Roadmap

This is the **order of delivery**. Detailed designs, rejected alternatives, and long-form decisions stay in `docs/TODO.md`. When a milestone finishes, update the documents listed under **Docs gate** so tracked docs match the device and companion again – code is ground truth until that catch-up lands in the same change set.

Product rule (unchanged). The device display is the approval surface. The companion may show, configure, and diagnose; it never approves. A compromised host must be able to lie about everything the companion shows and still be unable to authorise anything.

Milestone map

#	Milestone	Hardware	Primary surface
0	Companion UI/UX foundation	None	Companion
1	Companion features on today's firmware	None	Companion
2	Firmware features (GPIO2 only), including HOTP	None (existing board + confirm button)	Firmware
3	Companion catch-up for milestone 2	None	Companion
4	microSD on the device	Card slot / bulk USB	Firmware
5	Companion + SD safe	None beyond 4	Companion
6	RTC on the device	Fitted RTC + cell	Firmware / board
7	Firmware with RTC: TOTP + operation journal	Uses 6	Firmware
8	Companion catch-up for milestone 7	None beyond 6–7	Companion
9	Fingerprint module	Expansion port + sensor	Firmware (+ later companion UV UX)

Commercial / identity work (trademark, entity, AAGUID, VID/PID, attestation, FIDO certification) is **parallel**, not a numbered milestone – see `docs/TODO.md` Product identity. Board-family display variants are also parallel once `ui::Panel` is exercised on a second controller.

Milestone 0 – Companion UI/UX foundation

Status: done (shell and patterns locked in `internal/frontend/web/` and `docs/backend/companion-ui.md`).

Goal. Lock the companion's visual and interaction system **before** packing it with features. There is little surface today (Passkeys + Device); that is exactly when structure, tokens, and patterns are cheap to get right and expensive to retrofit later.

Why now. Every later companion milestone (1, 3, 5, 8) adds panels, tables, forms, and empty states. Doing those against a settled shell avoids a second pass that rewrites markup and CSS under live features.

Landed:

- Device-aligned CSS tokens; IBM Plex Sans/Mono; shared workspace width for status rail and panels; pixel `cP` favicon; brand tagline **sign what you see** and footer claim **No blind signing. No black boxes.**
- Multi-device: list + select API; one device auto-selected (read-only identity); two or more require an explicit choice.
- Passkeys Resident / Site-side subnav with cursor paging; Device Overview / PIN / Storage / Security / Erase subnav; Rescan on the status rail; WaitingOnDevice modal without host Approve.
- Compact unlock / Change PIN / Erase form grid; edge-aware `data-tip` popups; first-field focus on view and Device panel changes.

Out of scope here (and deferred correctly): new management commands, authenticatorConfig UI, rename flows, Display / screensaver configuration. Those are milestone 1+.

Docs gate: `docs/backend/companion-ui.md`, `docs/backend/csipass.md`, `docs/api.md`, this roadmap status. No device `user-manual.md` unless a shared string moves.

Milestone 1 – Companion features on today's firmware

Status: done for the agreed companion surfaces on today's firmware (Display get/set, PIN policy, rename, erase/recovery copy). Extra screensaver presets and carousel remain later milestones; idle CPU frequency presets are already on the Display panel.

Goal. Finish what the companion can already do against the firmware that ships now – **on top of the milestone 0 shell**: policies the device already enforces, credential metadata the wire already carries, honest recovery UX, and **owner Display preferences the firmware already stores in NVS** but does not yet expose on the management wire.

Work (from TODO companion Phases 2–3 and Display settings):

- Expose `authenticatorConfig` from the companion: `alwaysUv`, `setMinPINLength`, RP allow-list for min PIN length, `forcePINChange`.
- Credential rename / update user info where CTAP credMgmt allows it; still confirm on device when destructive. **Never** create or import a FIDO credential from the companion.
- Guided erase that walks the owner through unplug for CTAP `authenticatorReset`'s ten-second window (management erase already waits on a press).
- Connection / recovery copy placeholders that will match a future bootloader path (wording only until milestone 2–3 deliver recovery).

