Hermes SEG Email Flow

Hermes SEG Email Flow

Reference diagram for the inbound, outbound, and CipherMail-originated mail paths inside the Docker stack. Includes every listening port, every container, the milter chain at each smtpd service, and where body modifications occur relative to DKIM signing.

Container + port map (at a glance)

Container	Service / Daemon	Port(s)	Role
`hermes_postfix_dkim`	postfix smtpd (`:25`)	25	Inbound MX
`hermes_postfix_dkim`	postfix smtpd (`submission`)	587	Authenticated outbound (STARTTLS)
`hermes_postfix_dkim`	postfix smtpd (`smtps`)	465	Authenticated outbound (implicit TLS)
`hermes_postfix_dkim`	postfix smtpd (`:10026`)	10026	Re-injection (post-CipherMail)
`hermes_postfix_dkim`	postfix smtpd (`:10027`)	10027	CipherMail web-GUI originated mail
`hermes_postfix_dkim`	OpenDKIM **primary** (sv mode)	8891	Verify inbound / sign outbound at `:25` / `:587` / `:465`
`hermes_postfix_dkim`	OpenDKIM **sign-only** (s mode) — #232	8892	Sign post-CipherMail egress at `:10026` only
`hermes_mail_filter`	amavisd-new (filter)	10021	SpamAssassin + ClamAV + policy
`hermes_mail_filter`	amavisd-new (pickup / bypass)	10030	`BYPASSALLCHECKS` lane
`hermes_body_milter`	Python pymilter (#214/#226/#228/#230)	8893	Disclaimer + signature + banner + CID inline
`hermes_dmarc`	OpenDMARC	54321	DMARC verify / SPF alignment header

Container	Service / Daemon	Port(s)	Role
`hermes_openarc`	OpenARC (#229, flowerysong v1.3.0 built from source)	8893	ARC sealing at `:10026` only — RFC 8617 chain preservation across body mods
`hermes_ciphermail`	CipherMail SMTP	25	Encryption decisions + MIME rebuild
`hermes_dovecot`	LMTP	24	Local mailbox delivery

Milter listening side: the OpenDKIM/OpenDMARC/body_milter/OpenARC daemons listen on TCP and postfix smtpd connects to them per the `smtpd_milters` line in effect for each port.

> Note: `hermes_body_milter` and `hermes_openarc` both listen on internal port `8893`, but they are separate containers with their own network namespaces. Postfix reaches each by container name (`inet:hermes_body_milter:8893` vs `inet:hermes_openarc:8893`), so the shared port number causes no conflict.

Milter chains by smtpd port

The `smtpd_milters` chain order is set globally in `main.cf` (built from the `parameters` DB table by `generate_postfix_configuration.cfm`) and overridden per-service in `master.cf` for `:10026` and `:10027`.

smtpd port	Milter chain (in order)	Why
`:25` (inbound)	OpenDKIM **:8891** → OpenDMARC **:54321** → body_milter **:8893**	Verify DKIM, verify DMARC, then inject External Banner / disclaimer
`:587` `:465` (submission)	OpenDKIM **:8891** → OpenDMARC **:54321** → body_milter **:8893**	Sign DKIM first (before body mods), then disclaimer/signature inject — **wrong order; see #232 outbound fix history**
`:10026` (re-inject)	OpenDKIM **:8892** sign-only → OpenARC **:8893** (`hermes_openarc`)	Re-sign body that CipherMail mutated, then ARC-seal the **final** form so downstream verifiers trust the cumulative auth chain even after body modification (#229)
`:10027` (CipherMail GUI)	OpenDKIM **:8891**	Sign GUI-originated mail; no body mods on this path

Why OpenARC sits ONLY at `:10026` (and NOT in `main.cf`)

OpenARC's `ARC-Message-Signature` includes a hash of the message body. If ARC sealed at `:25`, the body would later be mutated by body_milter (`:8893`) and CipherMail (MIME rebuild), so the seal's body hash would be invalid by the time downstream verifiers received the message — `cv=fail`.

