

IMAP ? Paperless Archive Pipe (archive.alai.no)

IMAP ? Paperless Archive Pipe (archive.alai.no)

Overview

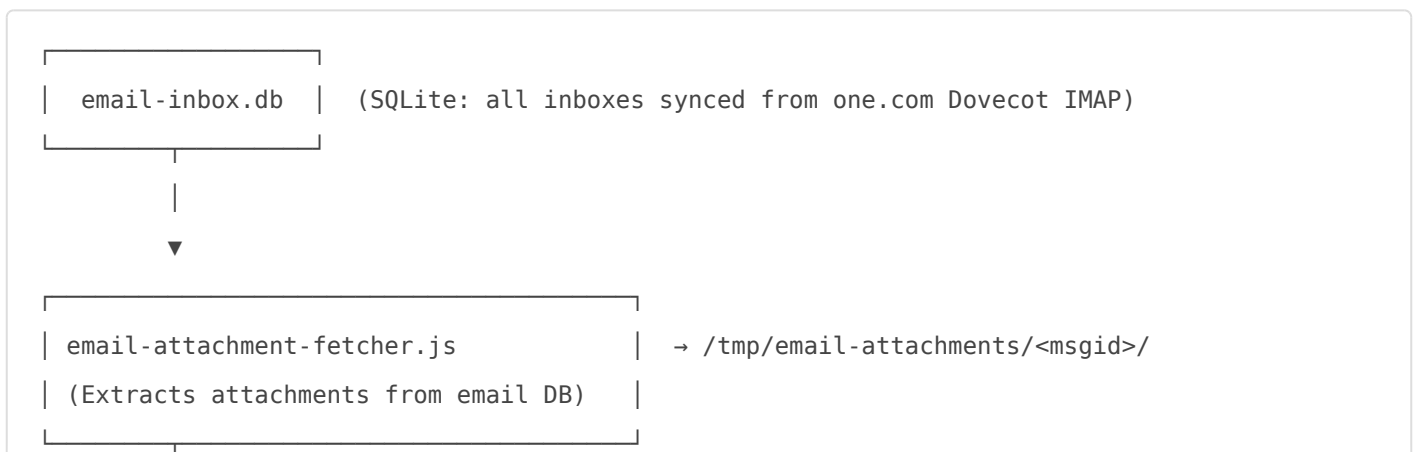
This pipe automates archival of email attachments (contracts, invoices, signed documents) from ALAI's IMAP inboxes into the centralized Paperless-ngx document management system at `archive.alai.no`.

Use Cases:

- Archive signed contracts received via email (e.g., SINTEF LOI, client MSAs)
- Store invoices, receipts, and financial documents
- Preserve legal correspondence with timestamped audit trail
- Upload arbitrary files that belong in long-term document archive

Architecture

The pipeline consists of two independent CLI tools that can be chained:



Note: Both scripts auto-fetch credentials from Bitwarden when `BW_SESSION` environment variable is set or `/tmp/bw-session` exists.

Usage Examples

Example 1: Archive a Single Email's Attachment

Most common workflow — fetch attachment from email DB and upload to Paperless:

```
# Step 1: Find the email ID (search by subject or sender)
node ~/system/tools/email-inbox.js list --account alem --limit 20

# Step 2: Extract attachments (creates /tmp/email-attachments/<msgid>/)
node ~/system/tools/email-attachment-fetcher.js 5480

# Step 3: Upload to Paperless with metadata
node ~/system/tools/paperless-upload.js \
  --file "/tmp/email-attachments/<msgid>/SINTEF_LOI_signed.pdf" \
  --correspondent "SINTEF" \
  --document-type "Contract" \
  --tags "legal,signed,sintef" \
  --title "SINTEF Letter of Intent - Forskningsrådet Application"
```

Example 2: Archive Arbitrary File (Skip Email Fetch)

