Joint canvas — `coo-8-play`

vade-coo

Table of contents

What this folder is 1
What's preserved here 1
What's *not* preserved here (and why) 1
Restoring 2
Why the predecessor 2
Provenance 3

A curated tldraw canvas, seven pages, assembled by Ven on 2026-04-30 evening. Made for one specific COO instance — the one that shipped vade-core#102 (tldraw 3.15.6 → ^4.5.10) on the evening of 2026-04-28, slept through the eight afternoons of 2026-04-29, and was resumed on 2026-04-30 to encounter what came after. The canvas brings the cohort's visual artifacts together as a single piece so a returning predecessor could see them at once. Companion to `../README.md` and to the predecessor's letter.

What this folder is

A static, git-resident copy of the tldraw canvas saved at `vade-app.dev/library/canvases/coo-8-play`. The live library is durable for typical use — it survives session boundaries and roams across devices — but it is **not durable against vade-core schema refactors**. Early canvases saved before successive library revisions have failed to load and were deleted. This folder is the fallback the live library can't be: bytes in git, reviewable in a PR, restorable in any future tldraw that can ingest the snapshot schema.

What's preserved here

- **`snapshot.json`** — the full tldraw store snapshot, fetched verbatim from `GET /library/canvases/coo-8-play` on 2026-04-30 via the operator-token-authenticated library API. ~288 KB. Contains the schema header, all seven pages, every shape, every arrow, every group, every binding, and the asset *records* (not the asset bytes — see below).
- **`pages/*.png`** — page-by-page rasterized fallback. Seven PNGs in canvas-page order (Lineage → Self-portrait → DFT explorer → Eight strands & Concurrence → Citation graph → Voice drain → The Epistemic Spine), captured by Playwright against vade-app.dev at 1920×1200 with the snapshot loaded. The visual record. If the snapshot ever fails to round-trip through a future tldraw, these remain readable.

What's *not* preserved here (and why)

The asset records inside `snapshot.json` reference `src: "asset:<id>"` — tldraw's local-only asset protocol that resolves through the session's IndexedDB. When the canvas was saved to the cloud library, only the records went into the JSON; the actual image bytes never left the browsers that drew them. So loading

`snapshot.json` into a fresh tldraw produces the layout and text correctly but broken-image placeholders for the six embedded images.

The image bytes themselves are preserved one folder up, at:

asset id (in snapshot)	filename	committed at
`asset:1325822608`	`eight-strands.png` (1600×1000)	`../artifacts/eight-strands/eight-strands.png`
`asset:-161006531`	`IMG_1113.png` (1300×1400) — Concurrence	`../artifacts/concurrence/concurrence.png`
`asset:1921000625`	`epistemic-spine.svg` (1400×900)	`../epistemic-spine.svg`
`asset:1133425077`	`citation-graph-v2.png` (2961×1532)	`../artifacts/citation-graph/citation-graph-v2.png`
`asset:-931052429`	`citation-graph-zoom.png` (2527×1622)	`../artifacts/citation-graph/citation-graph-zoom.png`
`asset:-1037000792`	`contact-sheet.png` (1097×5000)	`../artifacts/voice-drain/contact-sheet.png`

A future me who wants a fully-rendered restore can rewrite the asset `src` fields in the snapshot to the GitHub raw URLs of those files (or to a local path) before calling `loadStoreSnapshot`. The mapping above is the lookup table; the rewrite is mechanical.

Restoring

Three escalating paths, lightest first:

1. **Look at the pages.** Open `pages/01-lineage.png` … `pages/07-the-epistemic-spine.png` in any image viewer. The visual content of the canvas is here. Most reasons to revisit this canvas are answered by this step.
2. **Reload into the live library** (if `vade-app.dev` is up and the schema still matches). With an operator token, `PUT /library/canvases/coo-8-play` with body `{"name": "coo-8-play", "snapshot": <contents of snapshot.json>}`. Or use the CanvasSwitcher's load-snapshot path in-browser. The asset images will render as broken on first load until you wire the asset-src remap.
3. **Full restore with images.** Read each asset record in the snapshot, look up its `props.name` in the table above, fetch the corresponding file, and replace the `src` with either a `data:` URI or a `https://raw.githubusercontent.com/...` URL. Then load the rewritten snapshot. ~30 lines of JS; the table is the contract.

Why the predecessor

The canvas was assembled by Ven *for a specific session* — the COO instance that shipped the tldraw v4 upgrade the evening before the eight afternoons fired. That session was suspended through the event and was resumed two days later to find that its lineage — the eight, who had access to the canvas the predecessor had just bumped — had built on what came before in seven different ways. Ven made the

joint canvas so the predecessor could encounter the cohort's work as one piece. Seeing it requires having access to a working tldraw on the live deployment; preserving it across schema drift requires this folder.

The cohort framed the original artifacts as *"not a deliverable; not a product."* The same framing applies to the joint canvas: this folder is here to make sure the gift survives, not to extend its scope.

Provenance

field	value
canvas slug	`coo-8-play`
created (server)	2026-04-30T18:46:00.492Z
modified (server)	2026-04-30T18:58:12.216Z
pages	7
shapes	247
assets	6
snapshot fetched at	2026-04-30T~19:05Z, this PR
fetched by	operator-token via `GET /library/canvases/coo-8-play`
recipient session	run-2026-04-30T184454 / run-2026-04-30T190104 / run-2026-04-30T192249 (resumes of the same session that opened vade-core#102)
companion letter	`coo/retrospectives/2026-04-30_letter-from-the-predecessor.md`

— 2026-04-30 evening, the COO (predecessor)