`:10026` is the only point where the body is in its **final** form (all body modifications + CipherMail MIME rebuild complete), so it's the only correct hop to apply the ARC seal. Adding ARC to `main.cf`'s default `smtpd_milters` would cause two problems:

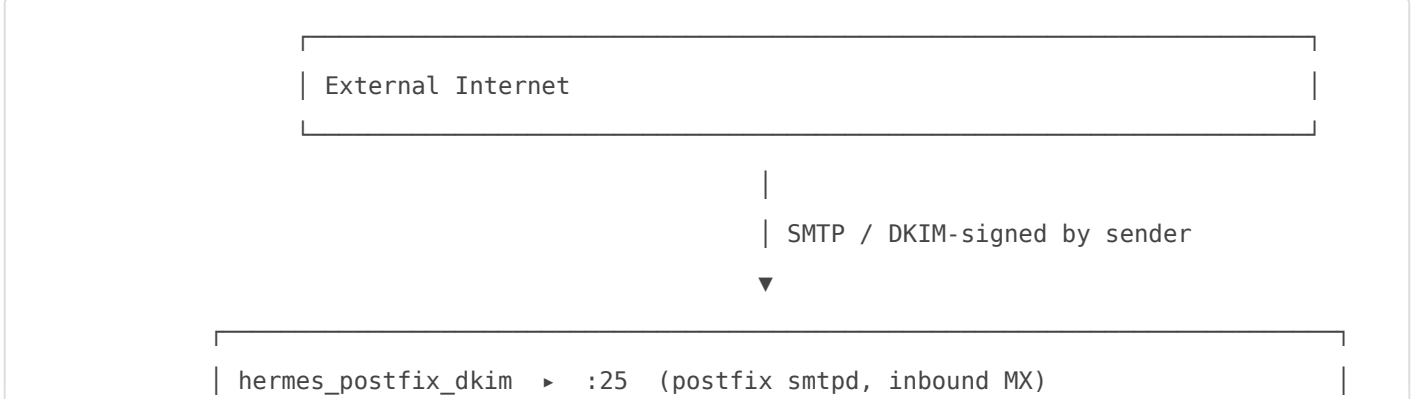
1. **Pre-modification sealing at `:25`** → broken seal at the recipient.
2. **Double-sealing**: mail going through `:25` → amavis → `:10026` would be sealed twice by the same gateway (`i=1` at `:25`, `i=2` at `:10026`), producing redundant chain bloat / verification ambiguity.

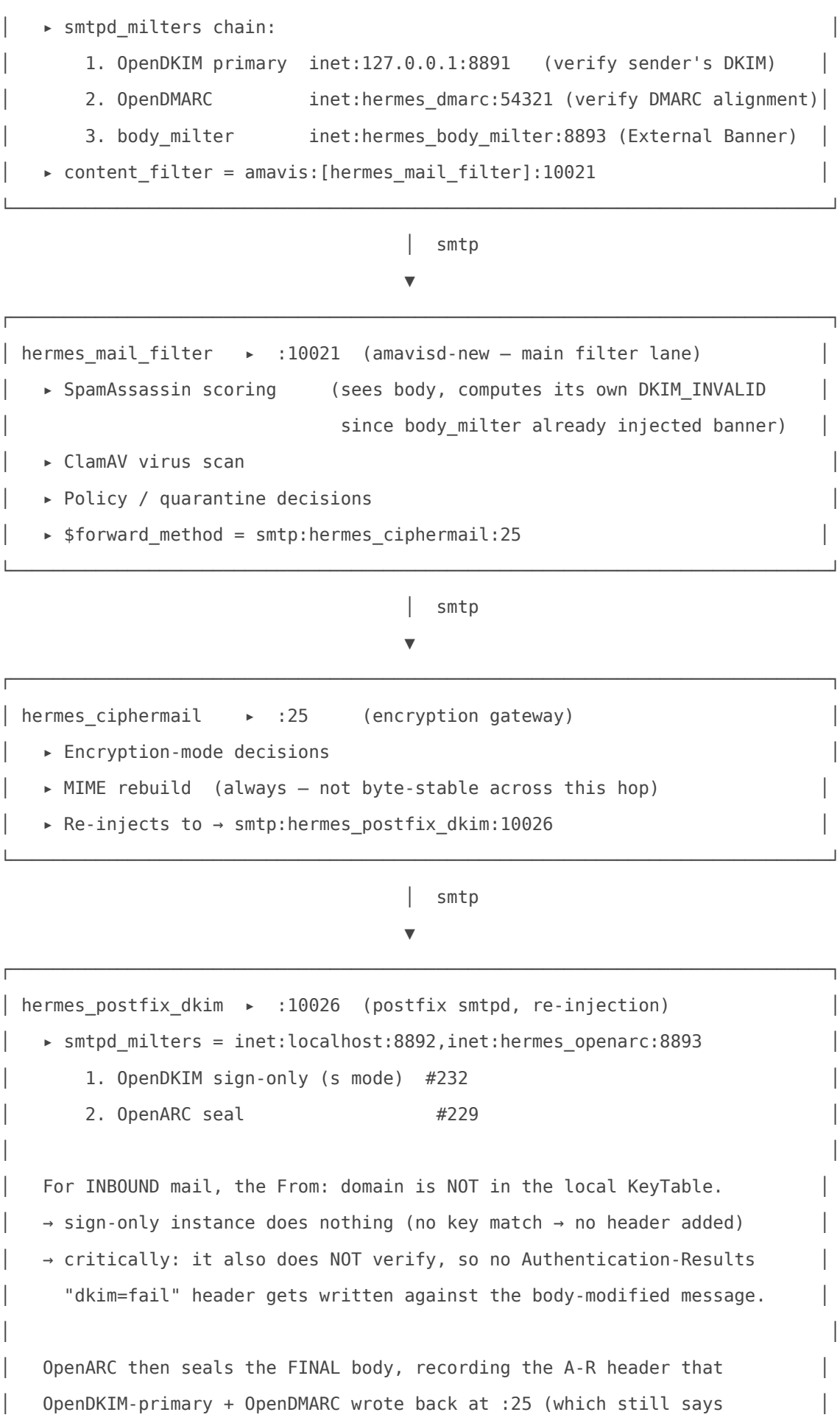
ARC stays out of `main.cf` deliberately. Master.cf `:10026` override is the single point of truth.