- **Device Display panel** (management get/set for `device_settings`, on-device confirm for writes):
  - Screensaver: **None**, **csiPass Logo** (wandering wordmark), **Breathing Dot**, **Cipher Text**, **Lissajous**, **Starfield**, or **Floating Clock** (uptime until RTC). **CPU while idle** dropdown (80 / 160 / 240 MHz). Image carousel waits for a later milestone.
  - Screensaver idle delay (respect firmware floor: off, or ≥10 s).
  - Brightness while the screensaver runs (panel).
  - Status-LED brightness while the screensaver runs.
  - Maximum / normal panel brightness (when not in screensaver).
  - Maximum / normal status-LED brightness (when not in screensaver).
- Use milestone 0 patterns for any new panels; do not invent a second visual language.

Depends on: milestone 0 (done).

Docs gate: `docs/api.md`, `docs/protocol-management.md`, `docs/backend/csipass.md`, `docs/backend/companion-ui.md` (Display panel), companion sections of `docs/TODO.md` (mark done). Device `user-manual.md` if a device-visible string or owner-facing Display behaviour is named there.

Milestone 2 – Firmware features without new hardware (incl. HOTP)

Goal. Everything that fits the current Waveshare board and the dedicated confirm button (GPIO2). No SD, no RTC, no fingerprint satellite.

Work:

- **HOTP (OATH)** on the trusted display: enrol via management (companion in milestone 3), codes shown on device after a fresh gesture; secrets in a dedicated OATH vault band (256 slots after the registration journal, packed seven per sector; HOTP/TOTP share one record and differ by a kind byte — **done this slice** with `ListOath` / `EnrolOath` / `DeleteOath`). No clock required for HOTP.
- Signed recovery image + authenticated update transport; repartition for a permanent recovery slot (layout already decided in TODO Firmware).
- U2F / CTAP1 verification with the elevated HID client (`deploy/testing/u2f-check.py`); browser U2F on Windows is not the path. Hardware run still pending when a device is present.
- Remaining CTAP 2.1 / 2.2 / 2.3 items that need no new silicon (see TODO CTAP tables); skip what was **decided against** (`largeBlobs`, keyboard OTP, …). **Done this slice:** GetInfo `attestationFormats` = `none`, `maxPINLength` =63 (`maxRPIDsForSetMinPINLength` =8 was already advertised).
- Index-tag lookups when / if the RAM collection stops fitting; other firmware polish already listed.
- **Screensaver depth (beyond the built-in presets already on glass):**
  - Built-in effects through Floating Clock (ids 0–6) already ship; companion names the effect id only. Remaining depth is carousel and RTC wall time for Floating Clock.
  - **Carousel:** up to **five** owner images. Companion decodes, scales, and converts to the panel's raw frame format; firmware accepts size-checked buffers only (no image decoder on device). Store outside `secure_store` (planned `reserved` region). Upload is a

management op with on-device confirm. Absolute rule unchanged: a pending confirmation always wins over idle / carousel.
    - **CPU frequency while the screensaver runs — done** as NVS / management / Display dropdown presets 80 / 160 / 240 MHz (USB floor 80). Remaining depth is carousel and RTC wall time for Floating Clock.
    - Optional: dwell / transition knobs for carousel once the second frame buffer exists (see TODO Board family).
    - Optional: swap Floating Clock from uptime to HH:MM:SS when the RTC module in milestone 6 is present and powered.
- Optional: raise vault record payload ceiling early only if HOTP/OATH storage design needs it; otherwise keep the big raise for milestone 9 templates.

Not in this milestone: TOTP (needs trustworthy time → milestone 7), event log wall-clock provenance that pretends to be an RTC, SD, fingerprint. Companion carousel upload UI lands in milestone 3 unless a thin read-only GetInfo field is enough for M2 bring-up. Built-in effect ids through Floating Clock are already named on the Display panel.

