

ALAI Mail Topology — Migadu Domains, Mailbox Inventory, John's 19-Account Ingest Loop (2026-06-08)

ALAI Mail Topology & John's Email Ingest Loop

Last updated: 2026-06-08 (v2 — 19 accounts, daemon-path docs, himalaya touch-points) | **MC:** #103182 | **Built by:** FlowForge | **Validated by:** Proveo (Angie Jones) — PASS

1. Mail Infrastructure — Migadu (Single Account)

All ALAI product domains are hosted on **one Migadu account**. MX records for every domain point to the same two servers:

- `aspmx1.migadu.com` (priority 10)
- `aspmx2.migadu.com` (priority 20)

Domains on this account: `alai.no`, `bilko.io`, `bilko.cloud`, `bilko.company`, `snowit.ba`, `basicconsulting.no`, `basicfakta.no`, `lumiscare.com`

Migadu Admin Access

Item	Value / Location
Admin login	<code>alem@alai.no</code>

Item	Value / Location
API key	Vaultwarden item "migadu key" (86-char token — do NOT print)
IMAP host	<code>imap.migadu.com</code>
SMTP host	<code>smtp.migadu.com</code>
Web UI	https://admin.migadu.com

Migadu API Quirks (DO NOT FORGET)

- **GET aliases** — response key is `address_aliases`, not `aliases`.
- **Create alias** — must send JSON body `{"local_part": "...", "destinations": ["..."]}` with header `Accept: application/json` (omitting `Accept` = HTML response, silent fail).
- **Alias destinations MUST be same-domain.** Cross-domain targets (e.g. `info@alai.no` → `john@basicconsulting.no`) return HTTP 400. Route to a real mailbox on the same domain instead.
- **No catch-all rewrites** — verified via `/rewrites` endpoint (empty on all domains). Any email to a non-existent local-part that has no alias **bounces**.
- App-passwords for new mailboxes are created via `PUT /v1/domains/{domain}/mailboxes/{local_part}` and stored as Vaultwarden items (never in logs).
- **Migadu catch-all copy (alem@alai.no):** `alem@alai.no` is configured as a global catch-all copy recipient for all outgoing ALAI-managed-domain mail. This means emails sent FROM any ALAI account will also appear in `alem`'s INBOX. Because `alem` iterates before product accounts in the daemon list, it ingests those Message-IDs first; the `UNIQUE(message_id)` constraint causes product-account inserts to be no-ops. This affects ingest attribution for ALAI-origin probes only — external (non-ALAI) mail is not affected. See Section 6 for forwarding removal note.

2. Real Mailbox Inventory

These are the real mailboxes that exist in Migadu (verified 2026-06-08 via admin API). Only real mailboxes can be used as alias destinations.

Domain	Real mailboxes (local parts)
<code>alai.no</code>	john, alem, dev, post, admin
<code>bilko.io</code>	admin, sales, privacy
<code>bilko.cloud</code>	admin, sales
<code>bilko.company</code>	admin, sales

Domain	Real mailboxes (local parts)
snowit.ba	admin, info, asmir, enis
basicconsulting.no	john, info
lumiscare.com	hello, admin

Note: `basicfakta.no` is on this Migadu account but has no actively polled mailboxes in John's loop.

Note: `lumiscare.com` is ALAI's Migadu domain (our infrastructure). It is distinct from `caresafetyinnovations.com`, which remains a hard-stop boundary (see Section 6).

3. John's Email Ingest — All 19 Monitored Accounts

John's email ingest is managed by `~/system/tools/email-inbox.js` and polled by `~/system/daemons/email-agent.js`. As of MC #103182 final state (2026-06-08), **19 accounts** are registered in `email-inbox.db → email_accounts`.

Original 6 Accounts (pre-MC #103182)

Account name (DB key)	Email address	Vault item
john	john@basicconsulting.no	existing
info	info@basicconsulting.no	existing
alai	john@alai.no	existing
dev	dev@alai.no	existing
alem	alem@alai.no	existing
gmail	alembasic@gmail.com	existing

11 Product/Role Accounts (added MC #103182 round 1)

Account name (DB key)	Email address	Vault item name
post-alai	post@alai.no	Migadu — post@alai.no
admin-alai	admin@alai.no	Migadu — admin@alai.no