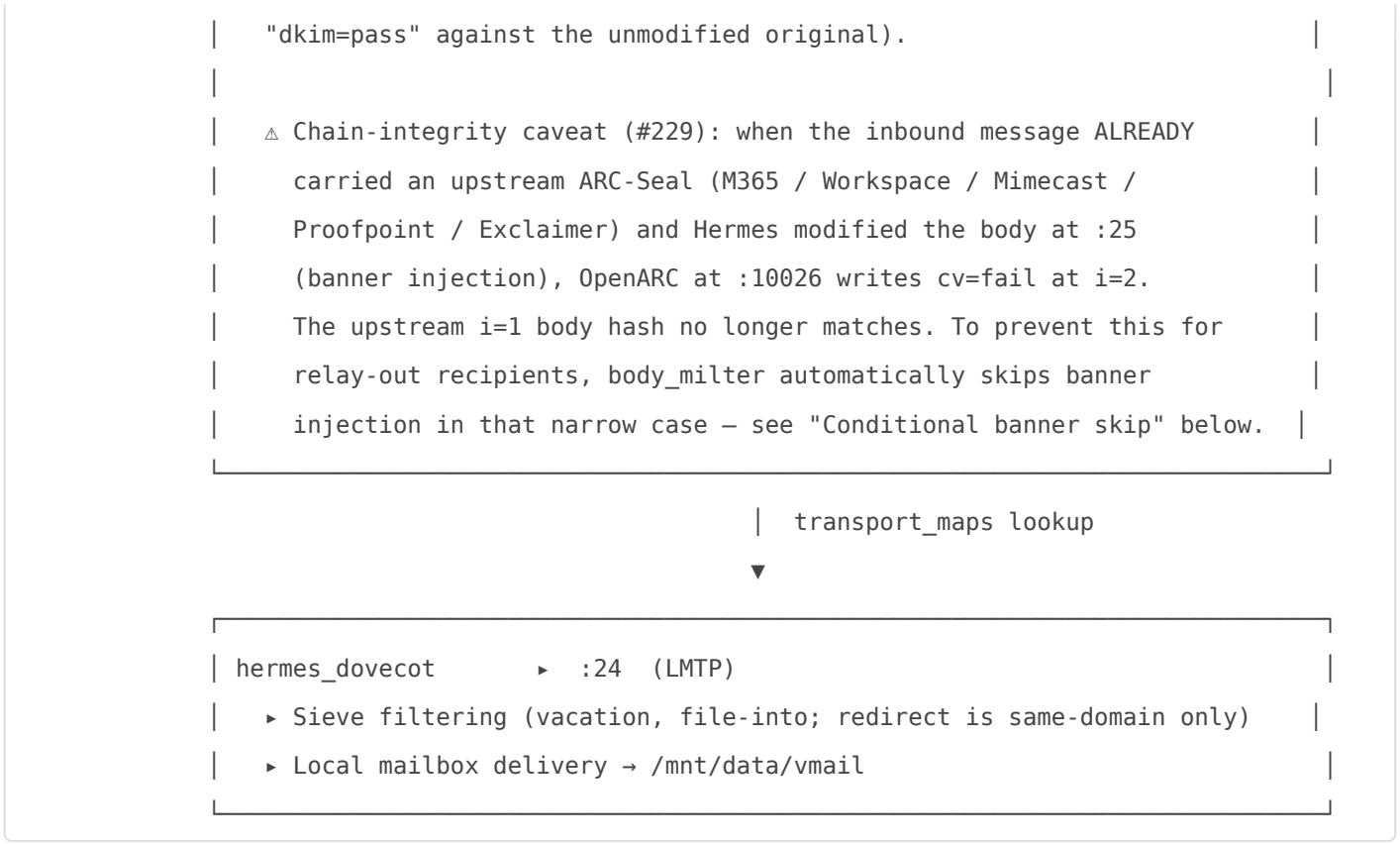
> The body_milter ordering is recorded in `parameters` as `order1=3.1` so it sits AFTER OpenDKIM signer (`0.5`) and OpenDMARC. The retro-fix `UPDATE` in `updates/hermes-260119/sql/schema_updates.sql` corrects existing installs that had `0.5` for body_milter (which placed body mods BEFORE signing, the root cause of #232 outbound DKIM failures).

