Storage Topology (5 tiers)

Storage Topology (5 tiers)

Hermes SEG splits storage into five independent tiers so each can live on the right kind of disk for its workload. Four are operator-chosen at install time; the fifth (Config) is implicit — chosen by where the operator `git clone`d the repo.

Tier	Default path	Contents	Storage profile
1. Config	install root (implicit)	Repo working tree, `config/` subtrees, install script, secrets in `config/hermes/opt/hermes/keys/`, `.env`, `.hermes_install_config`	Fast SSD, modest size — chosen by where the repo lives
2. Data	`/mnt/data` (`DATA_MOUNT`)	DBs (MariaDB, Authelia, OpenLDAP), Amavis runtime state, ClamAV signatures, Fangfrisch state, Lucee server home, sieve scripts, all service logs, OpenDMARC, Postfix queue	Fast SSD; sized for DB growth + log retention
3. Archive	`/mnt/archive` (`ARCHIVE_MOUNT`) — added in #260	Amavis quarantine archive	Cheap bulk; sized for retention policy × quarantine inflow
4. Vmail	`/mnt/vmail` (`VMAIL_MOUNT`)	Dovecot mailboxes	Cheap bulk; sized for users × quota
5. Nextcloud	`/mnt/files` (`FILES_MOUNT`)	Nextcloud app + user files + Nextcloud's Redis cache	Cheap bulk; sized for user file storage

Each tier is **one host path**; the install script lays out the canonical sub-directory structure underneath it.

Why split storage

Tier	Why it gets its own disk

Config	Frequent reads (every container start); small footprint; lives with the install script + version control
Data	High write rate (logs + DBs); benefits from fast SSD; backup hot spot
Archive	Grows unboundedly with retention policy; cold reads (admin browses quarantine occasionally); cheaper bulk storage; backup cadence independent of Data
Vmail	Grows linearly with user count × quota; cheaper bulk storage; separate backup cadence (often less frequent than Data)
Nextcloud	Same growth characteristics as Vmail but different access pattern; often shared across multiple Hermes installs in larger deployments

Smaller deployments can collapse tiers — point Archive, Vmail, and Nextcloud at the same path as Data for a single-disk install. Each tier is its own prompt so the operator picks per workload.

Canonical sub-directory layout

Tier 2 — Data (default `/mnt/data/`)

Sub-path	Named volume	Service
`dbase/`	`db_data`	MariaDB
`authelia/db/`	`authelia_db`	Authelia state DB
`authelia/logs/`	`authelia_logs`	Authelia logs
`authelia/redis/`	`authelia_redis`	Authelia Redis
`commandbox/serverhome/`	`commandbox_serverhome`	Lucee server home
`dmarc/logs/`	`dmarc_logs`	OpenDMARC logs
`dovecot/logs/`	`dovecot_logs`	Dovecot service logs
`dovecot/sieve/`	`dovecot_sieve`	Sieve scripts (shared by commandbox + dovecot)
`ldap/data/`	`ldap_data`	OpenLDAP data
`ldap/logs/`	`ldap_logs`	OpenLDAP logs
`mail_filter/data/amavis/`	`mail_filter_data_amavis`	Amavis runtime state (PID files, scan tmp dirs — small, latency-sensitive)
`mail_filter/data/clamav/`	`mail_filter_data_clamav`	ClamAV signatures

Sub-path	Named volume	Service
`mail_filter/data/fangfrisch/`	`mail_filter_data_fangfrisch`	Fangfrisch state
`mail_filter/logs/`	`mail_filter_logs`	Mail filter logs
`nginx/logs/`	`nginx_logs`	Nginx logs
`openarc/logs/`	`openarc_logs`	OpenARC logs
`postfix_dkim/logs/`	`postfix_dkim_logs`	Postfix logs
`postfix_dkim/queue/`	`postfix_dkim_queue`	Postfix mail queue

Tier 3 — Archive (default `/mnt/archive/`) — added in #260

Sub-path	Named volume	Service
`amavis/`	`amavis_data`	Amavis quarantine archive (admin-browsable, grows with retention)

Note: the Amavis runtime state (`mail_filter/data/amavis/` named volume `mail_filter_data_amavis`) stays on the Data tier — it's small, doesn't grow with retention, and benefits from fast SSD latency. Only the quarantine archive moved.