Account name (DB key)	Email address	Vault item name
sales-bilko-io	sales@bilko.io	Migadu — sales@bilko.io
privacy-bilko-io	privacy@bilko.io	Migadu — privacy@bilko.io
admin-bilko-io	admin@bilko.io	Migadu — admin@bilko.io
sales-bilko-cloud	sales@bilko.cloud	Migadu — sales@bilko.cloud
admin-bilko-cloud	admin@bilko.cloud	Migadu — admin@bilko.cloud
sales-bilko-company	sales@bilko.company	Migadu — sales@bilko.company
admin-bilko-company	admin@bilko.company	Migadu — admin@bilko.company
info-snowit	info@snowit.ba	info@snowit.ba IMAP
admin-snowit	admin@snowit.ba	Migadu — admin@snowit.ba

2 LumisCare Accounts (added MC #103182 round 2 — CEO directive 2026-06-08)

CEO directive: LumisCare must be in John's reading loop. lumiscare.com is ALAI's own Migadu domain — these are operational mailboxes, not CareSafety-boundary addresses.

Account name (DB key)	Email address	Vault item name
hello-lumiscare	hello@lumiscare.com	Migadu — hello@lumiscare.com
admin-lumiscare	admin@lumiscare.com	Migadu — admin@lumiscare.com

Note on hello@lumiscare.com forwarding: A Migadu direct forward from hello@lumiscare.com → alem@alai.no was active since 2026-05-24. This was **removed 2026-06-08** so the mailbox is polled directly under hello-lumiscare with clean labeling. Before removal, LumisCare contact mail appeared in the DB under alem (Migadu ingested the forwarded copy first). After removal, external mail to hello@lumiscare.com is stored under hello-lumiscare only. Confirmed behaviourally: gmail-origin probe stored as DB id=9195 under hello-lumiscare, not duplicated under alem.

App-passwords for the 5 newly created admin@* mailboxes (round 1) were generated via the Migadu API and stored as Vaultwarden items. Vault IDs: 558181ec, 8dfe8d2d, 2f38a16a, 7d0f9216, 2fb07c20.

4. Alias Map — Dead-Address Fixes (2026-06-08)

The following addresses were previously advertised (on websites, legal pages, landing pages) but did not correspond to any real mailbox — all mail to them was silently bouncing. Migadu aliases were created to route them to the nearest real same-domain mailbox.

Dead address (was bouncing)	Now routes to	Why
<code>info@alai.no</code>	<code>john@alai.no</code>	alai.no contact form was sending to this dead address — all website contact submissions were lost
<code>support@bilko.io</code>	<code>sales@bilko.io</code>	bilko.io landing mailto link
<code>podrska@bilko.io</code>	<code>sales@bilko.io</code>	bilko.io Bosnian support address on legal/terms pages
<code>legal@bilko.io</code>	<code>admin@bilko.io</code>	bilko.io legal/terms page
<code>security@bilko.io</code>	<code>admin@bilko.io</code>	bilko.io security disclosure address
<code>support@bilko.cloud</code>	<code>sales@bilko.cloud</code>	bilko.cloud landing mailto
<code>support@bilko.company</code>	<code>sales@bilko.company</code>	bilko.company landing mailto

Pre-fix state: Only `postmaster@{domain} → admin@{domain}` aliases existed. No rewrites, no catch-all. All other non-existent local-parts bounced.

Post-fix: All advertised addresses now deliver to a real monitored mailbox. Nothing bounces.

5. Contact-Form Routing

Product	Contact form path	Where mail ends up
alai.no website	Vercel serverless: <code>~/business/ALAI-Holding-AS/web/api/contact.js</code> (nodemailer)	Sends to <code>info@alai.no</code> (which now aliases to <code>john@alai.no</code> — monitored). <i>Was dead before 2026-06-08 fix.</i>
Bilko landing pages	Cloudflare Pages function: <code>apps/landing-*/functions/api/lead.js</code>	Posts to Slack <code>#ceo</code> channel (C0AFJDP9V6U) + writes to Cloudflare KV (<code>BILKO_LEADS</code>). No email path — separate from IMAP polling.

6. Boundary Accounts — NOT Polled (intentional)

Address	Reason not polled
---------	-------------------

<code>asmir@snowit.ba</code>	Personal mailbox belonging to Asmir (SnowIT partner). He reads his own mail.
<code>enis@snowit.ba</code>	Personal mailbox belonging to Enis. Same reason.
Any <code>*@caresafetyinnovations.com</code>	CareSafety hard-stop boundary — health/patient-adjacent service under external ownership. NOT on ALAI's Migadu account. Never poll. See CareSafety boundary memo in MEMORY.

Important distinction: `lumiscare.com` (ALAI's Migadu domain — `hello@`, `admin@`) IS polled. `caresafetyinnovations.com` (external operator) is the hard boundary, not `lumiscare.com`.

