CI Architecture, Release Contracts, and Review Lanes

Tiered CI Pipelines, Quality Dashboard, Mapping Workbench, Standalone Package

and Dual-Release Contracts, Review Processes (HR-001–HR-006, MEM/SPARQL), and Runbooks

Scope of this volume

Complementing Volume IV (`M0`–`M7` maturity, `CQR-*`, `GVR`/`APV`/`CPV`, semantic atomicity, and Champion trials), Volume IVb specifies SkillPilot's operational verification and release infrastructure: the two-tier GitHub Actions and `./run_ci.sh` suite, the Quality Dashboard and Mapping Workbench, fail-closed `full-standalone-v1` package readiness, release-model conformance, and dual JSON/OWL release contracts, plus the manual and semi-automated review lanes (`HR-001`–`HR-006`, atomic/requires review, MEM SPARQL consistency, source rationales, and Abi 2026 deep-link QA).

As of 11 November 2026 · License: Apache-2.0 (technical CI, package, and review documentation; see LICENSING.md).

Machine conformance never substitutes for human review gates (HR-001 through HR-006) or practical classroom trials.

Contents

1 Continuous Integration (CI) 4
1.1 Triggers and Two-Tier Execution . . . 4
1.2 Quick-Tier and Complete-Check Jobs . . . 4
1.3 Curriculum, View, and Archive Boundaries and Optional Package CI . . . 5
1.4 Ubuntu Package Sources and Failover . . . 6
1.5 Running CI Locally and Status Snapshots . . . 6

2 Curriculum Quality Dashboard 8
2.1 Generators, Artifacts, and Local Commands . . . 8
2.2 Maturity Levels `M0`–`M7`, Source Evidence, and `curriculum-quality-v5` Rules . . . 8
2.3 Mapping Pipeline, Bundesland Evidence, and Route Profiles . . . 9

3 Curriculum Mapping Workbench 11
3.1 Seven-Layer Repository Data Flow . . . 11
3.2 Fail-Closed Pipeline Gates (`MAPPING-1` through `MAPPING-3`) . . . 11
3.3 Two-Stage Interaction Model and Audit Interpretation . . . 12

4 Curriculum Package Readiness (`full-standalone-v1`) 13
4.1 Status Vocabulary and Exit Codes . . . 13
4.2 Conformance Suites, Build, and Hermetic Package Consumer . . . 13
4.3 Offline Trust Model, Review Separation, and Implementation Boundary (`DPK-004a`–`DPK-008d`) . . . 14

5 Curriculum Release Model Conformance 16
5.1 Overview and Trusted Inputs . . . 16
5.2 Generated Model Layout, Digests, and Publication Evidence . . . 16
5.3 Verification Commands and Hosted PDF Mode . . . 17

6 Dual Curriculum Release Contracts 19
6.1 Binding Contract Invariants . . . 19
6.2 Executable Gate and Boundary . . . 19

7 Requires Relation Checks 20

8 Atomic Review Process 21
8.1 Scope and Working Definition . . . 21
8.2 Seven-Step Review Workflow and Decision File . . . 21

9 Requires Review Process 23
9.1 Transition Semantics vs. Canonical Target State . . . 23
9.2 Six-Step Review and Repair Workflow . . . 23

10 Human Review Gates for Curriculum Releases (`HR-001`–`HR-006`) 25
10.1 Overview of the Six Independent Gates . . . 25
10.2 Domain, Rights, and Source-Text Gates (`HR-001` through `HR-004`) . . . 25
10.3 Ontology Sign-Off and Stable Promotion (`HR-005` & `HR-006`) . . . 26

11 MEM SPARQL Consistency Audit 27
11.1 Sources of Truth, Generated Artifacts, and Execution . . . 27
11.2 Fingerprint-Bound Review Ledger and Expansion Criteria . . . 27

12 MEM SPARQL Consistency Runbook **29**
12.1 Running the Audit and Inspecting the Queue 29
12.2 Recording Decisions, Interpreting Diagnostics, and Verifying Locally 29

13 Goal Source Rationales Runbook **31**
13.1 PoC, Custom Goal Sets, and MEM/FWU Showcase Commands 31
13.2 Runtime Public Index (Repository vs. Package Mode) & Deployment Smoke . . 31
13.3 Review Checklist and Follow-Up Issues 32

14 Abi 2026 Mathematics DeepLink MVP (BlueSky) **33**
14.1 URL Contract and Cockpit Pre-Provisioning 33
14.2 Campaign Events and QA Verification (3 Test Links) 33

1 Continuous Integration (CI)

This chapter documents the two-tier CI workflow in `.github/workflows/ci.yml`, the optional package and ontology workflow in `.github/workflows/owl-ci.yml`, and the local `./run_ci.sh` suite.

1.1 Triggers and Two-Tier Execution

The `.github/workflows/ci.yml` workflow runs on:

- pushes to `main`,
- pull requests targeting `main` (when opened, updated, reopened, or marked ready for review),
- manual dispatch (`workflow_dispatch`).

Documentation-only and temporary changes (`README.md`, `AGENTS.md`, `docs/**`, `sandbox/**`, `tmp/**`) do not trigger `ci.yml`; documentation is gated by its dedicated `.github/workflows/docs_checks.yml` workflow.

Execution is organized into two tiers whose invariants are enforced by `scripts/ci_tiers.test.mjs` inside `quick-checks`:

- **Quick tier:** Runs on every trigger, including draft pull requests. Its jobs depend on no upstream job and therefore report early feedback first.
- **Complete check:** Runs in parallel across all full verification jobs on pushes to `main`, manual starts, and non-draft pull requests.

Binding tier rules:

- The complete check contains every required gate on its own; the quick tier only repeats fast checks early and never substitutes for a full check.
- A draft pull request runs only the quick tier (complete-check jobs appear as `skipped`). Marking the PR ready for review starts the complete check for that head; because GitHub never merges drafts, every mergeable PR carries a complete check for its current head.
- Neither tier selects or skips individual jobs by changed file paths.
- Any job split off from a parent job preserves that parent's runner image, toolchain, and restored goal-book artifact.

