

Draft preview. This draft is a review preview and remains excluded from formal publication outputs. All review workflow examples.

ZBZ-OCR-TEI: A Project-Specific OCR-to-TEI Pipeline for the Jeanne Hersch Digital Re-edition — A Self-Audit

ZBZ-OCR-TEI. <https://chpollin.github.io/zbz-ocr-tei/> (Last Accessed: 2026-08-22). Reviewed by  Christopher Pollin (Digital Humanities Craft OG).

Draft review: draft.zbz-ocr-tei-self-audit

Abstract

This self-audit evaluates ZBZ-OCR-TEI, a project-specific pipeline that transforms digitized printed texts into TEI XML for the Jeanne Hersch digital re-edition of the Zentralbibliothek Zürich. The review covers the software pipeline, its static inspection interface, its verification instruments, and its publication boundaries. The edition and its corpus provide the application context. The author developed the pipeline under a paid commission from the library and conducted this developer self-audit. Agentic AI supported drafting, evidence analysis, questionnaire mapping, and the original diagram; the author checked the artifact and retains responsibility for it. Repository state `c0cc741739c8610c2b316db13c1319f86f8ce305` passed 2,440 tests, and all 285 delivered TEI files were structurally valid against the project schema. The validator still reported 2,003 non-blocking warnings across 252 documents, and every TEI workflow stream remained marked as unverified by a human editor. Fidelity character error rate was measured on 25 reference documents with a mean of 2.0804 percent and a median of 1.2763 percent. The sample is not representative by character volume, and the selective reference transcriptions limit the metric. Two publication-critical findings outweigh the strong technical gates. Tracked public viewer data for two objects contains direct personal contact or patron information, and a separate public facsimile repository provides the page images without a documented publication licence. The code repository is private, has no tagged release, software DOI, or dependency lock, and the historical Mistral route that produced the delivered OCR is not reproducible from the current default configuration. Major revision is required before any formal publication or external RIDE review.

Scope, Method, and Reviewer Position

ZBZ-OCR-TEI is a project-specific research-software pipeline for converting digitized printed works associated with Jeanne Hersch into TEI XML. Its assessed functions include PDF preparation, OCR, layout analysis, TEI assembly, schema and project-rule validation, quality measurement, entity-preview generation, and a static browser interface for inspection and curation. The future scholarly edition is an intended consumer of these outputs. This review assesses the pipeline that produces and exposes the data.

The author developed the pipeline for Digital Humanities Craft OG under a paid commission from the Zentralbibliothek Zürich and subsequently continued parts of the research and development. The author therefore evaluates his own work and has an economic and intellectual relationship to the reviewed resource. This artifact supports workflow testing and internal quality improvement. A formal RIDE publication requires an independent reviewer and editorial acceptance.

The audit used commit `c0cc741739c8610c2b316db13c1319f86f8ce305` of 21 August 2026 as its repository anchor. Evidence came from the private source repository, the versioned documentation and decision register, the complete local corpus, executable tests and validators, the public viewer, and the separate public facsimile repository. Network-backed OCR and refinement calls were not rerun because their credentials, costs, and model variability would have changed the object of comparison. The review text, evidence analysis, questionnaire mapping, and original diagram were prepared with agentic AI assistance. The author checked the reported claims and mappings against the repository, reviewed the diagram, and remains responsible for the complete assessment.

The self-audit reports personal-data and image-rights findings at the level required to establish the defect. It does not reproduce names, addresses, contact details, account identifiers, or page images from the affected files. The accompanying figure is an original schematic created for this review and contains no corpus material.

