Paperless-ngx — CF Access SSO Setup Plan

Implementation

Now available as skill `/cf-access-sso`. Execute via Skill tool with args: subdomain, service, container, vm_rg, vm_name, cf_user_email, [service_token_id]. Manual paste-ready commands below remain as fallback.

Skill path: `~/.claude/skills/cf-access-sso/SKILL.md`

Invoke example:

```
Skill('cf-access-sso',
  subdomain='archive.alai.no',
  service='paperless',
  container='alai-paperless-1',
  vm_rg='RG-ALAI-SUPPORT',
  vm_name='vm-alai-support',
  cf_user_email='alembasic@gmail.com',
  service_token_id='9d63505b-2e07-49e4-beb6-28b545a93bef'
)
```

Skill handles: pre-flight checks, user rename, env apply, container restart, CF Access app creation (with service token bypass + email allow policies), verification gate (curl 302 + Playwright screenshot), rollback script emission. Evidence written to: `/tmp/evidence-cf-sso-paperless/`

Paperless-ngx — CF Access SSO Setup Plan

> **STATUS: PLAN — NOT YET EXECUTED** Written: 2026-05-15 by John (AI Director) Execution: CEO terminal (az vm run-command + CF API) Prerequisite: review this page fully before executing

---

Current State (verified 2026-05-15)

Component	Current Config
URL	https://archive.alai.no
CF Access app	"All ALAI Services" wildcard `*.alai.no` (id: cd7cf0f0)
Dedicated archive app	None — wildcard catches all
IdP	Email OTP only (`alai-no.cloudflareaccess.com`)
Human login	Username + password (user `alembasic`, superuser)
API auth	DRF Token (`c9ec30192db3c95802349335edea4bca864a937a`)
IMAP pipe auth	CF service token (BW: e4fd63de) + Paperless API token
SSO	Not configured
Browser access	IP bypass fires for LAN (92.221.168.61) — no CF auth challenge

Key finding: CF Access only injects `Cf-Access-Authenticated-User-Email` when the **allow** policy fires. When IP bypass matches first, no identity header is set. Current bypass-first config means SSO cannot work for LAN browser sessions without restructuring the CF app.

Architecture Decision: Dedicated CF Access App

Create a **separate** CF Access app for `archive.alai.no` that:

- Authenticates CEO via Email OTP (allow policy, no IP bypass for browsers)
- Bypasses for the IMAP pipe service token (machine-to-machine remains token-based)
- Does NOT inherit the IP bypass from the wildcard app (exact-match app takes precedence)

The wildcard `*.alai.no` app continues to handle all other services and IP-bypass API access.

Header Chain (after SSO enabled)

```
CEO Browser
    ↓
Cloudflare CF Access (Email OTP challenge — once per 24h)
    ↓  injects: Cf-Access-Authenticated-User-Email: alembasic@gmail.com
Caddy reverse proxy (archive.alai.no → paperless:8000)
    ↓  forwards all headers by default
Paperless-ngx (Django) reads: HTTP_CF_ACCESS_AUTHENTICATED_USER_EMAIL
    ↓  matches username "alembasic@gmail.com" → auto-login
CEO is logged in, no password prompt
```

Execution Script

Run from CEO terminal (has full `az` auth). Do NOT execute all at once — verify each phase.

Phase 1: Rename Paperless user (preserve document ownership)

```
# SSH to Azure VM
ssh -i ~/.ssh/azure_alai alai-admin@4.223.110.181

# Rename 'alembasic' → 'alembasic@gmail.com'
docker exec alai-paperless-1 python manage.py shell -c "
from django.contrib.auth import get_user_model
User = get_user_model()
u = User.objects.get(username='alembasic')
print('Before:', u.username, u.email)
u.username = 'alembasic@gmail.com'
u.email = 'alembasic@gmail.com'
u.save()
print('After:', u.username)
"
# Expected output: After: alembasic@gmail.com
```

```
# Verify: list users
docker exec alai-paperless-1 python manage.py shell -c "
from django.contrib.auth import get_user_model
for u in get_user_model().objects.all():
    print(u.id, u.username, u.is_superuser, u.is_active)
"
```

Phase 2: Update Paperless env vars for trusted-header SSO

```
# On Azure VM — find docker compose file
ls /opt/alai/ /home/alai-admin/ 2>/dev/null
# Likely: /opt/alai/docker-compose.yml or /home/alai-admin/docker-compose.yml

# Add/update these env vars in the paperless service:
# PAPERLESS_ENABLE_HTTP_REMOTE_USER=true
# PAPERLESS_HTTP_REMOTE_USER_HEADER_NAME=HTTP_CF_ACCESS_AUTHENTICATED_USER_EMAIL

# Example edit (adjust path as needed):
# In docker-compose.yml, under paperless service environment:
#   - PAPERLESS_ENABLE_HTTP_REMOTE_USER=true
#   - PAPERLESS_HTTP_REMOTE_USER_HEADER_NAME=HTTP_CF_ACCESS_AUTHENTICATED_USER_EMAIL

# Restart paperless (NOT the whole stack — don't restart redis/gotenberg/tika):
docker compose -f /path/to/docker-compose.yml restart alai-paperless-1
```

