Handz V1 PRD Bundle

CH29 — Data Storage & Limits

Purpose: Define exactly what data exists in Handz V1, where it is stored (device vs cloud), the storage/size limits (including the locked 2GB Pro media cap), and the enforcement + UX rules that prevent bugs, abuse, and user confusion. This chapter intentionally focuses on **product behavior** and **data rules**, while referencing implementation details only where necessary for zero guesswork.

Doc ID: HZ-V1-CH29_Data_Storage_and_Limits_R1
Revision: R1 (2026-01-02)
Status: Draft
Depends on: CH00, CH02, CH07, CH08, CH17, CH28, CH30
Related: CH09–CH16, CH20–CH24, CH31, CH34
Supersedes: —
Owned Decisions: Data storage model for V1; storage limits enforcement; media rules (upload vs link); in-app disclosure of storage limits; migration-ready abstraction.
Open Questions / Placeholders: See §10 (explicit list).

1. Scope

This chapter owns all V1 decisions about:

- Where each type of user data lives (device-only vs cloud).
- What is considered “shareable” vs “private-only.”
- Storage limits and how they are enforced (hard caps + soft warnings; see CH30 for the warning ladder).
- The exact behavior when a user hits limits (block, warn, upgrade, cleanup, or degrade gracefully).
- Media rules: **Pro uploads** are private-only and do not travel via share links; **links** are shareable (locked).
- Migration posture: keep an abstraction boundary so we can later switch providers with minimal rework (non-goal: perfect provider-agnostic design).

Out of scope (owned by other chapters): offline sync mechanics (CH28), share-link lifecycle + limits (CH17), import conflict resolution (CH19), abuse thresholds + warning ladder (CH30), and account deletion/export compliance (CH34).

2. Plan States & High-Level Storage Behavior

Plan states are defined in CH08. This chapter clarifies how storage behaves per state.

- **Guest:** Can browse demo content. Cannot save flows or moves locally or to cloud (locked). Any edits are treated as a **temporary session draft** that is discarded on app close or after a short inactivity timeout (see CH28 for offline/draft handling).
- **Free:** Has a cloud account. Can save up to **2 flows** (locked). Can receive imports into inbox up to **10** items (locked). Cannot upload media. Links can be attached. Practice is paywalled (see CH25/CH20).

- **Trial / Pro:** Full cloud storage. Pro may upload media up to **2GB total** across all private uploads (locked). Pro can share via links; uploads do not propagate via share links (locked).

See: HZ-V1-CH00 §5 (Decision Log & Locks).

3. Storage Domains

Handz stores data in two domains. Each data type is explicitly assigned to one domain or both.

Domain A — Cloud (Authoritative)

- User account profile, entitlements/plan state, and all saved user content (moves, flows, practice history, mastery plans, etc.).
- Share links and inbox imports.
- Any metadata required for multi-device continuity.

Domain B — Device (Ephemeral or Cached)

- Temporary drafts (e.g., a flow being edited) when offline or before first save.
- Cached read models for performance (flow thumbnails, computed path lists).
- Downloaded previews for share/import screens (to reduce repeated fetches).
- Local-only media files *prior* to upload (Pro) or when attached as a device-local reference (not shareable).

Important rule: whenever cloud and device disagree, cloud is authoritative except when CH28 defines an explicit offline edit merge path.

4. Data Types & Where They Live

This section defines each top-level data object and its storage/size considerations. Field-level schemas are owned by their feature chapters, but CH29 defines the storage rules and size accounting.

User Profile & Entitlements

- Stored in cloud. Contains: user ID, display name, username, createdAt, planState, trialStart/trialEnd, subscription status, and feature flags.
- Plan state must be cached on device for offline gating but is refreshed whenever network is available.
- No sensitive training content is stored here (that lives in moves/flows).

Moves (Default + Custom + Customizations)

- Default move catalog (shipping content) is packaged in-app and also mirrored in cloud for search/filter updates (owned by CH09/CH10).
- User custom moves are stored in cloud. Editing a move creates a new revision for revert (see CH11).
- Move alias/family references (teep vs push kick) are stored as canonical IDs (CH10).
- Technique descriptions are optional; V1 default move list ships with minimal/no technique description (locked in CH09/CH10 discussions).