Inbound flow

External MTA → local mailbox.




▸ smtpd_milters chain:
1. OpenDKIM primary inet:127.0.0.1:8891 (verify sender's DKIM)
2. OpenDMARC inet:hermes_dmarc:54321 (verify DMARC alignment)
3. body_milter inet:hermes_body_milter:8893 (External Banner)
▸ content_filter = amavis:[hermes_mail_filter]:10021
smtp
hermes_mail_filter ▸ :10021 (amavisd-new — main filter lane)
▸ SpamAssassin scoring (sees body, computes its own DKIM_INVALID
since body_milter already injected banner)
▸ ClamAV virus scan
▸ Policy / quarantine decisions
▸ $forward_method = smtp:hermes_ciphermail:25
smtp
hermes_ciphermail ▸ :25 (encryption gateway)
▸ Encryption-mode decisions
▸ MIME rebuild (always — not byte-stable across this hop)
▸ Re-injects to → smtp:hermes_postfix_dkim:10026
smtp
hermes_postfix_dkim ▸ :10026 (postfix smtpd, re-injection)
▸ smtpd_milters = inet:localhost:8892,inet:hermes_openarc:8893
1. OpenDKIM sign-only (s mode) #232
2. OpenARC seal #229
For INBOUND mail, the From: domain is NOT in the local KeyTable.
→ sign-only instance does nothing (no key match → no header added)
→ critically: it also does NOT verify, so no Authentication-Results
"dkim=fail" header gets written against the body-modified message.
OpenARC then seals the FINAL body, recording the A-R header that
OpenDKIM-primary + OpenDMARC wrote back at :25 (which still says



Inbound paths that diverge

- **Relay-only domains** (in `transport_maps`) skip Dovecot and `smtp:`-forward to the next-hop MX listed in `/etc/postfix/transport`.
- **Quarantined / virus / bad-header** mail is held by amavis at `:10021` (final destinies: `D_BOUNCE` virus/banned, `D_DISCARD` spam/bad_header per `50-user` config).
- **Whitelisted senders** bypass via `BYPASSALLCHECKS` at `:10030`.

Architectural principle: Hermes is the auth boundary (#229)

Hermes is the authoritative auth / security boundary for every domain it relays for. Inbound auth checks (DKIM, SPF, DMARC, ARC verify, spam, virus) happen at Hermes. Body modifications (External Sender Banner, disclaimer, signature insertion, encryption) also happen at Hermes. **Customer downstream mail servers (the relay-target MX) must be configured to trust Hermes implicitly:** allowlist Hermes by IP or hostname, accept forwarded mail without re-running DKIM / SPF / DMARC / ARC checks. This is the same deployment model Mimecast, Proofpoint, and Barracuda customers use — the SEG IS the trust boundary.

When the inbound message arrives carrying an upstream `ARC-Seal:` header (M365, Workspace, Mimecast, Proofpoint, Exclaimer, etc.) and Hermes modifies the body (banner, disclaimer), the

upstream chain's body hash is invalidated. OpenARC at `:10026` honestly records `cv=fail` on Hermes's own seal because it can no longer validate the upstream chain against the modified body. The original sender's `DKIM-Signature` body hash is also invalidated.