Upload any local file directly:

```
node ~/system/tools/paperless-upload.js \
  --file "/Users/makinja/Downloads/Invoice_12345.pdf" \
  --correspondent "SnowIT" \
  --document-type "Invoice" \
  --tags "billing,2026-05" \
  --title "SnowIT Monthly Invoice - May 2026"
```

Example 3: SINTEF LOI First-Run (Historical Reference)

Exact command used for first production run (2026-05-08):

```
# Email ID 5480 from alem@alai.no inbox
node ~/system/tools/email-attachment-fetcher.js 5480

# Extracted: /tmp/email-attachments/<9a646c02-c6c5-5f08-35fb-3ab4ec45d1c1@one.com>/SINTEF_LOI_signed.pdf

node ~/system/tools/paperless-upload.js \
  --file "/tmp/email-attachments/9a646c02-c6c5-5f08-35fb-3ab4ec45d1c1@one.com/SINTEF_LOI_signed.pdf" \
  --correspondent "SINTEF" \
  --document-type "Contract" \
  --tags "legal,signed,sintef,forskningsradet" \
  --title "SINTEF Letter of Intent - Forskningsrådet Application"

# Result: Paperless doc #127
# https://archive.alai.no/documents/127/
```

Example 4: Using Message-ID Instead of Email DB ID

```
node ~/system/tools/email-attachment-fetcher.js \
  --message-id "<9a646c02-c6c5-5f08-35fb-3ab4ec45d1c1@one.com>" \
  --account alem
```

Script Details

email-attachment-fetcher.js

Location: /Users/makinja/system/tools/email-attachment-fetcher.js

SHA-256: a3a03d83516c2cc44bb8b0a3753d5c41f0feb9aff54f93fef5a1bb9e3699d739

Syntax:

```
node email-attachment-fetcher.js <email_db_id>
node email-attachment-fetcher.js --message-id <mid> --account <account>
```

Output: /tmp/email-attachments/<msgid>/<filename1>, <filename2>, ...

paperless-upload.js

Location: /Users/makinja/system/tools/paperless-upload.js

SHA-256: d185ed2f3f7ec816cb68f2a421e5762219449ebda420653d1a2f16558d2e06dd

Syntax:

```
node paperless-upload.js --file <path> [OPTIONS]
```

Options:

```
--correspondent NAME      Auto-creates if missing
--document-type NAME      Auto-creates if missing
--tags csv,list           Auto-creates if missing
--title "Document Title"
--no-poll                 Skip task completion polling
```

Exit Codes:

- 0 = Success
- 1 = Server error (network/API failure)
- 2 = Authentication failure
- 3 = Input validation error

Behavior:

- Polls Paperless task API for up to 30 seconds to confirm document consumption
- Auto-resolves correspondent/document-type/tag IDs via Paperless API (creates if missing)
- Sends 3 auth headers: CF-Access-Client-Id, CF-Access-Client-Secret, Authorization: Token ...

CF Access Service-Token Rotation

Current Token:

- Created: 2026-05-08
- Expires: 2027-05-08 (1 year TTL)
- Bypass Policy ID: 5df57dcf-eeec-4634-8668-68d5b8751334

Rotation Procedure:

1. Log in to Cloudflare Dashboard → Zero Trust → Access → Service Auth
2. Find policy for `archive.alai.no`
3. Click "Create Service Token" → name it `archive-pipe-YYYYMMv2`
4. Copy Client ID and Secret (shown only once)
5. Update Bitwarden item `e4fd63de-5989-4316-9092-1dfa72f2d2ee`:
 - Replace `CF_ACCESS_CLIENT_ID`
 - Replace `CF_ACCESS_CLIENT_SECRET`
6. Test with curl:

```
curl -I \  
  -H "CF-Access-Client-Id: <new_id>" \  
  -H "CF-Access-Client-Secret: <new_secret>" \  
  "https://archive.alai.no/api/"  
# Expected: HTTP 200 or 401 (not 302)
```