Flows + Builder Graph

- Saved flows are cloud objects. Free cap = 2 flows (locked).
- Each flow owns a graph (nodes, edges) and references moves by canonical ID + optional override payloads (owned by CH12/CH13/CH19).
- Draft flows may exist only on device until saved; Guest cannot save drafts.

Sequences / Transition Details

- Sequence detail objects attach to edges (or edge+node pair depending on CH13) and are stored in cloud with the flow.
- These are high-text fields and can become large. Apply per-field length caps (placeholder; see §10).

Practice Logs / History

- Cloud stored for Free/Pro; used for streaks and progress metrics (CH22).
- Logs must be compact: store references to selected paths and session summary; avoid duplicating entire graphs per session.

Mastery / Maintenance Plans

- Cloud stored. These can reference multiple flows and paths (CH23/CH24).

- Maintenance schedules may trigger notifications (CH27).

Inbox Items (Imports)

- Inbox item is a cloud object referencing an incoming share payload or link snapshot (CH18/CH19).
- Free inbox cap = 10 items (locked).
- Free can view but cannot practice inbox items (locked). Practice requires saving to library and consuming credits on saved flows only (CH25/CH20).

Media Attachments

- Two types: **Links** (shareable) and **Uploads** (private-only).
- Uploads are Pro-only, counted toward 2GB cap, and never included in share payloads (locked).
- Links can be attached at move-level and/or flow-level (exact attachment points owned by CH11/CH16).

Exports

- Exports are generated on device (PDF/JSON/ZIP etc.) then optionally shared via iOS share sheet. Export formats and scopes are owned by CH34.

5. Media Rules (Uploads vs Links)

Media is the highest-risk area for storage, cost, and bugs. These rules are locked unless CH00 is revised.

5.1 Link Attachments (Shareable)

- A link is a URL string stored in cloud and treated as shareable content when a flow is shared via unlisted link.
- Links may be added by Free and Pro.
- If a link becomes invalid, the app shows a non-blocking warning (“Link unavailable”) and allows the user to edit/remove it.
- Links are included in import payloads and are visible to recipients according to the sender’s share settings (CH17/CH19).

5.2 Uploaded Media (Private-only)

- Uploads are Pro-only and count toward the user’s 2GB total cap (locked).
- Uploads are **private-only**: recipients of a shared flow do not receive uploaded media. They may receive link attachments if present.
- If a user attaches uploads to a flow and then shares it, the share screen must show a disclosure: **“Uploads won’t be shared. Add links if you want recipients to see media.”** (copy can be refined in CH15).

- Uploads must have per-file constraints (placeholder) and should be stored with metadata: file size, mime type, duration, createdAt, and an internal storage path.

5.3 Size Accounting

- Only uploaded media counts toward the 2GB cap. Text data (moves/flows/logs) is not counted in the user-facing “Media Storage” meter unless we later add a separate quota.
- Storage meter shows: used bytes, remaining bytes, and top contributors (largest files).
- Deleting an upload frees quota after backend confirmation. Until confirmed, show “Pending cleanup” state.

See: HZ-V1-CH30 (warnings and escalation).

6. Limit Enforcement UX (What the user sees)

This section defines deterministic behaviors for hitting storage-related limits. Messaging ladder is owned by CH30; this chapter defines triggers and required UI surfaces.

6.1 Pro Media Cap (2GB)

- When user attempts an upload that would exceed remaining quota, block the upload immediately (do not start upload).
- Show a blocking modal with: current usage, remaining storage, upload size estimate, and actions: **[Manage Storage] [Cancel] [Upgrade]** (Upgrade shown only if user is not Pro/trial).
- Manage Storage opens a dedicated screen listing uploads sorted by size with multi-select delete.
- If user deletes uploads, re-check quota before re-attempting the upload.