1.2 Quick-Tier and Complete-Check Jobs

Quick-tier job	Verified scope
`action-regression-ci`	Spring-hosted Custom GPT Action regression client and deployment entrypoint guards
`openai-mcp-app-ci`	OpenAI MCP App contracts, plugin release, and submission checks
`claude-connector-v1-ci`	Claude MCP Apps, plugin package, and direct-install publication
`openai-v1-public-edge-ci`	OpenAI V1 public edge and mTLS verification scripts
`quick-checks`	CI structure tests plus early runs of ESLint, TypeScript type-checking, `validate:graph`, and `validate_schemas.py`
`backend-quick-checks`	CI runner tests and compilation of backend main and test sources (without running backend tests)

In the complete check, `goal-book-publication` builds and verifies all registered learning-goal books once; every downstream job marked `yes` restores that artifact from the same run:

Complete-check job	Goal books	Verified scope
`goal-book-publication`	–	Build and verification of all registered learning-goal book publications
`frontend-ci`	yes	Fixture-based application logic, package/runtime adapters, learner-goal selection, GPT instruction limits, deterministic ZIP primitive, ESLint, `npm run build:application` (application-only against committed assets, unlike integrated `npm run build`), and built-app checks
`goal-book-contracts`	yes	`test:goal-book-pipeline`
`graph-validation`	yes	Skill graph, view, source, review, and status gates
`curriculum-assessment-proofs`	yes	Transitive-minimality proofs plus assessment, exam, and route tests
`curriculum-integrated-build`	yes	Runtime assets, visualization checks, integrated application build, and package-only frontend
`curriculum-package-contracts`	yes	Package contracts and self-tests, schemas, goal IDs, archive, and legacy-reference validators
`backend-ci`	yes	Every backend test class except prerequisite-reduction proofs across three disjoint parallel test JVMs (`scripts/run_backend_ci.py`; previously failed tests retry first)
`backend-reduction-proofs`	no	Prerequisite-reduction proofs across two parallel partitions (all non-Physics subjects plus the Physics method isolated); plain `./gradlew check` still runs the entire suite serially

1.3 Curriculum, View, and Archive Boundaries and Optional Package CI

Together, the four curriculum jobs validate code, contracts, models, reviews, schemas, and committed status files, without materializing a real 1.7-GB curriculum ZIP in per-commit CI:

- **Graph and projection rules:** `validate:graph` runs in strict mode by default (`GVR-*` failing; legacy warning mode locally via `VALIDATE_GRAPH_STRICT_RULES=0 npm run validate:graph`). In CI, `validate:view-filters` covers the 17 canonical DE Gymnasium landscapes (`Mathematik`, `Physik`, `Chemie`, `Biologie`, `Informatik`, `Deutsch`, `Englisch`, `Französisch`, `Griechisch`, `Chinesisch`, `Geschichte`, `Politik und Wirtschaft`, `Musik`, `Latein`, `Spanisch`, `Wirtschaft`, `Overview`; locally widenable via `APPLICABILITY_VALIDATION_SCOPE=all`). Reviewed residual `APV-201` warnings are tracked in `docs/qa-ci/applicability-accepted-warnings.json`; `APV-202` is diagnostic only.
- **Duration, source, documentation, and memory-card gates:** CI enforces the Hessen Mathematics Sek I G8/G9 duration projection (`check:he-math-duration-projection`), visible source coverage for Mathematics and Physics (`quality:source-coverage-audit:check`), explicit learner-facing composition views (`validate:composition-views`), generated-doc notices, status registries, links, and indexes (`check:generated-doc-notices`, `check:generated-status-registry`, `check:docs-links`, `check:docs-indexes`), and all configured memory-card ledgers (`quality:memory-card-review:check:all`, where missing `CQR-302` configuration blocks `M6` rather than core `M5`).
- **Archive and legacy-reference fences:** Validators enforce the Hessen upper-secondary retained-asset boundary under `curricula/DE/Gymnasium/input/HE/abi` (`curricula/DE/Gymnasium/input/HE/retained-asset-registry.json` and `curricula/DE/Gymnasium/`

`provenance/hessen-upper-secondary-retirement-registry.json`), the lower-secondary boundary under `curricula/DE/Gymnasium/input/HE/lower-secondary` (`curricula/DE/Gymnasium/provenance/hessen-lower-secondary-retirement-registry.json`), and the Hessen 2026 chemistry Abitur pipeline artifacts (`slot_matrix.json`, `coverage_requirements.json`, `task_bank.json`).

- **Optional package and FWU-OWL CI (`.github/workflows/owl-ci.yml`):** Runs weekly or manually on fresh runners covering the real 1.7-GB standalone JSON package and hermetic consumer, all subject-export ZIPs, and the Core-first FWU-OWL/roundtrip lane. Production runtime consumes only JSON packages and requires no OWL; the required JSON release-model gate verifies only the pinned FWU Core commit and file hash recorded as provenance without creating RDF or invoking ROBOT/HermiT. Real 2.36-GB FWU-OWL wrappers remain explicit release operations.

1.4 Ubuntu Package Sources and Failover

Every job that installs OS packages first runs `.github/actions/ubuntu-apt`, keeping APT strictly on Ubuntu sources while excluding the runner image's Chrome and Microsoft repositories:

- **Mirror-list failover:** Each Ubuntu stanza reads `/etc/apt/skillpilot-ubuntu-mirrors/` (`archive.txt` prefers `https://archive.ubuntu.com/ubuntu/` and `security.txt` prefers `https://security.ubuntu.com/ubuntu/`, with the runner image's Azure mirror `http://azure.archive.ubuntu.com/ubuntu/` second in both). APT's mirror transport shifts any file that fails or stalls beyond 30 s to the alternate source, with same-source retries disabled (`Acquire::Retries "0"`).
- **Pre-update `InRelease` probe & signature pinning:** Before every `apt-get update` (including Playwright's nested call), `/usr/local/sbin/skillpilot-order-ubuntu-mirrors` probes each source's `InRelease` (at most 20 s), assigns priority 1 to the first answering source, and logs `SkillPilot Ubuntu mirror order: …`. Suites, components, and `Signed-By` remain unchanged so every mirror is verified against the Ubuntu archive keyring and signed index hashes; ci_ubuntu_apt.test.mjs pins this policy offline against a signed local fixture repository.