Phase 3: Verify Caddy forwards the header

```
# Test from Azure VM (loopback):
# Simulate what CF Access would inject:
curl -s -o /dev/null -w "%{http_code}" \
  -H "Cf-Access-Authenticated-User-Email: alembasic@gmail.com" \
  -H "Cf-Access-JWT-Assertion: test" \
  http://localhost:8000/accounts/login/
# This should NOT auto-login (no Caddy = no trusted proxy check) — that's expected
# The real test is through Caddy (HTTPS from browser)

# Check Caddy config:
```

```
cat /opt/alai/Caddyfile 2>/dev/null || docker exec alai-caddy-1 cat /etc/caddy/Caddyfile
2>/dev/null
# Verify archive.alai.no block does NOT strip headers explicitly
# Caddy default: all request headers are forwarded to upstream
```

Phase 4: Create dedicated CF Access app for archive.alai.no

```
# Use CF API to create the dedicated app
CF_ACCOUNT_ID="d0ac2afb6bb5b298723b85a114151a04"
CF_EMAIL="john@basicconsulting.no"
CF_API_KEY="$(bw get item 'Cloudflare Global API Key' --session $(cat /tmp/bw-session) | jq -r
'.login.password')"
OTP_IDP_ID="ff0a28e6-2220-4de2-a82f-48385d88b163"
PIPE_TOKEN_ID="9d63505b-2e07-49e4-beb6-28b545a93bef"

curl -s -X POST \
  "https://api.cloudflare.com/client/v4/accounts/$CF_ACCOUNT_ID/access/apps" \
  -H "X-Auth-Email: $CF_EMAIL" \
  -H "X-Auth-Key: $CF_API_KEY" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "archive.alai.no — Paperless SSO",
    "domain": "archive.alai.no",
    "type": "self_hosted",
    "session_duration": "24h",
    "auto_redirect_to_identity": false,
    "http_only_cookie_attribute": true,
    "same_site_cookie_attribute": "lax",
    "app_launcher_visible": true,
    "allowed_idps": ["'"$OTP_IDP_ID"'"],
    "policies": [
      {
        "name": "archive-pipe service token bypass",
        "decision": "bypass",
        "precedence": 1,
        "include": [{"service_token": {"token_id": "'"$PIPE_TOKEN_ID"'"}}]
      },
```

```
      {
        "name": "CEO alembasic access",
        "decision": "allow",
        "precedence": 2,
        "include": [{"email": {"email": "alembasic@gmail.com"}}]
      }
    ]
  }'
# Save the returned app id — needed if you want to update or delete this app
```

Phase 5: Verify SSO works

```
# From CEO browser (Mac Air, NOT Mac Studio with VPN):
# 1. Clear cookies for archive.alai.no
# 2. Navigate to https://archive.alai.no
# 3. Should see CF Access OTP challenge — enter alembasic@gmail.com
# 4. Enter OTP from email
# 5. Should land directly on Paperless dashboard (logged in as alembasic@gmail.com)
# 6. Check: Profile → Settings — should show alembasic@gmail.com as username

# API/pipe verification (no regression):
source ~/.config/alai/paperless-token.env
curl -s --interface "$PAPERLESS_BIND_INTERFACE" \
  -H "Authorization: Token $PAPERLESS_TOKEN" \
  "$PAPERLESS_BASE/api/documents/?page_size=1" | grep '"count"'
# Should return document count — confirms API token auth still works
```

Rollback Procedure

If SSO breaks login:

```
# Method 1: Disable SSO via env (SSH or az run-command)
# Edit docker-compose.yml: set PAPERLESS_ENABLE_HTTP_REMOTE_USER=false
# docker compose restart alai-paperless-1
# Then login with alembasic@gmail.com + password

# Method 2: Emergency password reset (if locked out completely)
```

```
az vm run-command invoke \
  --resource-group RG-ALAI-SUPPORT \
  --name vm-alai-support \
  --command-id RunShellScript \
  --scripts "docker exec alai-paperless-1 python manage.py changepassword alembasic@gmail.com"

# Method 3: Delete the dedicated CF Access app (reverts to wildcard + IP bypass)
# Get the app id from Phase 4 output, then:
curl -s -X DELETE \
  "https://api.cloudflare.com/client/v4/accounts/$CF_ACCOUNT_ID/access/apps/<APP_ID>" \
  -H "X-Auth-Email: $CF_EMAIL" \
  -H "X-Auth-Key: $CF_API_KEY"
```