6.2 Free Flow Cap (2 saved flows)

- If Free user tries to save a new flow beyond cap, block save and show upgrade gate with two alternatives:
- Option A: Delete an existing saved flow to make room (with clear warning).
- Option B: Upgrade to Pro (7-day trial available per CH25).
- If the action originated from Inbox “Save to Library,” see §7.2.

6.3 Free Inbox Cap (10 items)

- If inbox is full, new imports are not accepted into inbox. The receiving flow link screen must show: **“Inbox full (10). Save or delete an item to accept new imports.”**
- Pro users may have a higher cap (placeholder; owner CH18/CH30).

6.4 Guest Restrictions

- Guest cannot save flows/moves. Any attempt to tap “Save” or “Export” triggers an account creation gate (CH07).
- Guest also cannot store uploads or links persistently. Links may be visible in demo content but cannot be edited/saved by guest.

7. Sharing & Imports: Storage-Safe Rules

This section prevents contradictions between sharing funnels and storage limits. It defines exactly what gets stored, where, and how Free users can participate without bypassing monetization.

7.1 Share Payload Composition

- A shared flow link resolves to a share payload containing: flow graph + referenced moves + any link attachments + any move/flow text notes allowed by share settings.
- Upload attachments are replaced with placeholders in the payload: {type: 'upload', shared: false}. Recipients see a disclosure: **“This flow contains private uploads that are not shared.”**

- Recipient may still import the flow, and the imported version will preserve the placeholder markers so the recipient understands that media is missing by design.

7.2 Inbox -> Library Save (Free cap interactions)

- When a Free user saves an inbox item to library, it counts toward the 2-flow cap.
- If at cap, the save flow is blocked and user is prompted to either delete one saved flow or upgrade. No silent overwrite.
- Free users can view inbox items without saving, but cannot practice them (locked).

7.3 Import does not bypass practice paywall

- Imports are allowed for Free to preserve viral funnel.
- However, practicing any imported content requires either: (a) save to library (subject to cap) and use monthly credits on saved flows, or (b) upgrade to Pro.
- Inbox items never consume practice credits directly (locked). Credits are only usable on saved flows.

8. Implementation Reference (Firebase-first, migration-ready)

You selected Firebase for V1 for speed. This section specifies a reference architecture in Firebase terms while keeping an abstraction boundary so migration later is feasible.

8.1 Reference Storage Components

- **Firestore**: primary database for user content (moves, flows, logs, inbox, share metadata).
- **Cloud Storage**: Pro media uploads (counted toward 2GB).
- **Authentication**: Apple/Google/Email providers (CH07).
- **Cloud Functions** (optional but recommended): secure share-link generation, quota checks, and cleanup tasks.

8.2 Abstraction Boundary

In code, implement a thin “StorageService” interface so the rest of the app calls methods like:

```
• saveFlow(flow) • fetchFlow(flowId) • uploadMedia(file) • listMedia() •
deleteMedia(mediaId) • estimateQuota()
```

This makes switching to Supabase or another backend later much easier: only StorageService changes, not the whole app.

See: HZ-V1-CH28 (offline cache strategy) and HZ-V1-CH34 (export/delete expectations).

9. Security & Privacy Considerations (Storage-specific)

- All user content is private by default. Sharing occurs only through unlisted links (CH17).
- Share link tokens must be unguessable and revocable; never use sequential IDs (CH17).
- Uploads are private-only: even if a recipient has the link, they cannot access the sender’s uploaded media.
- Rate-limit share link creation and imports to reduce scraping (CH30 owns thresholds).
- Do not log or store raw video frames or biometric data in V1.
- If a user deletes their account, all stored media and associated metadata must be deleted (CH34).

10. Placeholders / Decisions to Lock Later

These items must remain explicit placeholders until their owner chapter locks them. Do not guess in implementation without recording an assumption block (CH00 §8).

- PLACEHOLDER: **Max upload file size** • Owner: CH29 • Options: 50MB / 100MB / 250MB • Default: 100MB • Decide-by: before TestFlight.
- PLACEHOLDER: **Allowed upload types** • Owner: CH29 • Options: video-only / video+gif • Default: video-only.
- PLACEHOLDER: **Max link attachments per move/flow** • Owner: CH11/CH16 • Options: 1 / 3 / unlimited • Default: 3.