1.5 Running CI Locally and Status Snapshots

From the repository root, `./run_ci.sh` dispatches the local verification suites (unknown or multiple suite arguments exit with code 2):

```
./run_ci.sh
./run_ci.sh all
./run_ci.sh application
./run_ci.sh curriculum
./run_ci.sh package
./run_ci.sh owl
./run_ci.sh full
```

- `application`: Custom GPT Action regression, fixture-based frontend logic, GPT instruction limits, lint, `build:application`, and isolated Gradle backend checks.
- `curriculum`: Graph/data/review/asset gates, schemas, and deterministic JSON release-model conformance (without materializing the real 1.7-GB ZIP).
- `package`: Optional real standalone JSON ZIP, independent package validation, secure provisioning/activation, Readiness evaluation, and hash-bound hermetic consumer.

- `owl`: Optional FWU-OWL builder/contracts, reverse-compiler tests, semantic MEM/FWU roundtrip, and OWL 2 DL/HermiT validation.
- `all` (default with no argument): Every required application and curriculum check once (always finish cross-cutting runtime changes with `./run_ci.sh` or `./run_ci.sh all`).
- `full`: `all` plus the optional `package` and `owl` lanes.

Optional local status and MEM snapshots:

```
cd app
npm run quality:curriculum-status
npm run quality:mem-sparql-consistency
```

The first command updates curriculum-quality-status.json and `.md` for `/quality-dashboard`; the second writes the non-blocking MEM/FWU audit and review-queue artifacts under `docs/qa-ci/status/` (see 12).

2 Curriculum Quality Dashboard

This chapter specifies the persisted quality-status layer backing the read-only local Workbench view at `/quality-dashboard` (for detailed maturity and `CQR-*` semantics, see Quality Handbook Volume IV).

2.1 Generators, Artifacts, and Local Commands

Rather than reimplementing validation logic in React, the dashboard reads generated status snapshots through `/__quality-dashboard/status` and `/__quality-dashboard/file?path=docs/qa-ci/status/<file>.md`:

- **Primary status:** `app/scripts/generateCurriculumQualityStatus.ts` writes curriculum-quality-status.json and curriculum-quality-status.md.
- **Memory-card review (`CQR-302`):** `app/scripts/memoryCardReview.ts`, `runMemoryCardReviewConfigs.ts`, and `reportMemoryCardReviewRollout.ts` validate `curricula/DE/Gymnasium/quality/memory-card-review/*.review.jsonl` plus `*.cards.review.jsonl` and emit `docs/qa-ci/status/memory-card-review-*.md` (currently completed and enforced in CI for Mathematics, Physics, Chemistry, Biology, German, History, Informatics, Latin, Politics and Economics, and Economics).
- **Remediation, source-coverage, and MEM reports:** `reportM0RemediationPlan.ts` (`m0-remediation-plan.md`), `reportEnglishRemediationPilot.ts` (`english-remediation-pilot.md`), `generateCurriculumSourceCoverageAudit.ts` (`curriculum-source-coverage-audit.{json,md}`), and the deliberately non-blocking `generateMemSparqlConsistencyAudit.ts` (`mem-sparql-consistency-audit.{json,md}`).

Regenerating local snapshots and bootstrapping a visible open memory-card queue (`needs_developer_review` records without semantic approval; `pilot-dossier` prepares non-enforced subject triage):

```
cd app
npm run quality:curriculum-status
npm run quality:memory-card-review:report:all
npm run quality:memory-card-review:check:all
npm run quality:memory-card-review:rollout
npm run quality:memory-card-review:pilot-dossier
npm run quality:m0-remediation
npm run quality:english-remediation-pilot
npm run quality:source-coverage-audit
npm run quality:mem-sparql-consistency
npm run quality:memory-card-review -- --mode=bootstrap \
  --config=curricula/DE/Gymnasium/quality/memory-card-review/<id>.config.json
```

2.2 Maturity Levels `M0`–`M7`, Source Evidence, and `curriculum-quality-v5` Rules

The dashboard assigns one conservative maturity level `M0`–`M7` per curriculum alongside the `M5-M7 review gates` strip (`CQR-301`, `CQR-401`, `CQR-501` for core `M5`, `CQR-302` for `M6`, and `CQR-303` for `M7`). Human Champion trials remain an orthogonal statement (Quality & Human Trials). Official sources are tracked in two layers: committed Git metadata stores structured `sourceDocument`/`sourceDocuments` references with official HTTP(S) URLs and readable extracted inventories, whereas local PDFs (`curricula/**/*.pdf`) are gitignored working copies, so a green source

badge remains valid with `0/n` local PDFs whenever all `n/n` official URLs and inventories are present.

Rule	Target	Meaning in `curriculum-quality-v5`
`CQR-000`	`M1`	Source inventories readable, linked to official HTTP(S) URLs, and extracted goals registered in the membership/closure ledger.
`CQR-001`	`M0`	Basic graph integrity: IDs, local/global references, self-reference guards, and direct DAG check.
`CQR-002`	`M0`	Explicit `type` metadata matches structural atomic/cluster classification.
`CQR-003`	`M2`	Bundesland coverage: full source backing for source-relevant view atoms, zero unsupported assignments, zero missing registered source goals.
`CQR-004`	`M2`	Course-level consistency: source goals (`GK_LK`, `LK`, `unspecified` defaulting to `GK_LK`) map only to compatible `GK`/`LK` goals.
`CQR-101`	`M3`	Effective route coverage via $R_{\text{effective}}$ from motivation anchor through atomic goals to terminal autonomy.
`CQR-102`	`M3`	Direct atomic route coverage via authored atomic `requires` edges.
`CQR-103`	`M3`	Zero remaining route-scope cluster-level `requires` for ordinary sequencing.
`CQR-201`	`M4`	Terminal autonomy goals carry `examData` (or an explicitly reviewed exception convention).
`CQR-301`	`M5`	Semantic-atomicity review ledgers for content leaf goals are complete, current, and resolved.
`CQR-302`	`M6`	Memory-card ledgers explicitly decide memorization need, trace every kept primary card, and ensure visibility in learner views.
`CQR-303`	`M7`	Central report from `reportDeepUnderstandingRollout.ts` proves the exact D/P/A/M/V gate intersection across all current `curricularAtomic` goals (technical image deferrals never count; `humanApproved`/`humanIssue Identified` neither satisfy nor veto Gate V/`M7`).
`CQR-401`	`M5`	At least one learner-facing composition view exists.
`CQR-501`	`M5`	Applicability warnings are cleanly split into active, accepted-current, and obsolete-accepted counts (`active = 0`).

