

Legend

Green household data (apps, files, photos)

Blue backup copies, one shade per tier

Orange operator infrastructure

Purple keys and secrets (names only)

Grey outside services

Red outline single point of failure / gap

solid = data traffic

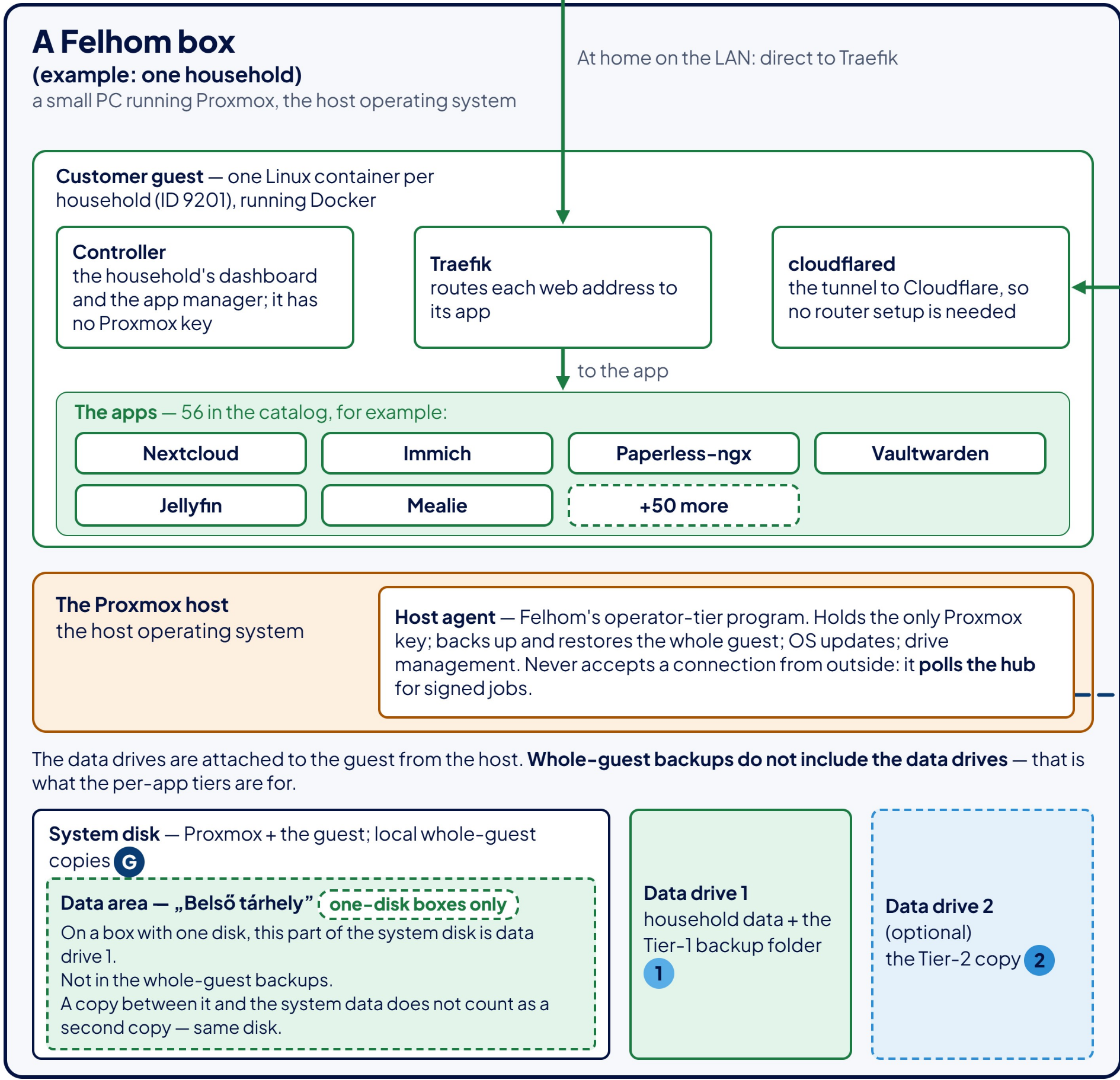
dashed = backup copy (with its time)

dotted = control (reports, jobs)

1 · The household's home

Remote access, HTTPS ends at Cloudflare's edge, so Cloudflare could technically see this traffic (later item: our own relay).

https only — http:// is sent to https:// before any app answers.



2 · The internet

Cloudflare

- DNS for the customer domains and for felhom.eu
- the tunnels
- the geo-block: only Hungarian visitors
- flood protection

The hub holds one Cloudflare key per customer zone and checks that it reaches only that customer's own domain.

felhom.eu
public website, served from DooPlex, Cloudflare DNS only

iso.felhom.eu
households download the installer

web pages

via the tunnel

Whole-guest backup — encrypted on the box, sent through WireGuard —

weekly, 04:30–08:30

per-app off-site (restic)

nightly 04:15, encrypted on the box

3 · The operator's home

DooPlex — the operator's home server
a small Kubernetes cluster (k3s) plus host services

Protected — everything else can be rebuilt from it

Gitea
all the code (10 repositories) and the container registry (the images the boxes run); the CI runner runs here too