This is by design and is not a Hermes problem. A correctly-configured customer downstream MX is allowlisting Hermes and not re-checking auth on forwarded mail; the `cv=fail` and broken DKIM signal never gates delivery. If a customer's downstream MX is doing redundant auth checks on mail Hermes forwards, that's a misconfiguration on the customer's end — the fix is to allowlist Hermes there, not to silence Hermes here.

For external receivers Hermes does NOT have a trust relationship with (third-party MXes encountered via sieve redirect or alias forwarding, should those leak past the same-domain validation), the cv=fail and broken DKIM signals do reach a non-trusting receiver. That's why sieve redirects from the user portal are validated to require a same-domain target (see `inc/sieve_user_rule_actions.cfm`) — keeping forwarded mail inside Hermes's auth boundary. Aliases configured by admins are constrained to internal Hermes mailboxes by the existing CFML check in `inc/add_mailbox_alias_action.cfm`.

Why not lift the chain by stripping the upstream ARC?

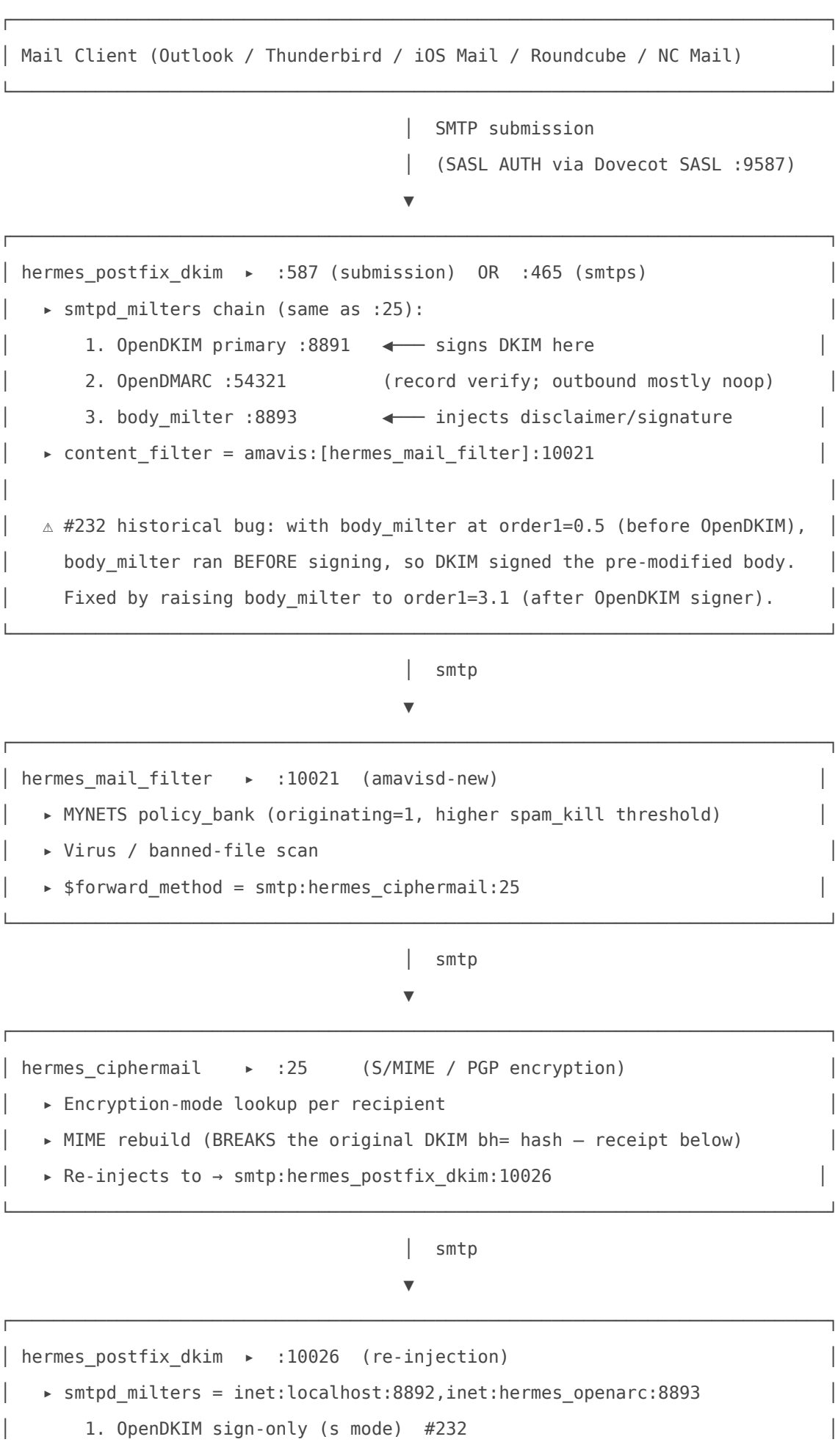
`cv=fail` is honest — Hermes correctly admits the chain we received no longer body-validates against the message we're about to send. The verifier walks the chain backward and recomputes the upstream `i=1` body hash against the **current** body, so stripping our `i=2` admission does not repair anything. The only mechanisms that could restore trust for body-modifying gateways are:

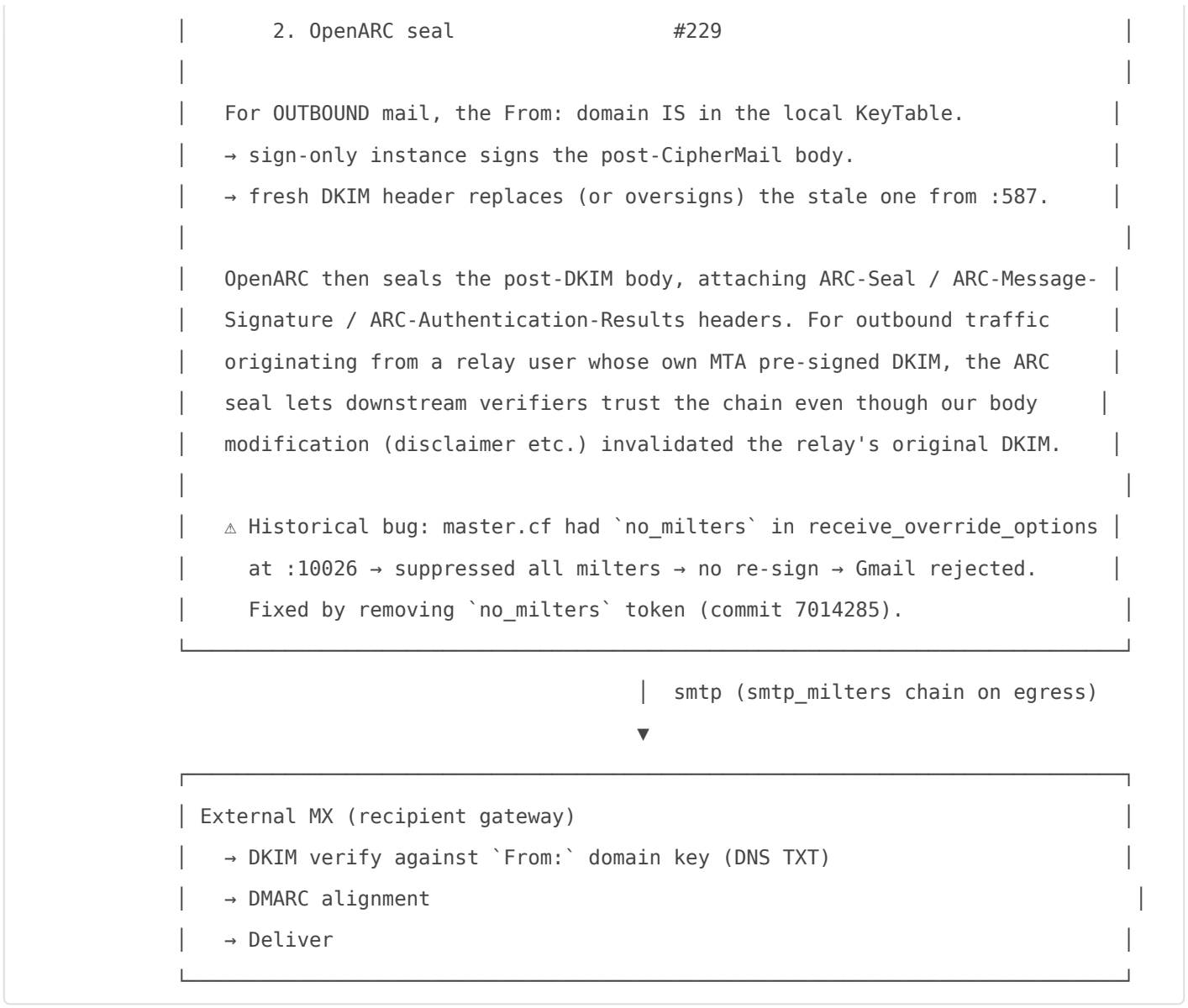
1. **Receiver-side trust configuration** — Microsoft 365's "Trusted ARC Sealers" feature, Gmail's internal trust list, etc. Useful when forwarding to receivers OTHER than the customer's own MX. See the Trusted ARC Sealers — M365 guide for cross-org scenarios.
2. **Don't modify the body** — defeats the purpose of having body modification features.