2.3 Mapping Pipeline, Bundesland Evidence, and Route Profiles

Separately from `M0`–`M7`, the dashboard surfaces per-source `source-extraction` pipeline progress (`complete`, `incomplete`, `blocked`; missing extractions render as `0/3` with `MAPPING-1` open):

1. **`MAPPING-1`:** Official syllabus passages completely extracted.
2. **`MAPPING-2`:** Source goals created from those passages.
3. **`MAPPING-3`:** Source goals mapped to canonical SkillPilot goals (completed for Hessen Mathematics Sek II across `E.1`–`E.7`, `Q1.1`–`Q1.5`, `Q2.1`–`Q2.5`, `Q3.1`–`Q3.5`, `Q4.1`–`Q4.3` with `316/316` source goals, and for Hessen Mathematics Sek I across all `426/426` KC 7.3, KC 7.1/7.2/6, and G9 topic-line source goals).

For `CQR-003` **(Bundesland coverage)**, only source-coverage-relevant atomic goals rendered by learner-facing composition views count (excluding memory/SRS, practice, assessment, motivation, orientation, and `examData` nodes). Direct syllabus coverage accepts **only** `provenance`, reviewed `mapping` edges (including `partial`, which stay visible as precision warnings), and explicitly accepted surrogate entries for genuine prerequisite gaps — **never** manual `override` overlays, `child-union`, or automatic `requires-closure`.

Currently required route profiles and local verification sequence (e.g., advancing Mathematics Sek I from `M3` toward `M4`):

- **canonical-math-sek1 / canonical-math-sek2:** Mathematik (Gymnasium, DE), motivation anchors Warum Mathematik? - Entdecken, Muster & Alltag and Warum Mathematik? - Denken, Muster & Zukunft, terminal exam tasks under Prüfungen Jahrgangsstufe 5–10 and Übungen E-Phase, Q1–Q4, plus Übungen Prozesskompetenzen.
- **canonical-physics-sek1 / canonical-physics-sek2:** Physik (Gymnasium, DE), motivation anchor Warum Physik?, terminal tasks under Übungen Sekundarstufe I Physik and Übungen E-Phase plus Q1–Q4.

```
cd app
npm run quality:curriculum-status
npm run validate:graph
npm run validate:view-filters
npm run validate:composition-views
npm run quality:semantic-atomicity:check -- \
  --config=curricula/DE/Gymnasium/quality/semantic-atomicity/canonical-math-full.config.
json
```

3 Curriculum Mapping Workbench

Whereas the Quality Dashboard (2) reports *whether* generated QA rules are green, yellow, or red, the read-only local **Curriculum Mapping Workbench** lets curriculum owners inspect *why* — tracing every step from an official syllabus PDF into the learner-facing SkillPilot tree.

```
Route:     /curriculum-mapping-workbench
Endpoints: /__curriculum-mapping-workbench/list
           /__curriculum-mapping-workbench/load?sourceLandscapeId=<id>
```

3.1 Seven-Layer Repository Data Flow

The workbench joins seven existing repository artifact layers without mutating registries:

1. **Source landscape registry (`curricula/DE/Gymnasium/provenance/source-landscape-registry.json`):** Identifies retained source snapshots by `landscapeId`, `jurisdiction`, title, and source path.
2. **Official PDF passage extraction (`source-extraction/*.source-extraction.json`):** For reviewed curricula (such as `DE_HE_MATHEMATIK_SEKII_KC2024.source-extraction.json` from `kerncurriculum_gymnasiale_oberstufe-mathematik.pdf`), `pdftotext` segments the PDF by topic fields (`E.1`, `Q1.3`, `Q4.2`), strips running headers/footers, normalizes German text to NFC against mojibake (`ä`, `ö`, `ü`, `ß`), and converts formulas into inline LaTeX for KaTeX rendering.
3. **Source goals:** When a persisted `source-extraction` file exists, source goals derive directly from literal official bullet aspects (retaining parent bullet text, `sourceSpan`, and PDF page); otherwise the view falls back to the retained snapshot (`*.de.json.snapshot`).
4. **Source-goal membership registry (`source-goal-membership-registry.json`):** Marks which source goals count as registered; unregistered source goals highlight visibly red.
5. **Source-goal closure registry (`source-goal-closure-registry.json`):** Decomposes broad original goals into the atomic source goals that close them.
6. **Mapping and review files (`curricula/DE/Gymnasium/mapping/...`):** Link `legacyGoalId` or extracted source goals (`exact`, `partial`, or explicit gap decisions in `*.review.json`) to `canonicalGoalId`.
7. **Canonical skill graph and composition view:** Pairs `curricula/DE/Gymnasium/canonical/*.de.json` with the matching learner-facing view under `curricula/DE/Gymnasium/composition-views/` so the right-hand tree matches what learners see.

3.2 Fail-Closed Pipeline Gates (`MAPPING-1` through `MAPPING-3`)

A downstream mapping stage must never be reported complete while any prerequisite stage remains incomplete or blocked (`pipelineStatus` inside `source-extraction`):

Gate	Meaning	Completion criterion
`MAPPING-1`	Official passages extracted	Every expected topic field is present exactly once with official bullet text and clean German/LaTeX display text.
`MAPPING-2`	Source goals created	`MAPPING-1` complete, every goal-bearing passage has at least one unique source goal with span/parent-bullet/ page reference and clean encoding (policy-only Leitfaden passages may stay non-goal-bearing).
`MAPPING-3`	Canonical mapping reviewed	`MAPPING-2` complete, every accepted source goal carries a reviewed decision targeting existing canonical goals and is substantively covered (`needs_canonical_goal` or `needs_ view_placement_review` show progress without closing `MAPPING-3`).

3.3 Two-Stage Interaction Model and Audit Interpretation

The page separates inspection into two dual-pane stages plus a bottom detail pane:

- **Stage 1 (Official PDF passage ↔ source goals):** Left pane lists topic-field PDF passages; right pane lists assigned source goals with bidirectional highlighting.
- **Stage 2 (Source-goal tree ↔ SkillPilot tree):** Left pane shows the source-goal tree with registration/mapping badges; right pane renders the learner-facing composition view with direct/descendant support counts and synchronized scroll focus; bottom pane inspects PDF page, source reference, closure atoms, and target nodes.