7. Daemon Architecture — Production Path

Understanding the daemon path is critical when debugging ingest issues or adding accounts.

Production Execution Path

- **LaunchAgent:** `com.john.email-agent` — starts `email-agent-wrapper.sh`, sets `HIMALAYA_DISABLED=1` via plist `EnvironmentVariables` key.
- **Wrapper:** `~/system/daemons/email-agent-wrapper.sh` — thin shell wrapper, does not set `HIMALAYA_DISABLED` itself.
- **Daemon:** `~/system/daemons/email-agent.js` — when `HIMALAYA_DISABLED=1`, all 19 accounts use the **legacy unseen-fetch IMAP path** (direct node-imap, proven stable).

Himalaya Layer — Present but Bypassed in Production

Even with `HIMALAYA_DISABLED=1`, the daemon still routes account resolution through `himalaya-adapter.js` `ACCOUNT_MAP`. If an account name is missing from `ACCOUNT_MAP`, the daemon throws `Unknown account: <name>` and the account is skipped entirely.

- **himalaya-adapter.js ACCOUNT_MAP** — must list all 19 accounts (currently L34–56).
- `~/config/himalaya/config.toml` — must have 19 `[accounts.*]` stanzas (verified: `grep count = 19`).
- When run without `HIMALAYA_DISABLED=1` (bare wrapper invocation), the `himalaya` binary is called and times out after 120s per account (~82 min total for 19 accounts). This is expected and non-destructive but slow. Production LaunchAgent always sets the env flag.

Validated (2026-06-08T13:15Z): Zero "Unknown account" errors in both daemon runs (wrapper + legacy). All 19 accounts have `last_checked_at = 2026-06-08T13:09:39Z`.

8. Components — All 8 Touch-Points

Adding any new account requires updating **all 8** of the following. Missing any one will cause silent failures or "Unknown account" errors.

#	File	What to change
1	<code>~/system/tools/email-inbox.js</code>	(a) Add <code>INSERT OR IGNORE INTO email_accounts (name, email) VALUES ('<name>', '<email>')</code> seed row. (b) Add a guarded migration block to extend the <code>emails</code> table CHECK constraint to include the new account name. The CHECK constraint is hardcoded and cannot be altered without rebuilding the table (SQLite limitation). The guard must use a unique string from the new account name (e.g. <code>!ddlRow.sql.includes("'<name>')</code>). All existing rows and all 25 columns must be preserved in the rebuilt table. This is the most error-prone step — see Section 9 for the gotcha detail.
2	<code>~/system/tools/mail-native.js</code>	Add account-name → Vaultwarden item-name entry in <code>VAULT_NAMES</code> map.
3	<code>~/system/tools/himalaya-adapter.js</code>	Add account-name → email entry in <code>ACCOUNT_MAP</code> (L34–56 area). Without this, the daemon throws "Unknown account" and skips the account entirely even in legacy mode.
4	<code>~/config/himalaya/config.toml</code>	Add a new <code>[accounts.<name>]</code> stanza. Required even when <code>HIMALAYA_DISABLED=1</code> .
5	<code>~/system/daemons/email-agent.js</code>	Add account to counts map (L2459 area). Also confirm it is present in the fetch loop and <code>last_checked_at</code> update loop (both must be mirrored).
6	<code>~/system/tools/email-imap-db-audit.js</code>	Add account to <code>ACCOUNTS</code> constant.
7	<code>~/system/tools/email-action-hard-check.js</code>	Add account to <code>ALL_MONITORED_ACCOUNTS</code> constant.

#	File	What to change
8	Vaultwarden (via <code>bw</code> CLI)	Create app-password item named <code>Migadu - <email></code> with the IMAP/SMTP password. New admin@ mailboxes require a new app-password generated via Migadu API (<code>PUT /v1/domains/{d}/mailboxes/{lp}</code>). Existing sales@/privacy@/info@ mailboxes may already have creds in Vaultwarden — check before creating.

Files Changed in MC #103182 (round 1 — 11 accounts)

All files modified additively. Round 1 changed 5 files (himalaya touch-points were added in round 2 as BLOCKER-2 fix).