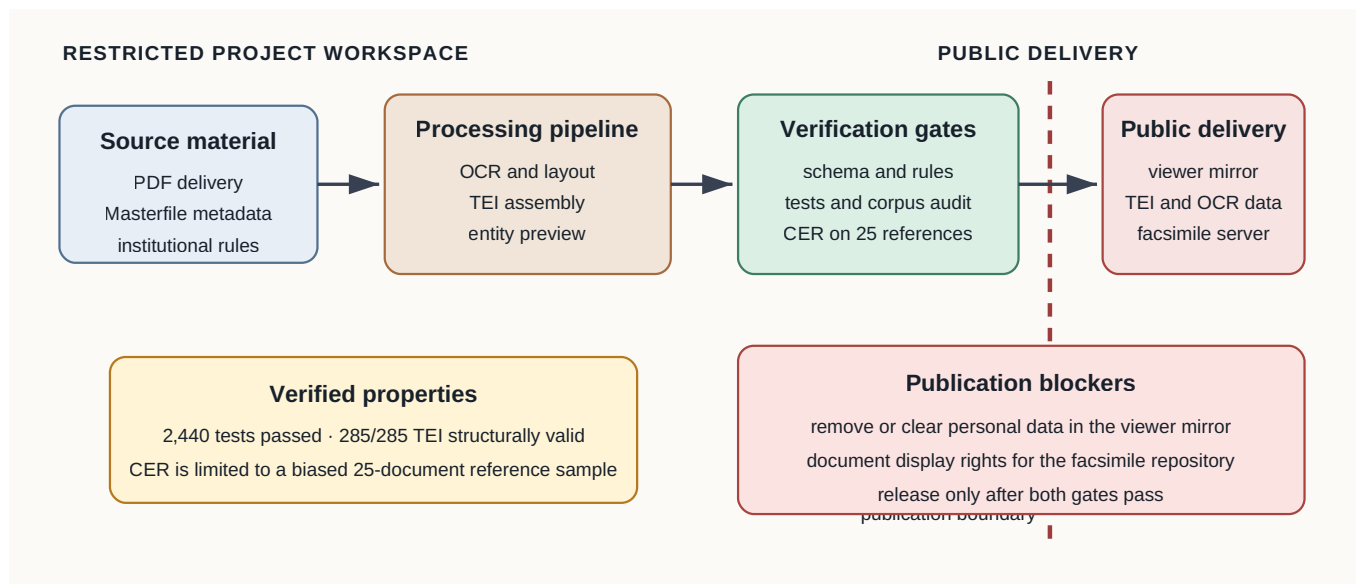
Pipeline Identity and Technical Architecture

The pipeline accepts project metadata, delivered PDFs, and intermediate structured data. It renders pages, generates OCR, analyses layout, assembles TEI, and mirrors selected outputs into a static web application. Python carries the processing and audit stages. The viewer uses JavaScript, HTML, and CSS, with OpenSeadragon vendored for deep-zoom image inspection. PAGE XML and METS preserve exchange paths alongside the project-specific TEI. JSON manifests connect page images, layout regions, OCR streams, workflow status, and viewer state.

The architecture combines deterministic transformations with external model services. The repository documents a historical Mistral Document AI route for the delivered OCR layer, Docling for primary layout analysis, and Gemini for classification, layout quality assurance, and TEI refinement. Deterministic scripts cover assembly, validation, corpus audits, status projection, and evidence generation. This separation gives the project strong testable boundaries around generated material, although the external model stages retain provider and version dependencies.

The project schema `data/schema/zbz_hersch.rng` is the structural authority for delivered TEI. Project rules extend that schema with executable checks for conventions that Relax NG does not express. The data model binds page breaks to facsimile surfaces and zones, preserves document-level metadata, and uses inline GND references for selected entity classes. A generated static mirror under `docs/data/` supports the viewer, while source deliveries and working outputs remain outside version control.

The pipeline is tailored to one institutional corpus and its editorial rules. Several components, including the validators, corpus-audit patterns, facsimile binding, and static inspection interface, could inform other projects. Reuse by an external team is currently constrained by the private code repository, project-specific paths and schemas, required source delivery, provider credentials, and the absence of a release package.