Green means registered/mapped — not content-approved

A green badge indicates only that a registration or mapping edge exists in current repository data. Reviewers must still verify that every original PDF goal was captured, broad goals have a sound atomic closure, canonical targets are semantically justified, and no Bundesland view exposes goals lacking legitimate syllabus backing.

Currently resolved Hessen audit entrypoints at `http://127.0.0.1:5174/curriculum-mapping-workbench`:

Subject	Stage	Source goals	Mappings	Official PDF in Workbench
Mathematik	Sek I	132	37	`g9-mathematik.pdf`
Mathematik	Sek II	316 persisted goals	316 reviewed / 383 edges	`kerncurriculum_gymnasiale_oberstufe-mathematik.pdf`
Physik	Sek I	60	53	`g9-physik.pdf`
Physik	Sek II	376	376	`kerncurriculum_gymnasiale_oberstufe-physik.pdf`

4 Curriculum Package Readiness (`full-standalone-v1`)

This gate answers a narrower, stronger question than legacy subject-export validation: can an artifact honestly claim the self-contained `full-standalone-v1` runtime profile? Reproducible subject ZIPs remain valuable handoff and roundtrip artifacts, yet do not prove standalone execution from the ZIP alone and therefore receive `not-ready-legacy` (`standaloneProfile Ready: false`).

4.1 Status Vocabulary and Exit Codes

Policy rules live in `contracts/curriculum-package/v1/profiles/full-standalone-v1.readiness-policy.json` and reports conform to `contracts/curriculum-package/v1/package-readiness-report.schema.json` (evaluator `1.3.0` validates its own output against that schema):

Status / condition	Exit	Semantics
`ready` (or exact `--expect-status` / self-test)	0	Every blocking check in the versioned policy passed (open publication/human gates still prevent `ready` for the current Mathematics candidate).
CLI usage error	2	Invalid arguments or missing required flags.
`invalid`	10	Input safety, target identity, or contract binding failed (partial target claims never downgrade to legacy).
`not-ready-incomplete` / `not-ready-legacy`	20	Target contract consistently declared while completeness, publication/redistribution, or consumer gates remain open (`not-ready-incomplete`), or legacy subject-export artifact (`not-ready-legacy`) without matching `--expect-status`.
`unsupported`	30	Internally consistent target claim for a contract/compatibility version unsupported by this evaluator.
`--expect-status` mismatch	40	Evaluated status differs from the explicitly expected status.
Internal / contract failure	70	Internal evaluator error or corrupted trusted contracts.

4.2 Conformance Suites, Build, and Hermetic Package Consumer

Running the self-tests and contract suites from the repository root:

```
python3 -B scripts/validate_curriculum_package_contracts.py
python3 -B scripts/validate_curriculum_runtime_catalog_contract.py
python3 -B scripts/validate_curriculum_schema_catalog_contract.py
python3 -B scripts/evaluate_curriculum_package_readiness.py --self-test
python3 -B scripts/validate_full_standalone_curriculum_package.py --self-test
python3 -B scripts/run_package_consumer_smoke.py --self-test
python3 -B scripts/validate_fwu_owl_curriculum_package.py --self-test
python3 -B scripts/reconstruct_json_curriculum_package_from_fwu_owl.py --self-test
python3 -B scripts/run_fwu_owl_reverse_compiler_hermetic.py --self-test
python3 -B scripts/validate_curriculum_fwu_owl_reverse_compilation_contract.py \
  --self-test
python3 -B scripts/generate_curriculum_package_redistribution_review.py --check
python3 -B scripts/generate_curriculum_source_verification_review.py --check
```

Building (`app/scripts/buildFullStandaloneCurriculumPackage.ts`), independently validating (`scripts/validate_full_standalone_curriculum_package.py`), and evaluating a provisioned staging candidate (or running the full sequence via `scripts/run_curriculum_full_package_conformance.sh`):

```
npm --prefix app run export:full-standalone-package -- \
  --release-root tmp/curriculum-release-model/mathematik-a \
  --output-dir tmp/curriculum-release-model/full-standalone-package \
  --zip --expect-entry-count 897 --expect-manifest-file-count 895 \
  --expect-binary-asset-count 715 \
  --expect-content-digest \
  sha256:9d3c0e03f95c06161abd0944335807704ffef1fcf20433f52d83167015637c64

PKG_DIR=tmp/curriculum-release-model/full-standalone-package
ZIP_NAME=skillpilot-curriculum-de-gymnasium-mathematik-0.1.0-conformance.3.json.zip
python3 -B scripts/evaluate_curriculum_package_readiness.py \
  --zip "$PKG_DIR/$ZIP_NAME" \
  --consumer-smoke-store tmp/curriculum-release-model/provisioned-store \
  --consumer-smoke-work-dir tmp/curriculum-release-model/package-consumer-smoke \
  --consumer-smoke-report "$PKG_DIR/package-consumer-smoke-report.json" \
  --report "$PKG_DIR/readiness-report.json" \
  --expect-status not-ready-incomplete
```

The evaluator invokes both the independent full-package validator v2 and the policy-pinned consumer runner (`contracts/curriculum-package/v1/package-consumer-smoke-report.schema.json`) directly; without `--consumer-smoke-store`, any pre-existing report is treated strictly as `external-unattested`. The consumer builds a runtime-catalog-only frontend (`publicDir: false`) and slim backend free of embedded curriculum UUIDs or static domain payloads, executing inside `bubblewrap --clearenv` with a hidden checkout, loopback-only networking, read-only store, `O_NOFOLLOW`-rebound JDK/Python/Node/`strace`, and real Playwright 1.59.1/Chromium probes (including 5 poison sentinels, 15 ordered functional checks, and a fail-closed UI marker on Catalog-404 with zero fallback requests).