Docs gate: `docs/hardware/features.md`, `docs/conformance.md`, `docs/security.md`, `docs/user-manual.md`, `docs/api.md` / `docs/protocol-management.md` for new management commands, regenerate `docs/images/screens/` for any new device screens, `docs/BUILD_JOURNAL.md`, `docs/hardware/test-protocol.md` (+ a run when HIL is done).

Milestone 3 – Companion catch-up for milestone 2

Goal. Companion surfaces for everything milestone 2 put on the wire and on the glass.

Work:

- OATH enrol / list / rename / delete (HOTP first): `otpauth://` under PIN + on-device gesture; codes stay on the device by default.
- Firmware upgrade UI: version pick, upgrade, application vs recovery indicator (depends on signed update from milestone 2).
- Any new posture / capacity / config fields milestone 2 added.
- **Display catch-up:** carousel slot manage (add / replace / clear up to five images) with host-side convert. Screensaver presets and idle CPU dropdown already live on the Display panel.

Docs gate: `docs/api.md`, `docs/backend/csipass.md`, `docs/protocol-management.md`, `docs/backend/companion-ui.md`. Keep device `user-manual.md` authoritative for on-glass OATH; companion docs describe enrol/list and Display upload only.

Milestone 4 – microSD on the device

Goal. Boards with a card slot expose an **encrypted file safe** on the device: names and contents sealed under the vault root; host never sees keys.

Work: detailed design already in TODO (“The file store, in detail”):

- Bulk USB interface (WinUSB / libusb), separate from management HID.
- On-device encrypt/authenticate; reject a card edited off-device.
- No mass-storage class; no CDC-ACM as the file pipe.

Open (record, do not silently decide): whether classical SSH private keys, or only owner documents / recovery material / wrapped blobs, belong in the safe – see Open questions below.

Docs gate: `docs/hardware/features.md`, `docs/hardware/spec/` (slot/pinout), `docs/security.md` (card-pulled threat), `docs/api.md` (bulk contract), `docs/user-manual.md` (owner-facing safe behaviour).

Milestone 5 – Companion + SD

Goal. Companion is a browser over the safe: list, upload, download, delete – always after PIN proof; never holds plaintext keys.

Docs gate: `docs/api.md`, `docs/backend/csipass.md`, companion UX notes in TODO / roadmap status. Device manual only for gestures the glass shows during file ops.

Milestone 6 – RTC on the device

Goal. Optional fitted RTC (e.g. DS3231 + cell) as a board variant; one firmware image probes for it.

Work: hardware bring-up, dead-cell detection, NVS "RTC last set from", management/time commands as already sketched in TODO. RTC time **outranks** host-asserted time when the cell is alive.

Docs gate: `docs/hardware/spec/`, `docs/hardware/features.md`, `docs/security.md` (time provenance), `docs/protocol-management.md`.

Milestone 7 – Firmware with RTC: TOTP + operation journal

Goal. Features that only become honest once milestone 6 exists.

Work:

- **TOTP** on the trusted display (same posture as HOTP: enrol from host, code on glass, confirm per code). Absent and explicit when no RTC / dead cell – never fake a clock.
- **Operation / event journal** (append-only, encrypted, provenance flags) as designed in TODO; off by default; turning off erases.
- Do not present TOTP as equal to a passkey (product rule in TODO).

Docs gate: `docs/hardware/features.md`, `docs/user-manual.md`, `docs/security.md`, `docs/conformance.md` if new CTAP/OATH claims, screenshots, test-protocol.

Milestone 8 – Companion catch-up for milestone 7

Goal. Enrol/list TOTP, configure and read the event journal (intervals and provenance, not forged absolute certainty), device-time line in the UI when a real source exists – never a fabricated clock.

Docs gate: `docs/api.md`, `docs/backend/csipass.md`, `docs/protocol-management.md`.

Milestone 9 – Fingerprint module

Goal. Expansion-port satellite + raw sensor; match on the main board next to the vault root (see `docs/hardware/expansion-port.md`). Fingerprint is a **user-verification** path for FIDO, not a second product.

Work:

- Raise vault record payload ceiling for templates; built-in UV verifier interface; enrolment UX on the trusted display.
- Measure FAR/FRR on hardware; never store raw images.
- Companion: enrol/manage fingers, show UV capability – still no host-side approve.