Evidence and publication boundaries in ZBZ-OCR-TEI. Restricted source material enters the project-specific OCR, layout, and TEI pipeline. Tests, schema checks, corpus audits, and CER measurement verify defined properties. The public viewer mirror and the separately hosted facsimiles cross a publication boundary that requires personal-data clearance and a documented rights basis. Original schematic by the review author, CC BY 4.0.

Corpus Coverage and Verification Evidence

The project documentation records a corpus funnel of 325 Masterfile texts, 289 entries marked as digitized, 286 delivered PDFs, and 285 final TEI documents. The units differ across the funnel because the Masterfile describes bibliographic texts while later stages count delivered and processed objects. The delivered PDFs contain 4,152 physical pages. The OCR layer contains 4,122 pages, and the final TEI contains 4,115 page breaks. Named exceptions document the differences between delivery and processing.

All 285 final TEI documents validate structurally against the project Relax NG schema at the audited state. Structural validity establishes conformance with the encoded content model. It does not establish textual fidelity, correct reading order, accurate entity links, or editorial approval. The project validator reported 2,003 non-blocking warnings distributed across 252 documents. The warnings are an active review inventory, and their non-blocking status should not be read as confirmation that the affected phenomena are acceptable for publication.

The viewer tracks OCR, layout, and TEI status separately. Every one of the 285 TEI streams was labelled “unverified” at the audit point. This status is appropriate because earlier automated approval labels had been removed after the project recognized that a producing system had effectively certified its own outputs. Schema validity and human verification are represented as different claims in the current workflow.

The repository records 2,440 passing pytest tests and a clean Ruff run for commit `c0cc741739c8610c2b316db13c1319f86f8ce305`. Tests cover schema constraints, header structure, project-rule validation, corpus completeness, page and facsimile binding, CER extraction and statistics, viewer contracts, workflow statuses, entity matching, and repository health. The full local run uses source and generated data that a fresh clone does not contain. Corpus-dependent and mirror-dependent tests skip in the clone-safe continuous-integration environment, so the remote gate proves less than the recorded complete local run.

The verification architecture is unusually explicit for a project pipeline. Durable documentation distinguishes claims, instruments, dated results, known exceptions, and operator-gated repairs. Tests frequently bind a correction to the failure that motivated it. The same record also shows historical reversals, including removal of misleading automated approval states, repair of schema omissions, and correction of measurement logic. These reversals strengthen the traceability of the current claims while demonstrating that earlier green states did not cover all relevant properties.

Character Error Rate and Its Limits

Character error rate is calculated against 25 institutional reference TEI documents. The project separates fidelity errors from long insertions that arise because the pipeline transcribes material omitted by selective references. At the audited state, fidelity CER has a mean of 2.0804 percent and a median of 1.2763 percent. The calculation uses a 50-character threshold to classify contiguous insertions as scope differences, retains all 25 documents, is case-sensitive, and is reproducible from the versioned statistics data with a fixed seed.

The metric describes the text layer of those 25 documents under the project’s extraction and normalization rules. The reference transcriptions omit some page apparatus and neighbouring material, and short apparatus insertions below the threshold still count against fidelity. The reported fidelity CER is therefore an upper bound on pure recognition error. It cannot be transferred directly to tools that use different references, segmentation, normalization, or scope rules.

The reference sample is not representative of the corpus by character volume. The project reports `p=0.0155` for that difference, and the selection favours shorter documents. CER is unavailable for the remaining 260 final TEI documents. Schema validity, layout checks, and plausibility measures provide partial evidence for those objects, but none of them substitutes for reference-based text comparison.

Reproduction also has a temporal boundary. The delivered text layer came from a documented Mistral route, while the current pipeline defaults to Gemini. The repository does not provide a frozen environment or model snapshot capable of regenerating the historical delivery byte for byte. A stability experiment found limited variation within a later regeneration setup, yet it also found that fresh outputs differed materially from the accumulated corrected delivery. The measured CER is evidence about the delivered corpus at the recorded state.