Tier 4 — Vmail (default `/mnt/vmail/`)

Sub-path	Named volume	Service
`dovecot/`	`dovecot_mail`	Dovecot mailboxes

Tier 5 — Nextcloud (default `/mnt/files/`)

Sub-path	Named volume	Service
`app/`	`nextcloud`	NC app + user files
`redis/`	`nextcloud_redis`	NC's Redis cache

How it works at install time

1. `prompt_mount_points()` asks the operator for four paths (Data / Archive / Vmail / Nextcloud) — Config is already chosen by where the repo lives. Defaults `/mnt/data`, `/mnt/archive`, `/mnt/vmail`, `/mnt/files`. Choices saved to `.hermes_install_config` at the install root.
2. `provision_mount_dirs()` creates the entire sub-directory layout under each chosen path with the correct UID/GID for the containers that will write to them. Critical: bind-mounted volumes (`type: none, o: bind` in `docker-compose.yml`) require the source directory to **pre-exist** — Docker refuses to start the container otherwise.
3. `generate_compose_override()` writes the four mount-point variables (`DATA_MOUNT` / `ARCHIVE_MOUNT` / `VMAIL_MOUNT` / `FILES_MOUNT`) to `.env` at the install root. `docker-compose.yml` references these variables directly in its `device:` lines (e.g. `device: ${ARCHIVE_MOUNT}/amavis`) — Docker Compose substitutes at runtime. The legacy `docker-compose.override.yml` approach was retired in #179; the function name was kept for backwards-compatibility with the `--generate-override` CLI flag.
4. All four mount points are **required**. Empty values would resolve to dangerous relative paths during `device:` substitution (e.g. empty `${ARCHIVE_MOUNT}/amavis` → `/amavis` at the host root). For single-disk installs, point all four prompts at the same path.

Self-locating scripts

`install_hermes_docker.sh`, `rotate_db_credentials.sh`, and any other Hermes script needing the install root use a **walk-up self-locator** pattern that finds `docker-compose.yml` by walking up from `BASH_SOURCE[0]`:

```
SCRIPT_DIR="$(cd "$(dirname "${BASH_SOURCE[0]}")" && pwd)"
if [[ -z "${HERMES_ROOT:-}" ]]; then
    HERMES_ROOT="$SCRIPT_DIR"
    while [[ "$HERMES_ROOT" != "/" ]] && [[ ! -f "$HERMES_ROOT/docker-compose.yml" ]]; do
        HERMES_ROOT="$(dirname "$HERMES_ROOT")"
    done
    if [[ "$HERMES_ROOT" == "/" ]]; then
        echo "ERROR: Could not locate docker-compose.yml in any parent of $SCRIPT_DIR" >&2
        echo "Set HERMES_ROOT environment variable manually and retry." >&2
        exit 1
    fi
fi
```

This is depth-independent (works at 1 level or 5 levels deep in the tree) and survives the script being relocated. **Do not use a hardcoded `dirname/..` chain** — it depends on the script's exact depth and breaks silently if the script moves.

Reading topology at runtime

`.hermes_install_config` is the source of truth for which paths the operator chose. Scripts that need this (`system_backup.sh`, `system_restore.sh`) source the file via the `load_config()` helper. Format:

```
DATA_MOUNT=/mnt/data
ARCHIVE_MOUNT=/mnt/archive
VMAIL_MOUNT=/mnt/vmail
FILES_MOUNT=/mnt/files
ENABLE_NEXTCLOUD=true
```

The file lives in the Config tier (install root), so it's part of every Config-tier backup automatically.

Revision #7
Created 2026-05-31 13:01:45 UTC by Dino Edwards
Updated 2026-06-13 12:29:55 UTC by Dino Edwards