4.3 Offline Trust Model, Review Separation, and Implementation Boundary (DPK-004a–DPK-008d)

- **Offline resolution and explicit JSON caps:** The package-local schema catalog is strictly a hash-bound lookup table beneath the consumer's supported contract version (remote retrieval forbidden; max `64 MiB` per JSON entry, nesting depth `128`, max `5,000,000` parsed nodes). Likewise, the runtime catalog resolves roots, scopes, and views explicitly without filename heuristics.
- **Structural validity vs. redistribution clearance:** Files with `redistributionStatus: review-required` must carry `licenseExpression: null`, causing `publication.redistribution-cleared` to fail (`not-ready-incomplete`). `allowed` requires a concrete license expression resolved to an inventoried license file (see 10 for the 715 Mathematics images, 3 open non-binary classes, and 479 open source-text decisions).
- **Completed technical milestones:** `DPK-004a/004b` and `DPK-005a/005b` establish the release model, 22 pinned payload schemas, and deterministic streaming ZIP32; `DPK-006a–006c` enforce the content-addressed store and CAS lock without fallback; `DPK-007` proves the real 1.7-GB consumer path (15 checks, 213 frontier goals). The historical `DPK-007a` freeze (`skillpilot-curriculum-de-gymnasium-mathematik-0.1.0-conformance.3.json.zip`, `914` entries, `757` images, `1,738,161,217` bytes, SHA-256 `403cc0bc6004da549c8b9ed9fafad222fe0ddda1107806fe087cfa871a6dbcf9`, `contentDigest sha256:e83936aaf3645ff5f6e8132c4a801bd4bd66f55d3c0304a5deda3d6a5d194101`), the independent

2.36-GB FWU-OWL carrier `DPK-008c` (`819` entries, `824,452` triples, SHA-256 `abab1d8a ac3e9394af26c614bbf231954ba45ab11f725dd0f93f088820dc3f94`, placement key `goal Id@unitId@index`), and the isolated reverse compilation `DPK-008d` (`1,737,571,471` bytes, SHA-256 `7dcd233dd495900f6d6bd971ff6e86bdcdcfe5701f3522ed480be8336de23195`, 111/ 111 normal-form oracles, 20-file Rawtrace evidence inventory) remain immutable historical receipts without preempting any human review gate.

5 Curriculum Release Model Conformance

DPK-004 compiles the real German Gymnasium Mathematics authoring state into a strict, unpacked release model (DPK-004a for runtime payloads, DPK-004b for authoritative publication evidence) as the executable bridge to the full-standalone-v1 JSON ZIP (4). DPK-004 through DPK-007a are technically passed.

5.1 Overview and Trusted Inputs

Never inferring release semantics from filenames or goal titles, the compiler binds four explicit trusted inputs:

1. **Release-model build profile** (contracts/curriculum-package/v1/profiles/de-gymnasium-mathematik-release-model-v1.profile.json): Pins source/output paths, package identity, views, decks, resource policy, path relocations, contracts, and exact counts (1 landscape, 1,147 goals, 88 views, 12 decks, 128 cards, 784 resource links, 715 active images totaling 1,549,052,741 bytes).
2. **Ontology profile** (contracts/curriculum-package/v1/profiles/de-gymnasium-mathematik-v1.profile.json, version 1.1.1): SHA-256-pinned; binds the Core-first mapping to repository, checkout path (tmp/lehrplan-ontologie), commit, src/ontology/components/lehrplan-core.owl, ontology IRI, and file digest.
3. **Semantic-kind ledger** (curricula/DE/Gymnasium/quality/release-model/mathematik.semantic-kinds.json): Current source fingerprint for every one of the 1,147 goals without title/ID heuristics.
4. **Field-semantics registry** (contracts/curriculum-package/v1/profiles/skillpilot-fwu-field-semantics-v1.registry.json): 456 explicit entries (325 runtime, 130 publication, 1 generic legacy entry; SHA-256 d19497f0b18b0c7733dcaf6f5dc63590352814c2d764c858b333751b5af25e9d), normalization profile, and closed Draft 2020-12 schemas (validated bidirectionally with zero dead registry paths; field-coverage.json separates observed instance coverage from schema-only entries).

Semantic kind (semanticKind)	Count	Role in the Mathematics release model
curricularAtomic	780	Atomic domain competencies (711 with active image, 69 provider-deferred, 152 human-approved)
curricularArea	218	Curricular topic and area clusters
practiceAssessment	129	Practice, exam, and Abitur task nodes
programStructure	7	Program-structure containers
memory	6	SRS / memorization goals
runtimeSupport	5	Technical runtime support nodes
orientation	2	Orientation and motivation anchors

5.2 Generated Model Layout, Digests, and Publication Evidence

Output is promoted atomically via a private staging directory strictly beneath tmp/ (conformance-model-only-not-a-package, rejecting symlinks; binary image bytes materialize only in the companion ZIP):

```
data/
  canonical/mathematik.landscape.json
  views/index.json & views/*.json
  cards/card-index.json & cards/*.json
  resources/resource-index.json
  mappings/source-to-canonical.json
  sources/source-index.json & sources/source-goal-references.json
  runtime/{catalog,dependency-closure,migration-aliases}.json
  assessment-sources/...
metadata/
  quality/release-quality-evidence.json
  {semantic-content-index,field-coverage,build-inputs,release-model-conformance}.json
```

- **Dependency closure and digests:** `dependency-closure.json` captures `2,429` runtime definitions and `20,523` hard references with zero external dependencies (`sha256:5d542bfb46658aca0f570fa21d4771008d3238ab4aa87e5a551d55ed9efc3bc6`; definition index `sha256:1af9cf9fecdcdfec27372c1fc57a700634efb998755070ffd84dff0d1cff0680`). `migration-aliases.json` uses an explicit `initial` baseline for the first release. `semantic-content-index.json` binds `111` normalized logical records and `715` image records path-neutrally into `contentDigest sha256:9d3c0e03f95c06161abd0944335807704ffef1fcf20433f52d83167015637c64` (preserving the historical `.3 DPK-007a/008` freeze untouched). When a prerequisite-reduction registry is declared, both compiler and validator independently run the Node proof verifier (`.nvmrc`) before comparing fingerprints.
- **Four publication roles (`runtimeRequired: false`):** `mapping` (`source-to-canonical-mappings`: `10,021` decisions → `33,234` exact/partial edges without unsafe `missing => exact` fallbacks), `source-index` (`official-source-index`: `31` collections, `55` documents — `52` Core `Lehrplan` + `3` `OfficialSourceDocument` — across `16` jurisdictions), `source-goal-reference-index` (`9,977` reviewed source goals typed as Core `Curriculares Element` with `CE-Verweis`; C0 controls and unpaired UTF-16 surrogates fail closed), and `quality-evidence` (`release-quality-evidence`: `2,335` decisions spanning `780` atomicity, `780` memory-goal, `64` active-card, and `711` active-image records; `publicationStatus: not-ready`).