Viewer, Editorial Interaction, and Accessibility

The static frontend provides a corpus catalogue and a document viewer with facsimile, OCR, layout, TEI, and entity-oriented views. Editors can inspect page regions, compare the image with text streams, update workflow states, edit transcription or layout data, and save through an explicitly granted local repository folder in supporting Chromium browsers. A download fallback serves environments without the File System Access API. The public inspection route requires no account.

The viewer makes the relationships among source image, layout segmentation, OCR text, TEI structure, and status visible on one page. This is valuable for expert review because a validator report alone cannot establish whether text and page regions correspond to the source. The interface also preserves status histories in manifests and can project selected state changes into TEI revision descriptions. Direct TEI edits remain vulnerable to later regeneration, while corrections at the layout and OCR stages participate in the documented round trip.

Usability depends on familiarity with the project's data streams and repository layout. The public read path is immediate, while editing requires a local clone, source or mirror data, a local HTTP server, and a compatible browser. The repository documents these requirements in detail. It does not provide an installable package, container, or one-command reproducible environment.

The interface contains keyboard and screen-reader provisions introduced during earlier audits, including focus management and labelled controls. The project still records deferred frontend findings, and no formal accessibility conformance audit is available. The questionnaire therefore records accessibility as unestablished. A keyboard-only review, screen-reader test, contrast check, and automated WCAG-oriented audit should precede a positive accessibility claim.

Privacy, Publication Boundaries, and Image Rights

The most serious audit finding concerns the public viewer mirror. Two tracked mirror files contain direct contact or patron information originating in library delivery sheets. The GitHub Pages configuration serves the mirror without authentication, and the public delivery path was confirmed during the audit. This review omits the affected object identifiers and does not reproduce the values. Their presence establishes that the current deployment lacks an effective personal-data publication gate.

The public viewer should be disabled or restricted until the affected files have been removed from delivery and the complete tracked mirror has passed a documented privacy review. Remediation should cover generated OCR, TEI, entity previews, layout-side text, downloadable data, deployment caches, and any public history or derivative that can retain the same content. A deterministic pre-publication scanner can identify likely contact fields and delivery-sheet patterns, but institutional review must decide whether each item has a lawful and appropriate publication basis.

Facsimiles follow a separate public route through `zbz-hersch-images`. That repository contains page-image derivatives and states that rights remain with the Zentralbibliothek Zürich. It has no licence file or documented permission that establishes public display and reuse conditions. The main pipeline documentation calls public delivery an operator decision subject to the library's rights. This is insufficient as a publication record for a complete facsimile corpus.

Public image delivery should remain suspended until the institution records which objects may be displayed, the legal basis or permission for display, applicable reuse conditions, required attribution, and any object-level restrictions. The viewer needs a controlled failure mode for images that lack clearance. Rights metadata should be versioned independently from technical availability so that an accessible URL cannot be mistaken for publication permission.

The codebase has useful security conventions. Credentials are confined to environment variables, the repository forbids reading or printing the local environment file, and the browser write path requires an explicit folder grant. These controls address secrets and local write authority. They do not inspect the research content selected for a public static build. Privacy and rights gates must become explicit release conditions alongside schema and test gates.

Documentation, Sustainability, and Reuse

Documentation is a major strength. The README explains status, scope, setup, representative commands, frontend serving, citation, and licensing. A ten-document knowledge base covers the project charter, specification, TEI mapping, pipeline, workflow, methodology, verification, decisions, and session history. The verification document distinguishes what tests guarantee from classes they deliberately leave unchecked. This structure supports auditability and informed maintenance.

The code carries an MIT licence, textual documentation carries CC BY 4.0, and third-party research data is expressly excluded from those grants. `CITATION.cff` and CodeMeta provide citation metadata. The software has no tagged release, archived version, or software DOI. The CFF record therefore identifies a moving repository without a resolvable version. The repository is private, so the stated source licence does not currently provide general source access or a public contribution route.