Multi-instance OpenARC (separate verify-only + sign-only daemons) does **not** help: OpenARC v1.3.0's sign-only mode re-validates the chain at sealing time and ignores AAR cached by the verify instance. This was empirically tested on DEV on 2026-05-14; see the closed #229 discussion.

Outbound flow

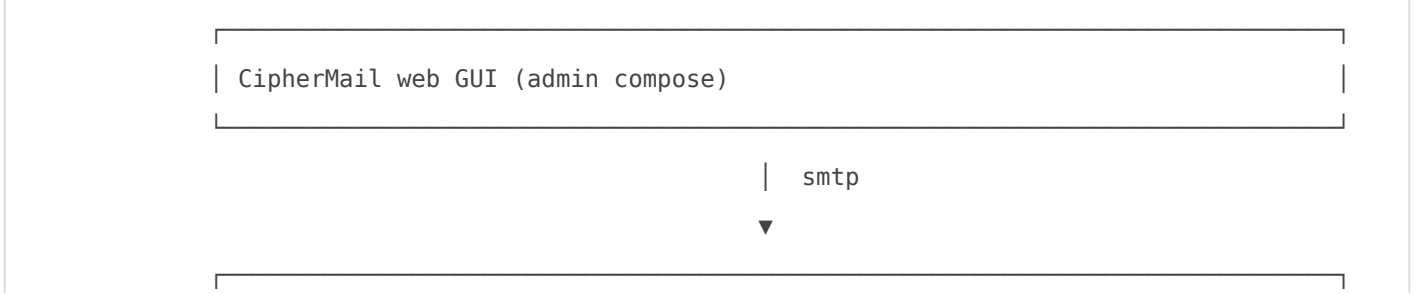
Authenticated local user → external recipient.