Risk Table

Risk	Likelihood	Mitigation
API token breaks after user rename	Low	Tokens bound to DB user ID (int), not username
Caddy strips CF header	Low	Default Caddy forwards all headers; verify Caddyfile
CEO locked out after SSO enable	Medium	Emergency: az run-command changepassword
IMAP pipe breaks	Low	Pipe uses service token + API token, unaffected by SSO
OTP fatigue	Low	24h session — one OTP per day max
`*.alai.no` wildcard still matches	Low	Exact-match app takes CF routing precedence
SSO header spoofing	Low	CF Access validates JWT; only CF can inject this header. Caddy only listens on localhost

---

What We Are NOT Doing

- **Not adding Google as IdP.** Email OTP is the only configured IdP. Google OAuth would require a Google Cloud project + OAuth consent screen setup. Out of scope for now.
- **Not using PAPERLESS_SOCIALACCOUNT_** approach. Trusted header is simpler and doesn't require OAuth app registration.

- **Not enabling PAPERLESS_APPS=allauth.** The HTTP remote user approach is the documented "trusted header" method for internal proxies.

Related Pages

- archive.alai.no — Paperless-ngx Setup & Operations — main ops runbook (page 2737)
- IMAP → Paperless Archive Pipe — IMAP pipe (page 2862)
- CF Access App ID (wildcard): cd7cf0f0-ab37-4b06-8d51-9f042fd7a4f6
- CF IdP (Email OTP): ff0a28e6-2220-4de2-a82f-48385d88b163
- BW: CF global key = "Cloudflare Global API Key", archive pipe token = e4fd63de

---

Appendix: Auth Strategy — Internal Phase (current)

Updated: 2026-05-16 by John (AI Director)

Status: ACTIVE — Email OTP only, 30-day sessions. Google OAuth deferred.

Auth Strategy (Internal Phase — current)

- **IdP:** Email OTP only (`onetimepin`, ID: `ff0a28e6-2220-4de2-a82f-48385d88b163`)
- **Session duration:** 720h (30 days) — CEO logs in once per month, not every visit
- **Allow list:** 3 CEO email aliases: `alembasic@gmail.com`, `alem@alai.no`, `alem@basicconsulting.no`
- **Rationale:** 1 internal user. Manual Google Cloud Console OAuth app setup is not justified for a single user. Email OTP with 30-day sessions provides equivalent UX (login friction ~once/month).

Session duration change evidence:

- Before: `24h` (required daily re-auth)
- After: `720h` (30 days) — applied 2026-05-16 via CF API
- Evidence files: `/tmp/evidence-cf-session-fix/app-before.json`, `/tmp/evidence-cf-session-fix/app-after.json`
- CF Access app updated at: 2026-05-16T19:51:55Z

Root Cause of Email OTP Failure (resolved 2026-05-16)

CF Access evaluates the allow policy **before** dispatching the OTP email. The original policy only had `alembasic@gmail.com`. When CEO entered `alem@alai.no`, CF rejected the request at the policy gate — no email was ever dispatched to Migadu. Migadu mailbox was healthy throughout.

Fix: policy updated to include all 3 CEO aliases. Policy ID: `a9e36b92-5158-4ced-a333-a8d84a67a705`.

Client-facing IdP Strategy — Deferred

Google OAuth IdP setup is deferred until the client-facing phase. Manual Google Cloud Console setup is not justified for 1 internal user when Email OTP + 30-day sessions already provides low-friction access.

Trigger to upgrade IdP:

- First paying customer onboarded to `archive.alai.no`, OR
- ALAI Workspace Google account configured (separate decision by CEO)

When triggered — build path:

- Option A (preferred): Build a proper client onboarding skill that creates a Google OAuth app via the Workspace Admin SDK (no Cloud Console UI required — fully automated). Dispatch to CodeCraft/Petter.
- Option B: Offer SAML 2.0 for enterprise clients (per-client IdP config in CF Access).

Until then: Email OTP scales to approximately 10 internal users without UX regression, given 30-day session duration.

IdP Tiers (target state — not yet active)

Tier	Who	Primary IdP	Fallback	Status
ALAI Staff	CEO + internal team	Email OTP (30d session)	—	ACTIVE
SME Clients	SnowIT and similar	Email OTP	—	Future
Enterprise Clients	Custom per-client	SAML 2.0 / OIDC	Email OTP	Future

---

Revision #6
Created 2026-05-16 17:55:39 UTC by John
Updated 2026-06-21 20:03:36 UTC by John