Docs gate: expansion-port (as-built), features, security, user-manual, conformance (UV), test-protocol / HIL.

Related open question: Windows Hello and OS login biometrics – see below. Do not conflate FIDO UV-with-fingerprint with being a Windows Biometric Framework / ESS sensor.

Docs gate (every milestone)

After each milestone, close the documentation gap before starting the next:

Always consider	Role
`docs/BUILD_JOURNAL.md`	What this build changed
`docs/TODO.md`	Mark done; leave rejected history
`docs/api.md` / `docs/protocol-management.md`	Wire contracts
`docs/hardware/features.md`	Device behaviour
`docs/user-manual.md`	Owner-visible device behaviour only
`docs/security.md`	Trust-boundary changes
`docs/conformance.md`	New or newly verified clauses
`docs/images/screens/`	Regenerate via `screenshot.ps1`
`docs/hardware/test-protocol.md` (+ runs)	What an operator must re-check
`CHANGELOG.md`	Only at public release

Do not put planned behaviour into features or the user manual: it stays in TODO / this roadmap until implemented.

Open questions

Windows Hello, and “login biometrics” elsewhere

Two different products are easy to confuse:

1. **FIDO2 security key** – what csiPass already is. Windows, browsers, and many IdPs already use it for web / Entra / “sign in with a security key”. Fingerprint on the device as **CTAP user verification** strengthens that path without any Hello driver stack.
2. **Windows Hello biometric provider** – fingerprint (or face) for *Windows sign-in* through the Windows Biometric Framework. Enhanced Sign-in Security (ESS) wants match-on-sensor hardware, vendor certificates, and a Microsoft- aligned peripheral story. Recent Windows 11 builds extend ESS to some *external* fingerprint readers, but only ones built for that program – not a raw FPC-class sensor whose templates live in our vault.

How realistic for csiPass?

- **Realistic and on-mission:** fingerprint as on-device UV for FIDO (milestone 9). That helps Windows Hello *security key* flows and every other FIDO RP.
- **Hard / parallel product:** shipping as a first-class Hello ESS fingerprint peripheral. Different certification, drivers, and likely a match-on-chip module – fights the “dumb satellite, match beside the vault” design.
- **Middle ground to study later:** whether a FIDO2 key with bio-UV is enough for the Windows login stories we care about (security key unlock / Entra) without ever joining WBF.

Linux: login with a FIDO key is already common (`pam_u2f` / `pam_fido2`). Local fingerprint is usually `fprintd` + a supported reader – again a different stack from CTAP UV.

macOS: third parties do not replace Touch ID for local unlock. FIDO security keys work in Safari/Chrome and some Apple Account flows; bio-UV on our device still helps those FIDO ceremonies, not “become Touch ID”.

Decision posture: pursue fingerprint as FIDO UV first (milestone 9). Treat Hello ESS / WBF as a separate go/no-go after that exists; do not bend the expansion port toward ESS match-on-sensor unless we explicitly open that product line.

SSH and the SD safe

Already true without SD: OpenSSH supports FIDO devices today (`ecdsa-sk` / `ed25519-sk`). ES256 alone is enough for `ecdsa-sk`; Ed25519-sk is preference, not a door locked until milestone 2 algorithms land.

What SD adds: a sealed place for *files* the owner chooses – recovery material, wrapped blobs, perhaps SSH *certificates* or config – still unlocked only with PIN + device policy. It must not become “mass storage of raw PEM private keys the host can copy,” or the safe collapses into a disk.

Under question for milestones 4–5:

- Allow storing owner-labelled secrets that the device will only release to a local companion session after PIN (still a host-visible secret after unlock – weaker than FIDO-sk).
- Prefer device-held signing keys (FIDO-sk or a future non-FIDO sign command) and let the SD hold only non-exportable wraps / certs / notes.
- Do **not** revive `largeBlobs` only for SSH-over-SSO unless a concrete consumer appears (already decided against in TODO).

Record a written decision before milestone 4 ships file-type policy.