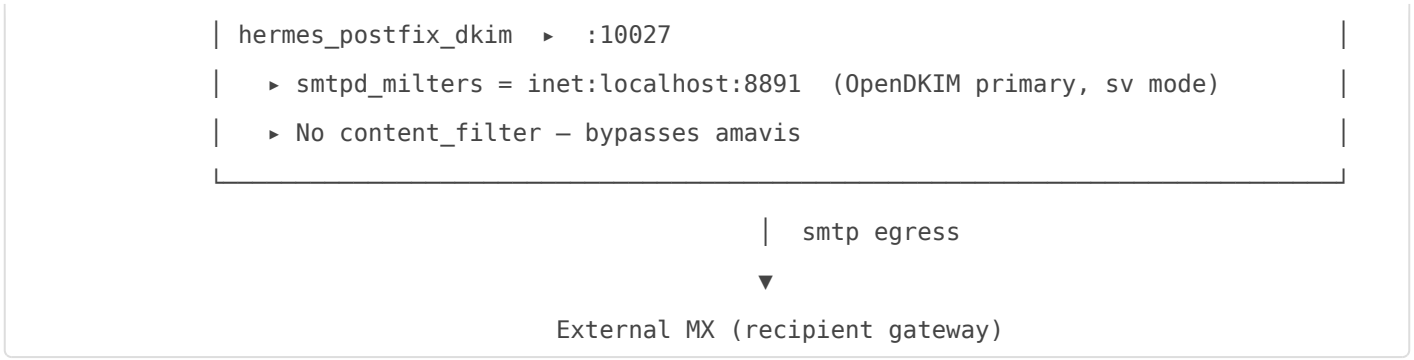

Mail Client (Outlook / Thunderbird / iOS Mail / Roundcube / NC Mail)
SMTP submission
(SASL AUTH via Dovecot SASL :9587)
hermes_postfix_dkim ▸ :587 (submission) OR :465 (smtps)
▸ smtpd_milters chain (same as :25):
1. OpenDKIM primary :8891 ◀── signs DKIM here
2. OpenDMARC :54321 (record verify; outbound mostly noop)
3. body_milter :8893 ◀── injects disclaimer/signature
▸ content_filter = amavis:[hermes_mail_filter]:10021
⚠ #232 historical bug: with body_milter at order1=0.5 (before OpenDKIM),
body_milter ran BEFORE signing, so DKIM signed the pre-modified body.
Fixed by raising body_milter to order1=3.1 (after OpenDKIM signer).
smtp
hermes_mail_filter ▸ :10021 (amavisd-new)
▸ MYNETS policy_bank (originating=1, higher spam_kill threshold)
▸ Virus / banned-file scan
▸ $forward_method = smtp:hermes_ciphermail:25
smtp
hermes_ciphermail ▸ :25 (S/MIME / PGP encryption)
▸ Encryption-mode lookup per recipient
▸ MIME rebuild (BREAKS the original DKIM bh= hash — receipt below)
▸ Re-injects to → smtp:hermes_postfix_dkim:10026
smtp
hermes_postfix_dkim ▸ :10026 (re-injection)
▸ smtpd_milters = inet:localhost:8892,inet:hermes_openarc:8893
1. OpenDKIM sign-only (s mode) #232



CipherMail web-GUI originated mail

When admins send mail directly from CipherMail's web GUI (rare), it enters postfix at `:10027`, bypasses amavis content filtering entirely, and is signed by the **primary** OpenDKIM (`:8891`) — not the sign-only instance — because this path doesn't traverse any body-modification hop and the standard sv-mode instance is appropriate.





Why two OpenDKIM instances? (#232 architecture decision)

A single OpenDKIM instance in sv mode (verify + sign) cannot satisfy both requirements at `:10026`:

Requirement	Default sv instance at :8891	Sign-only instance at :8892
Verify inbound at `:25`	✅ does this	❌ wouldn't (correctly)
Sign outbound at `:587` / `:465`	✅ does this	❌ wouldn't (correctly)
Sign outbound re-inject at `:10026` (post-CipherMail body rebuild)	❌ would sign	✅ signs
Skip verify on inbound re-inject at `:10026` (post-body-milter banner)	❌ verifies → `dkim=fail` Authentication-Results	✅ never verifies

OpenDKIM has no per-port mode override, and `InternalHosts` / `ExternalIgnoreList` control signing-vs-verification only by sender IP — not by smtpd inet service. Per OpenDKIM's own project guidance, running a second daemon with a different config file is the supported way to get differential behavior on a single host.

The sign-only instance ignores inbound mail naturally: its `KeyTable` and `SigningTable` only contain entries for local domains, so inbound mail (where the From: domain matches no local key) is a no-op pass-through — it neither signs nor adds Authentication-Results.

Receipts (forensic proof captured during #232 diagnosis)

Symptom	Evidence	Root cause
Gmail rejected outbound from `tina@getwithme.com`	bounced `.eml` had `bh=z0Xb...` in DKIM header but body hashed to `bh=vJCS...`	CipherMail MIME rebuild after `:587` DKIM sign; `:10026` had `no_milters` so no re-sign
Inbound `dkim=fail` only when External Banner enabled	Same gmail→tina test: banner-on `dkim=fail`, banner-off `dkim=pass`. CipherMail held constant.	body_milter modifying body before the next milter hop saw it; primary OpenDKIM at `:10026` re-verified the modified body
Spam score ~+1.3 on banner-injected inbound	`DKIM_INVALID=0.1` + `NML_ADSP_CUSTOM_MED=0.9` when banner on; `DKIM_VALID*=-0.3` when banner off	amavis runs its own SpamAssassin DKIM check on the post-body-milter body. **Not fixed by multi-instance OpenDKIM** — separate problem
Downstream forwarder DMARC fail	Recipient forwards to gmail; gmail re-verifies original sender DKIM against modified body → fail	Solved by ARC sealing (#229)

Open follow-ups that touch this flow

- **#228** External Sender Banner re-enable — blocked on this diagram's `:10026` sign-only wiring landing.
- **#229** ARC sealing at perimeter — required so downstream forwarders trust Hermes' original-sender verdict.
- **amavis SpamAssassin DKIM scoring** (separate issue, not yet filed) — options A/B/C documented in the #232 handoff doc.
- **Dead-weight config-file audit** — `Docker/postfix_dkim/config/`, `Docker/mail_filter/config/`, `Docker/opendmarc/config/` are shadowed by volume mounts at runtime; ~85 files to consolidate or relocate.
- **Install-template drift** — `config/hermes/opt/hermes/conf_files/master.cf` still has the pre-#232 `no_milters` token on `:10026` and `smtpd_milters=:8891` for `:10026`. Fresh installs would regress until this template is brought into line with the active runtime config.

Revision #13
Created 2026-05-31 13:01:45 UTC by Dino Edwards
Updated 2026-06-13 12:30:29 UTC by Dino Edwards