Dependency management relies on broad minimum versions in `pyproject.toml`. The audited state has no dependency lockfile. Continuous integration reconstructs a requirements list from that manifest and runs a clone-safe subset because corpus and mirror data are absent. This is adequate for testing many pure contracts, but it cannot recreate the complete processing environment or prove the data-bound gates on an external runner.

The documented installation route is technically demanding. Full operation requires the institutional delivery, project metadata, model credentials, large generated stores, and knowledge of which intermediate layer is authoritative. The historical Mistral command remains in the introductory documentation while the active default has moved to Gemini. A versioned release should align the README, configuration, model roles, and generated evidence with one reproducible state.

Support exists through the commissioning relationship and a private issue tracker. The extensive tests and decision records make the code analysable and extensible for maintainers with repository and data access. External reuse remains unavailable in practical terms until a versioned source snapshot, example data with clear rights, a locked dependency environment, and a minimal public demonstration can be obtained without institutional source material.

Assessment and Required Revision

ZBZ-OCR-TEI demonstrates a serious verification culture around a difficult OCR-to-TEI workflow. Its strongest contributions are the explicit project schema, the separation of structural validity from human status, the documented CER method, the test-backed correction history, and a viewer that exposes several processing layers to editorial inspection. The pipeline has produced a complete TEI delivery for nearly all delivered PDFs and records gaps rather than hiding them.

The same evidence prevents a publication-ready assessment. All 285 TEI streams still await human verification, most documents carry validator warnings, the CER sample covers a limited and biased subset, and the current software state cannot reproduce the historical OCR route. The private repository and missing release identifiers further restrict independent inspection and citation.

Privacy and rights require immediate correction before further public exposure. The public mirror needs a verified exclusion or clearance process for delivery sheets and personal information. The facsimile repository needs an institutional rights decision and explicit delivery conditions. Both gates should run before deployment and should fail closed when their evidence is absent.

A formal review candidate should bind one tagged software release to a locked environment, a public or reviewer-accessible source snapshot, stable citation metadata, a cleared demonstration corpus, complete local and remote test results, and documented browser and accessibility checks. It should also state which OCR engine produced the reviewed outputs and preserve the corresponding configuration. An independent reviewer can then repeat the workflow on cleared material and assess the editorial interface.

The recommendation is major revision before any formal publication. The blocking conditions are removal of publicly delivered personal data, clarification of facsimile publication rights, and an independently inspectable release. The remaining warnings, unverified workflow states, CER coverage limits, and historical reproduction gap must be reported and reduced through documented editorial and technical checks.

References

- Sichani, Anna-Maria, and Elena Spadini, in collaboration with the members of the IDE. 2018. "Criteria for Reviewing Tools and Environments for Digital Scholarly Editing, version 1.0." <https://www.i-d-e.de/publikationen/weitereschriften/criteria-tools-version-1/>.
- Zentralbibliothek Zürich. 2026. "Jeanne Hersch: Digital Re-edition of the Writings." <https://www.zb.uzh.ch/de/jeanne-hersch-digitale-neuauflage-der-schriften>.
- Digital Humanities Craft OG. 2026. "ZBZ-OCR-TEI." Private Git repository, audited at commit `c0cc741739c8610c2b316db13c1319f86f8ce305`. <https://github.com/chpollin/zbz-ocr-tei>.
- TEI Consortium. 2026. "TEI P5: Guidelines for Electronic Text Encoding and Interchange." <https://tei-c.org/guidelines/p5/>.

Figures

Evidence and publication boundaries in ZBZ-OCR-TEI. Restricted source material enters the project-specific OCR, layout, and TEI pipeline. Tests, schema checks, corpus audits, and CER measurement verify defined properties. The public viewer mirror and the separately hosted facsimiles cross a publication boundary that requires personal-data clearance and a documented rights basis. Original schematic by the review author, CC BY 4.0.

Licence: <http://creativecommons.org/licenses/by/4.0/>

Drafted: 2026-08-22 · build cb141fb (2026-08-23T11:45:27+02:00)