5.3 Verification Commands and Hosted PDF Mode

Compiling, running the independent 49-case validator, checking fixtures (9 positive, 22 negative, remote schema retrieval disabled), and running the required JSON conformance wrapper (stopping before the 1.7-GB ZIP):

```
PROFILE=contracts/curriculum-package/v1/profiles
PROFILE="$PROFILE/de-gymnasium-mathematik-release-model-v1.profile.json"
python3 -B scripts/compile_curriculum_release_model.py \
  --profile "$PROFILE" \
  --output tmp/curriculum-release-model/mathematik-a
python3 -B scripts/validate_curriculum_release_model.py \
  --profile "$PROFILE" \
  --release-root tmp/curriculum-release-model/mathematik-a
python3 -B scripts/validate_curriculum_release_model_fixtures.py
python3 -B scripts/compile_curriculum_release_model.py \
  --self-test-dependency-emission
bash scripts/run_curriculum_release_model_conformance.sh
```

By default, referenced local source PDFs must be regular non-symlink files hashed byte-for-byte. Because hosted GitHub checkouts omit gitignored official PDFs, CI workflows explicitly set `SKILLPILOT_SOURCE_PDF_MODE=committed-bindings`: only genuinely absent PDFs may fall

back to the byte count and SHA-256 already recorded in the committed source-verification ledger, while present, modified, or symlinked files still fail closed.

6 Dual Curriculum Release Contracts

DPK-003 defines the fail-closed semantic contract pairing a SkillPilot JSON runtime package with a Core-first FWU ontology package, without by itself publishing either production artifact.

6.1 Binding Contract Invariants

- **Total field-registry coverage:** Every release-model field must match exactly one versioned registry entry (scalars, ordered lists, sets, maps, binary references, generated non-semantic fields). Unknown fields block the ontology release.
- **Explicit zero-based ordering in RDF:** Goals, authored direct `contains` and `requires` edges, resource links, view children, cards, and scoring steps are ordered. RDF carries unique, contiguous zero-based positions because unordered Core relations cannot reconstruct authored order.
- **Strict value distinction & ban on broad JSON carriers:** Missing, `null`, empty, and default values remain distinct unless a specific registry entry states otherwise. Known curricular concepts use the FWU Core; SkillPilot vocabulary is restricted to runtime semantics and exact reverse reconstruction. Any canonical JSON literal requires a field-specific Core gap, granularity, and byte cap; package- or landscape-wide JSON carriers are forbidden.
- **Path-neutral binary identity:** Binary resources enter the shared index via stable resource identity, canonical public reference, MIME type, byte length, and SHA-256; variant-specific ZIP paths never alter `contentDigest`.
- **Equivalence report & signing projection:** A passing equivalence report binds both final ZIPs and inner manifests, trusted contracts, FWU Core IRI, toolchain, field coverage, role digests, graph/view/card/asset comparisons, isolated reverse compilation, validation of original and reconstructed JSON packages, hermetic consumer runs, and reproducibility. The external release index strips the signature block before canonical hashing (avoiding self-reference), and the DPK-003 validator rejects every `verified` or `stable` claim until DPK-011 performs cryptographic signature, identity, and provenance checks (`staging` may remain explicitly unsigned).

6.2 Executable Gate and Boundary

Run from the repository root (part of the optional ontology lane in `./run_ci.sh owl`, `./run_ci.sh full`, and `.github/workflows/owl-ci.yml`; requires `jsonschema==4.26.0`):

```
python3 -B scripts/validate_curriculum_dual_release_contracts.py
```

The validator rejects duplicate JSON keys, non-finite numbers, invalid Unicode, and RDF/XML-unsafe control characters, validates all schemas and positive fixtures, recomputes logical/binary records and the aggregate `contentDigest`, and runs a mutation matrix covering lost ordering, ambiguous paths, blurred `missing`/`null` semantics, broad JSON carriers, forged digests, substituted reports, non-isolated reconstruction, and unverified `stable` promotion.

Synthetic contract fixtures are never curriculum evidence

Positive release and equivalence files under `contracts/curriculum-package/v1/fixtures/dual-release/valid/` carry synthetic hashes and counts strictly for contract testing and must never be published as curriculum evidence.

7 Requires Relation Checks

This chapter serves as a stable compatibility entry point for references to structural relation checks.

The maintained normative rule catalog lives in Graph Validation Rules (`GVR/APV/CPV`) in Volume IV of the Quality Handbook, which is the single source of truth for:

- all implemented validator rules (`GVR-*`, `APV-*`, `CPV-*`),
- the strict default CI enforcement profile (`npm run validate:graph`),
- the local legacy-warning rollout mode (`VALIDATE_GRAPH_STRICT_RULES=0 npm run validate:graph`).

For manual didactic inspection of prerequisite step sizes and inherited edges, see the Requires Review Process in 9.

8 Atomic Review Process

This chapter documents the repeatable manual authoring and QA workflow for verifying whether existing leaf goals (`contains: []`) of a skill landscape represent genuinely atomic didactic steps and splitting bundled goals cleanly (complementing the ledger-enforced `CQR-301` gate in Semantic Atomicity Review).

8.1 Scope and Working Definition

Reviews inspect one landscape JSON file per pass across goals with `contains: []`. Goal kinds governed by dedicated QA lanes are intentionally **excluded**:

- memorization goals tagged `memorization` or `srs-deck:*`,
- exam or task-bundle goals carrying `examData`,
- motivation or orientation goals (`Motivation`, `Orientation`).

A domain goal counts as **atomic** when all four criteria hold:

1. It states **one** primary competence action.
2. It is realistically assessable in **1–3 tasks**.
3. It represents **one small didactic step** relative to its prerequisites.
4. It hides no independent sub-skills inside one umbrella sentence (warning signs: multiple verbs at distinct cognitive levels, multiple content subdomains, or separate proof obligations).