Vaultwarden
the operator's password manager

Prometheus + Alertmanager
the operator's alarms

The hub
customer records · box reports · the job queue · alarms · the escrow of household keys · the System page (OS updates, kernel approvals)

The signing keys
sign every agent update and config bundle a box accepts

The website (felhom.eu) and the contact form's mailer

Local backups
several restic sets on DooPlex's own disks — **all on the same machine**

ep0-copy
a nightly pull of the whole ep0 datastore back to DooPlex — ciphertext only; DooPlex cannot read households' copies

Left out of the DooPlex copy on purpose: the container registry (27.7 GB) — rebuilt from the code

Test machines — disposable

demo-hp (a mini PC) — ring 0: gets OS updates, Docker and kernels first. Also hosts the **scratch guest 9202** (no hub link, for tests) and the **Tester 1 VM** (a whole Felhom box inside a VM)

demo-felhom (Intel N100 mini PC) — ring 0

The test bench — a throwaway guest where app updates are tested before any box gets them

Tester 2 — the first outside tester's laptop, at his own home: a second, smaller Felhom box. **Real household, often off.**

Where a household's data is copied

Tier	What it copies	Where	When	Kept	Locked by
1 Tier 1 — recovery unit	each app; its settings, database dump, volumes	data drive 1, on the box	nightly at 02:30 (W)	1 point per app	not encrypted (it never leaves the home)
2 Tier 2 — cross-drive	a mirror of Tier 1, plus the app's files	data drive 2, on the box — counted only on a different physical disk; in a VM made, not counted	nightly 03:30	mirror	not encrypted
3 Tier 3 — off-site, per app	Tier 1 + the app's must-keep files	Hetzner Storage Box (Germany), one folder per household	nightly 04:15	7 daily, 4 weekly, 6 monthly	the household's own restic password
Whole guest — local	the guest's system (not the data drives)	the box's own system disk	daily, in the window 04:30–08:30	one copy per day, up to 3 days (fewer on a small disk)	not encrypted
Whole guest — off-site	the same	ep0 (Hetzner, Germany), one namespace per household	weekly, same window	one per week, 2 weeks	the household's own key, escrowed with a recovery code

W = the household's backup window start, 02:30 by default. Tiers 1–3 follow at fixed offsets so they cannot run out of order. A missed night is caught up 15 minutes after the box starts.

Where the operator's data is copied

00:20 · DooPlex copy → ep0 (operator): the code (614 MB), Gitea's database and settings, the **password manager**, the nightly secrets export, the **signing keys**. Encrypted on DooPlex with the **DooPlex off-site key**; the key that writes it **cannot delete** old copies. Tested back every Sunday.

02:30 · Hub database → ep0 (operator): encrypted with the **hub-DB key**. Tested back every Sunday at 04:30.

05:00 · ep0-copy: ep0 → DooPlex a second copy of ep0 at home. Keeps 8 weekly copies. A daily job removes a deleted customer's copy within 30 days.

Not copied: the container registry (27.7 GB) — rebuilt from the code.

Local backups on DooPlex's own disks: all on the same machine.

Who tells a box what to do

The **hub** queues a job → The **operator** signs it → The **agent** checks the signature, then acts

Updates, in rings

Ring 0: the demo boxes, first → Operator approves on the **System page** → **Ring 1:** the customers

Apps: each new version is tested on the bench and the scratch box first; boxes update at night, after a fresh backup.

Debian security fixes: nightly, after the whole-guest backup.

Proxmox programs, Docker, the kernel: slow lanes, one approved set at a time.

Kernel: the household gets a mail the day before; the box restarts at night; a crash falls back to the old kernel by itself; a freeze needs someone to unplug the box.

Hub buttons — no signature needed, a fixed safe list: off-site backup now · run a check now · stop or extend a deletion countdown.

If DooPlex is lost

Survives (off DooPlex)

- ep0: the DooPlex copy and the hub database, plus every household's whole-guest copies
- the **Hetzner Storage Box**: households' per-app copies
- the **boxes themselves**: they keep running and backing up, but cannot report or update until the hub is back
- the operator's workstation

Lost and not in any copy

- the container registry (rebuild from code)
- DooPlex's own SSH key
- local build files
- Prometheus history
- other homelab apps

The way back

- New machine
- Reach ep0
- Open the DooPlex copy with the **paper key**
- Password manager first
- Gitea → rebuild images
- Hub (mail held)
- Point DNS at the new home → boxes reconnect by themselves

Keys that must exist on paper, away from home

- DooPlex off-site key** — opens the DooPlex copy — the key to everything else
- Hub-DB off-site key** — opens the hub database copy
- Hub seal key** — opens the console passwords inside the hub
- DooPlex backup passphrase** — opens the secrets export
- Signing keys (recovery + operational)** — let the operator update boxes again
- ep0 read token** — reads the copies
- Hetzner login** — reaches ep0 if DooPlex is gone
- Cloudflare login** — moves DNS to the new home

The password-manager master password — memorised, never written on the sheet

A key kept only in the password manager is not off DooPlex — the password manager runs on DooPlex.

Known gaps

Cloudflare can read remote-access traffic (HTTPS ends at its edge). Later item: our own relay.

DooPlex's local backups all sit on one machine; only the DooPlex copy and the hub database leave it.

One night, hour by hour