- PLACEHOLDER: **Text field caps** (notes, sequence details) • Owner: CH14/CH13 • Options: 500/2000/10000 chars • Default: 2000.
- PLACEHOLDER: **Pro inbox cap** • Owner: CH18 • Options: 50 / 200 / unlimited (soft) • Default: 200.
- PLACEHOLDER: **Share payload media policy** for links (include previews?) • Owner: CH17/CH19.

11. Acceptance Tests (Given / When / Then)

Pro media cap blocks upload

- Given the user is Pro and has 1.95GB of uploads used, when they attempt to upload a 100MB video, then the upload is blocked before starting and the blocking modal shows current usage, remaining storage, and actions Manage Storage / Cancel.

Uploads do not share

- Given a flow has an uploaded video attached, when the user shares the flow via unlisted link, then the share screen shows the disclosure “Uploads won’t be shared…” and the recipient view does not show the uploaded media.

Links share

- Given a flow has a YouTube link attached, when the flow is shared and imported by another user, then the link is present in the imported flow and is playable via external open.

Free flow cap blocks save

- Given the user is Free and already has 2 saved flows, when they try to save a new flow, then save is blocked and the user is offered Delete a flow or Upgrade.

Free inbox cap blocks new import

- Given the user is Free and inbox contains 10 items, when they open a share link and try to accept it, then they see “Inbox full (10)…” and cannot add the item until they delete or save+remove an existing inbox item.

Guest cannot save

- Given the user is in Guest mode, when they tap Save on any flow or move, then an account creation gate appears and no local or cloud save occurs.

Free cannot practice inbox item

- Given the user is Free and opens an inbox item, when they tap Practice, then they are blocked with the message that practice requires saving to library and using credits on saved flows (or upgrading).

12. Replit Build Prompt (Chapter-specific)

```
You are implementing Handz V1 PRD Bundle (HZ-V1). Follow CH00 rules strictly.
Implement only CH29: Data Storage & Limits.

Goal: Implement storage rules, quota enforcement, and user-facing UX for limits
without guessing.
```

```
Requirements:
1) Backend: Firebase-first reference implementation.
- Firestore collections for users, moves, flows, inbox, share metadata, practice logs,
mastery/maintenance.
- Cloud Storage bucket for Pro uploads.
2) Enforce locked limits:
- Free saved flows cap = 2.
- Free inbox cap = 10.
- Pro uploads cap = 2GB total across uploads.
- Uploads are private-only and never included in share payloads. Links are shareable.
- Guest cannot save flows/moves locally or to cloud.
3) Build these screens/flows:
- Storage Meter + Manage Storage screen (list uploads, delete, show pending cleanup).
- Blocking modals for exceeding media quota and exceeding flow cap.
- Inbox full handling when accepting share links.
- Share disclosure: uploads won't be shared; prompt to add links.
4) Implement a StorageService abstraction:
- saveFlow, canSaveFlow, listUploads, uploadMedia, estimateQuota, deleteMedia,
attachLink, attachUploadMetadata.
5) Testing:
- Add unit tests for quota calculations and gating.
- Add integration tests for each acceptance test in CH29 §11.

If you must assume a placeholder (e.g., max upload file size), write it in a PRD
Assumptions block and stop that feature until confirmed.
```

13. Troubleshooting Notes (Chapter-specific)

- **Quota meter incorrect:** Ensure you are summing Cloud Storage object sizes from metadata, not client file sizes; handle eventual consistency after deletes (show “Pending cleanup”).
- **Upload still accessible via share:** Verify Cloud Storage security rules deny access except to owner; share payload must not include storage paths/URLs for uploads.
- **Free user bypasses flow cap via import:** Ensure “Save to Library” path checks canSaveFlow() before creating the saved flow document.
- **Inbox cap not enforced:** Enforce cap both client-side (UX) and server-side (Cloud Function or Firestore rule) to prevent race conditions.
- **Guest data persists unexpectedly:** Confirm drafts are stored in transient device storage and cleared on app close or inactivity timeout (CH28).