record: a hash over the entire holding, both identifiers per message, sorted, chained to the record of the previous run. On that record Filum obtains a timestamp under RFC

3161 and stores it raw next to it. If the timestamp service is down, the run finishes and the next one stamps afterwards.

Verification happens two ways that compute the same thing:

- in the app, on the project page,
- without Filum via `python3 -I .filum/pruefen.py`, which needs only the standard library and also recomputes the signature of the timestamps.

A parity test keeps both verifiers tied to each other so they cannot drift apart. The check has three outcomes rather than two: present, removed with a deletion record, missing without one. Without that distinction a lawful deletion would look like tampering.

7. Deleting

Deletion removes the original and the reading copy together with the attachments no other message needs, redacts the line in the record file and places a note with both identifiers next to it. The message is gone, the evidence that one was there remains, and the chain stays valid. Hashes without an original are no longer personal data.

Two properties that matter in operation: a deletion that breaks off mid-write can be selected again in «Auskunft und Löschung» and finished with the timestamp from back then. And if a symbolic link sits on the path to a file the deletion would touch, no plan is created at all, and Filum says so. A template for the office's deletion concept is in `docs/loeschkonzept-vorlage.md`.

8. The reading service, if staff should read in a browser

Anyone who does not want to mount a drive sets up the reading service on the NAS: a small service that serves exactly the reading view already present in the archive. It cannot write, there is no endpoint for it, and it does not record who reads which message when. Access is one shared reading password for the whole office; whoever wants finer separation separates at the network.

The key assurance: if the service fails or is never set up, every archive in the folder stays fully readable. The service is convenience, never a prerequisite. Setup: `docs/lesemaske-vorgabeblatt.md`.

9. Operation

Requirements. macOS 14 or newer. The Windows version requires Windows 10 version 1809, but it is on hold and not available today. Plus a Microsoft 365 account with an Exchange Online mailbox, an internet connection and write access to the archive share.

One run. The app lists the chosen folders, fetches the headers page by page, then the raw bytes in batches of twenty, builds the conversations, writes files, records and the reading view. Several mailboxes run alongside each other; within one mailbox at most three requests at a time, because that is where Outlook

draws the line. If Microsoft throttles, Filum waits the stated time and continues. A second run recognises what is already there and fetches only what is new.

After a break-off. An aborted run does not leave half an archive behind. Writing happens only once a file is complete, and the archive state is written last. A new run picks up the work.

Annual review. Once a year the app shows, across all projects, which retention periods have expired, and records the review as text. Nothing is ever deleted automatically.

Network and account. Without a valid licence no new run starts, and without a connection none either. Existing archives are untouched by this: opening, reading, checking, extracting, disclosure and deletion always work, even without an account and without a network. If bimover.ch does not answer for a while, stored licence evidence carries until its grace period, and the app says how many days that still is.

10. Licensing

What is counted are archived mailboxes, not people who start a run. Anyone who has access to a shared mailbox and archives it needs a licence for it.

The calculation is tiered like income tax: the rate of a tier applies only to the mailboxes within that tier. That way the total cannot fall when a mailbox is added. As at 20.09.2026; the source is `Kern/crates/filum-kern/src/lizenz.rs`:

Mailboxes	per mailbox and year
1 to 10	CHF 50
11 to 20	CHF 40
21 to 30	CHF 30
from 31	CHF 25

The smallest invoice is CHF 250. An office with twenty mailboxes pays CHF 900 a year; three years paid in advance cost two and a half annual fees. The licence covers the software. Support costs hours, and the first year comes with two of them. Purchasing happens on bimover.ch, not in the app.

11. Limits, stated openly

- Teams chats are not covered.
- Cloud attachments, that is OneDrive links instead of real file attachments, are archived as links. Graph provides no bytes for them.
- Drafts are skipped. They have neither a Message-ID nor transport headers.
- Filum does not prevent changes to the archive. It makes them detectable. Protection against accidental deletion comes from the NAS snapshot.

- A conversion run across very many messages can take longer than the two hours after which a lock counts as orphaned. If another run then takes over, a guard stops the conversion run before it writes, and it has to be started again.
- Anyone routing the traffic to Microsoft through their own proxy with their own certificate authority can forge the time against which Filum measures the grace period. That is deliberate effort, not a wrong clock.

12. Where it is written more precisely

Question	File
Archive format, identity, threading, reading view	`README.md`
What Filum holds in legal terms	`docs/aufbewahrung.md`
Archive share, snapshots, restoring, access logging	`docs/backup-vorgabeblatt.md`
Reading service on the NAS	`docs/lesemaske-vorgabeblatt.md`
The office's deletion concept	`docs/loeschkonzept-vorlage.md`
Tamper protection in detail	`docs/audit-veraenderbarkeit.md`
What counts as data collection in bimover apps	`../PRIVACY.md`