File	Lines changed
<code>email-inbox.js</code>	L159-172 (seeds) + L141-208 (CHECK migration, 17-account guard)
<code>mail-native.js</code>	L76-88 (11 VAULT_NAMES entries)
<code>email-imap-db-audit.js</code>	L51 (ACCOUNTS 5→16)
<code>email-action-hard-check.js</code>	L14-22 (ALL_MONITORED_ACCOUNTS 17 accounts)
<code>email-agent.js</code>	L1853-1861 (fetch loop), L1889-1895 (last_checked_at loop)

Files Changed in MC #103182 (round 2 — LumisCare + BLOCKER-2 fix)

File	Lines changed
<code>email-inbox.js</code>	L212-311 (second guarded CHECK migration, 19-account guard: <code>!ddlRow2.sql.includes("hello-lumiscare")</code>); 2 new email_accounts seed rows
<code>mail-native.js</code>	L90-91 (hello-lumiscare + admin-lumiscare VAULT_NAMES)
<code>himalaya-adapter.js</code>	L34-56 (ACCOUNT_MAP expanded to 19 entries)
<code>~/config/himalaya/config.toml</code>	2 new [accounts.*] stanzas (19 total)
<code>email-agent.js</code>	L1862 (fetch loop), L1899 (last_checked_at loop), L2459-2468 (counts map)

File	Lines changed
<code>email-action-hard-check.js</code>	L24 (hello-lumiscare + admin-lumiscare in ALL_MONITORED_ACCOUNTS)
<code>email-imap-db-audit.js</code>	L60 (both accounts in ACCOUNTS array)

Known Minor Issue (pre-existing, non-blocking)

After SMTP send via `mail-native.js`, the IMAP post-send copy to Sent folder times out with `ETIMEOUT`. Delivery succeeds (Message-ID is logged). This is a cosmetic issue in the IMAP cleanup code — pre-existing, unrelated to MC #103182. Separate MC recommended.

9. GOTCHA — emails Table CHECK Constraint

This is the most dangerous footgun when adding new accounts. **Read before touching email-inbox.js.**

The `emails` table in `~/system/databases/email-inbox.db` has a hardcoded SQLite CHECK constraint:

```
account TEXT NOT NULL CHECK(account IN ('john','info','alai','dev','alem','gmail',
    'post-alai','admin-alai',
    'sales-bilko-io','privacy-bilko-io','admin-bilko-io',
    'sales-bilko-cloud','admin-bilko-cloud',
    'sales-bilko-company','admin-bilko-company',
    'info-snowit','admin-snowit',
    'hello-lumiscare','admin-lumiscare'
    ))
```

The trap: `INSERT OR IGNORE` silently discards rows that violate CHECK constraints — no exception is thrown, no warning is logged. If a new account name is not in this list, every email received by that account is permanently lost at ingest time. In MC #103182 this caused 27 real emails to be silently dropped before the issue was caught by Proveo.

The fix: SQLite does not support `ALTER TABLE ... MODIFY COLUMN` with a new CHECK constraint. The only way to extend it is to rebuild the table:

1. Read current DDL: `SELECT sql FROM sqlite_master WHERE type='table' AND name='emails'`
2. Guard the migration: check that the new account name is NOT already in the DDL (idempotency)

3. In a transaction: `CREATE TABLE emails_new (...same schema + extended CHECK...) → INSERT INTO emails_new SELECT * FROM emails → assert row count matches → DROP TABLE emails → ALTER TABLE emails_new RENAME TO emails → recreate indexes → COMMIT`
4. Rollback on any error or row count mismatch

The pattern already exists in `email-inbox.js` — follow it exactly. All 25 columns must be listed explicitly, including the post-migration additions: `delegated_to`, `delegated_at`, `deadline`, `body`, `triaged_at`, `auto_forwarded`.

10. Runbook — How to Add a New Mailbox to John's Loop

1. **Verify the mailbox exists in Migadu.**

Check via `GET /v1/domains/{domain}/mailboxes` using the admin API key ("migadu key" in Vaultwarden).

If it does not exist, create it via the admin UI or API first.
2. **Create an app-password for the mailbox.**

Use Migadu admin UI (Mailbox settings > App Passwords) or `PUT /v1/domains/{domain}/mailboxes/{local_part}`.

Store the password as a new Vaultwarden item named `Migadu - {email}`.
3. **[Touch-point 2] Add to `mail-native.js` `VAULT_NAMES` map.**

Key = your chosen account name (e.g. `sales-newdomain`), value = the Vaultwarden item name.
4. **[Touch-point 3] Add to `himalaya-adapter.js` `ACCOUNT_MAP`.**

Add `'<name>': '<email>'` in the `ACCOUNT_MAP` object. Without this step the daemon throws "Unknown account" and the account is silently skipped.
5. **[Touch-point 4] Add stanza to `~/.config/himalaya/config.toml`.**

Follow the existing pattern for a Migadu account stanza.
6. **[Touch-point 1a] Add the email_accounts seed to `email-inbox.js`.**