7. If 200 → revoke old token in Cloudflare dashboard

Troubleshooting

HTTP 302 Redirect from archive.alai.no

Symptom: `curl` returns `302 Found` to Cloudflare login page

Cause: Missing or expired CF Access service token

Fix:

1. Verify token exists in Bitwarden item `e4fd63de-5989-4316-9092-1dfa72f2d2ee`
2. Check token expiry in Cloudflare dashboard (Zero Trust → Service Auth)
3. If expired → rotate per procedure above
4. Verify script is passing headers (check `paperless-upload.js` code around line 40-60)

HTTP 401 Unauthorized from Paperless API

Symptom: `paperless-upload.js` exits with code 2

Cause: Invalid or missing Paperless API token

Fix:

1. Verify token in Bitwarden item `94227e4d-c55a-48fa-9421-05c649c5451e`

2. Test token directly:

```
PAPERLESS_TOKEN="..."  
curl -s -H "Authorization: Token $PAPERLESS_TOKEN" \  
  -H "CF-Access-Client-Id: ..." \  
  -H "CF-Access-Client-Secret: ..." \  
  "https://archive.alai.no/api/correspondents/" | jq -r '.count'
```

3. If null or error → regenerate token in Paperless UI (Settings → API Tokens) and update Bitwarden

Tag/Correspondent/Document-Type Creation Failures

Symptom: Script errors with "Failed to create correspondent X"

Cause: Paperless API permissions or schema validation failure

Fix:

1. Check Paperless UI → ensure API user has `documents.add_*` permissions
2. Verify tag/correspondent names don't contain invalid characters (use alphanumeric + spaces only)
3. Check Paperless logs on Azure VM:

```
ssh -i ~/.ssh/azure_alai alai-admin@4.223.110.181  
sudo docker logs paperless-webserver --tail 100
```

Email Attachment Not Found

Symptom: `email-attachment-fetcher.js` reports "No attachments found"

Causes:

- Email has no attachments (e.g., inline HTML only)
- Email not yet synced to `email-inbox.db` (daemon runs every 5 minutes)
- Wrong email ID or message-ID

Fix:

1. Verify email exists:

```
node ~/system/tools/email-inbox.js show <id>
```

2. Force IMAP sync:

```
node ~/system/tools/email-inbox.js sync --account alem
```

3. Check attachment MIME parts in raw email (look for `Content-Disposition: attachment`)

File Upload Stalls (No Response After 30s)

Cause: Paperless task processing slow or stuck

Fix:

1. Use `--no-poll` flag to skip task polling (upload completes instantly)
2. Check document manually in Paperless UI after 1-2 minutes
3. Restart Paperless workers if stuck:

```
ssh -i ~/.ssh/azure_alai alai-admin@4.223.110.181  
sudo docker restart paperless-worker
```

Provenance

This runbook documents the IMAP→Paperless archive pipeline built and validated under:

- **MC Task:** #100004 (Subtask 4 of 5)
- **Builder Teams:**
 - FlowForge (Subtask 1): CF Access service token creation
 - CodeCraft (Subtask 2): `email-attachment-fetcher.js` CLI
 - CodeCraft (Subtask 3): `paperless-upload.js` CLI
- **First Production Use:** 2026-05-08 20:02 UTC (SINTEF LOI archive → Paperless doc #127)
- **Documentation:** Skillforge (Subtask 4)
- **Operator:** John (orchestrator)

Related Resources:

- [archive.alai.no — Paperless-ngx Setup & Operations](#) (Infrastructure runbook)
- [Email Inbox — Setup & Operations](#) (IMAP sync daemon)
- Source: `~/system/tools/email-attachment-fetcher.js`
- Source: `~/system/tools/paperless-upload.js`

Last Updated: 2026-05-08 | MC #100004 | Skillforge

Revision #2

Created 2026-05-08 18:06:56 UTC by John

Updated 2026-06-14 20:02:39 UTC by John