Append `INSERT OR IGNORE INTO email_accounts (name, email) VALUES ('<name>', '<email>')` in the seed block.
7. **[Touch-point 1b — CRITICAL] Add a guarded CHECK migration to `email-inbox.js` `getDb()`.**

Read Section 9 first. Guard: `!ddlRow.sql.includes("<name>")`. Extend CHECK to include new account. Rebuild table in a transaction preserving all 25 columns. Test idempotency.
8. **[Touch-point 6] Add to `email-imap-db-audit.js` `ACCOUNTS` array.**
9. **[Touch-point 7] Add to `email-action-hard-check.js` `ALL_MONITORED_ACCOUNTS` array.**
10. **[Touch-point 5] Add to `email-agent.js` `counts` map, `fetch` loop, and `last_checked_at` loop.**

All three locations must be mirrored.

11. Run syntax checks on all modified files.

```
node --check ~/system/tools/email-inbox.js && node --check ~/system/tools/mail-native.js && node --check ~/system/tools/himalaya-adapter.js && node --check ~/system/daemons/email-agent.js
```

12. Test connectivity.

```
node ~/system/tools/mail-native.js test --account <name> — expect IMAP OK + SMTP OK.
```

13. Restart the email-agent daemon (LaunchAgent: `com.john.email-agent`) so the updated accounts array and config take effect.

14. Proveo ingest probe.

Send a test email from a **non-ALAI sender** (e.g. gmail account) with subject `INGEST-PROBE-<name>-<timestamp>`. This avoids the Migadu catch-all pre-emption issue (see Section 1 API quirks). Trigger one daemon cycle. Confirm the row appears under the correct account name via `node ~/system/tools/email-inbox.js search "INGEST-PROBE"`.

15. If adding a new alias (not a real mailbox): create the Migadu alias first (same-domain destination only, with `Accept: application/json` header). Then proceed from step 3.

11. Validation Evidence (MC #103182 — Final)

Round 1 (17 accounts — 2026-06-08T11:24Z)

Check	Result
Code changes (5 files) verified by Proveo	PASS
DB registry — 17 rows in email_accounts	PASS
IMAP/SMTP connectivity — 11/11 new accounts	PASS
emails table CHECK migration (emails_new rebuild)	PASS — DDL confirmed, 4697 rows preserved
Ingest probes — 4/4 probe accounts persist to DB	PASS (round 2 probes after schema fix; DB ids 9052/9056/9057/9059/9062/9063/9064)
Regression — original 6 accounts	PASS — counts growing, timestamps advancing
No-loop / alias dedup (UNIQUE on message_id)	PASS — 0 duplicate message_ids
email-action-hard-check.js exit code	PASS — exit 0, 17 accounts in scope

Blocker found and fixed during round 1 validation: The `emails` table had a hardcoded CHECK covering only the original 6 accounts. `INSERT OR IGNORE` silently dropped 27 real emails before the migration was applied. See Section 9 for the full gotcha description.

Round 2 (19 accounts — LumisCare + daemon path — 2026-06-08T13:15Z)

Check	Result
ACCOUNT_MAP (himalaya-adapter.js) has 19 entries	PASS — L34-56 confirmed
config.toml has 19 [accounts.*] stanzas	PASS — grep count = 19
email-agent.js counts map has 19 accounts	PASS — L2460-2468
Zero "Unknown account" errors (wrapper run)	PASS — grep -c = 0 / 40 lines
Zero "Unknown account" errors (legacy/production run)	PASS — grep -c = 0
Zero silent drops / CHECK failures (production run)	PASS
admin-lumiscare ingest proof	PASS — DB id=9070 under admin-lumiscare
hello-lumiscare ingest proof (external sender)	PASS — DB id=9195 under hello-lumiscare (gmail-origin probe)
sales-bilko-cloud ingest proof	PASS — DB id=9193
sales-bilko-company ingest proof	PASS — DB id=9194
hello@lumiscare.com forwarding removal (behavioural)	PASS — gmail-origin stored only under hello-lumiscare, not duplicated under alem
All 19 last_checked_at fresh	PASS — 2026-06-08T13:09:39Z all accounts
No duplicate message_ids	PASS — 0 rows
Regression (orig 6 + prior 11)	PASS — row counts growing, timestamps fresh

Evidence files: </tmp/evidence-103182/flowforge-build.md>, </tmp/evidence-103182/proveo-validation.md>, </tmp/evidence-103182/daemon-wrapper-run.log>, </tmp/evidence-103182/daemon-legacy-run.log>

Revision #2

Created 2026-06-08 11:28:24 UTC by John

Updated 2026-06-08 13:16:09 UTC by John