8.2 Seven-Step Review Workflow and Decision File

`app/scripts/generateAtomicReport.ts` (`npm run report:atomic`) prepares the decisions scaffold and findings report (`../tmp/atomic_findings_<curriculum>.md` containing summary counts, `not_ok` findings, and a full appendix):

```
cd app
npm run report:atomic -- \
  --input ../curricula/<...>/json/<file>.json \
  --decisions ../tmp/atomic_decisions_<name>.json \
  --init-decisions
```

1. Initialize or update the decisions scaffold.
2. Decompose each leaf goal semantically (action verbs, content objects, expected competence proof).
3. Judge whether splitting into 2–4 goals materially improves clarity and assessment (mark `not_ok` only for genuine didactic precision gains, never for wording aesthetics alone).
4. Record `ok`, `not_ok`, or `pending` in the JSON decisions file:

```
{
  "goals": {
    "<goalId-a>": { "status": "ok" },
    "<goalId-b>": {
      "status": "not_ok",
      "problem": "why the goal is bundled",
      "proposal": "how to split it into 2-4 goals"
    },
    "<goalId-c>": { "status": "pending" }
```

```
  }
}
```

1. For `not_ok` goals, specify 2–4 replacement goals with updated `title`, `description`, `requires`, and parent `contains`.
2. After applying accepted splits, check acyclicity, reference integrity, and weights (atomic goals normally retain `weight = 1`; cluster weights reflect unique atomic descendant counts).
3. Run `cd app && npm run validate:graph`.

9 Requires Review Process

This chapter documents the repeatable manual authoring and QA workflow for reviewing prerequisite quality on atomic learning goals (`contains: []`): verifying that the didactic jump from prerequisites to target is teachably small and replacing coarse or misleading edges with clean atomic routes.

9.1 Transition Semantics vs. Canonical Target State

`app/scripts/generateRequiresReport.ts` (`npm run report:requires`) exposes both layers of the current compatibility model together with their origin:

- **`direct requires`:** Prerequisites authored directly on the atomic goal itself.
- **`effective requires`:** Direct prerequisites plus prerequisites inherited from ancestor clusters via `contains`.

While inspecting inherited origins is essential during migration, the binding target state remains a purely atomic prerequisite relation $R \subseteq A \times A$ (Graph Definition).

9.2 Six-Step Review and Repair Workflow

```
cd app
npm run report:requires -- \
  --input ../curricula/<...>/json/<file>.json \
  --decisions ../tmp/requires_decisions_<name>.json \
  --init-decisions
```

1. Initialize or update the decisions scaffold (default output: `../tmp/requires_findings_<curriculum>.md` containing summary counts, `not_ok` findings, and a full appendix listing `direct requires`, `effective requires`, and source clusters/goals).
2. Inspect direct and effective prerequisites plus inheritance origins for each atomic goal.
3. Formulate the **didactic delta**: what can the learner already do after the prerequisites, what is genuinely new in the target goal, and is that delta a single teachable step?
4. Record `ok`, `not_ok`, or `pending` in the JSON decisions file (mark `not_ok` only for real didactic defects — oversized jump, unrelated blocking prerequisites, or cluster edges hiding atomic dependencies — rather than graph neatness alone):

```
{
  "goals": {
    "<goalId-a>": { "status": "ok" },
    "<goalId-b>": {
      "status": "not_ok",
      "problem": "what is didactically wrong",
      "proposal": "how to repair the requires structure"
    },
    "<goalId-c>": { "status": "pending" }
  }
}
```

1. For `not_ok` goals, apply one of four standard repairs: (a) add a bridge goal, (b) split the target goal (8), (c) replace cluster-level `requires` with specific atomic prerequisites, or (d) remove redundant or over-blocking edges.

2. Re-run `cd app && npm run validate:graph` after accepted edits.

10 Human Review Gates for Curriculum Releases (HR-001–HR-006)

This chapter consolidates the six strictly independent human sign-off gates that the automated toolchain is forbidden to decide on its own (initially scoped to `org.skillpilot.curriculum.de.gymnasium.mathematik`; any hash or content drift automatically marks affected prior approvals stale).

10.1 Overview of the Six Independent Gates

Gate	Open scope	Blocks	Primary work source
`HR-001` Visual quality	`559` active images open (`152` approved; `69` of `780` goals provider-deferred)	Domain publication readiness	`curricula/DE/Gymnasium/quality/goal-visualization-qa/mathematik.qa.json`
`HR-002` Image rights	`715` per-asset decisions (`708` provider pipeline, `6` user claim, `1` deterministic SVG render)	Public redistribution	`curricula/DE/Gymnasium/quality/package-redistribution/de-gymnasium-mathematik-v1.review.json`
`HR-003` Non-binary rights	`3` class decisions (`skillpilot-data`, `official-source-evidence`, `generated-metadata`)	Public redistribution	Same redistribution ledger
`HR-004` Source-text verification	`479` record decisions (alongside `9,498` technical matches)	Source QA / promotion	`curricula/DE/Gymnasium/quality/source-verification/de-gymnasium-mathematik-v1.review.json`
`HR-005` Ontology sign-off	`1` hash-bound domain/ontology and legal expert sign-off	Ontology promotion	`DPK-008c` FWU receipt & `DPK-008d` reverse receipt
`HR-006` Signature & `stable`	`1` trust/release decision per release group	Public `stable` release	Future `DPK-011` release index

10.2 Domain, Rights, and Source-Text Gates (HR-001 through HR-004)

- **`HR-001` (Domain visual sign-off):** Verifies mathematical accuracy of all values/relations, fit to learning goal and age group, legibility, language, and alt text on the exact asset hash (`human-approved`, `rejected`, `deferred-provider-limitation`).
- **`HR-002` (Rights across 715 images):** Every embedded image defaults to `pending-human-review`, `review-required`, and `licenseExpression: null` (`708` provider-pipeline claims, `6` `user-provided-generated-claim` records requiring uploader attestation, `1` deterministic render via `librsvg rsvg-convert` 2.52.5 from reviewed SkillPilot SVG). Provenance labels never grant licenses; clearance requires provider/license verification, an SPDX-style `licenseExpression`, evidence, reviewer, and timestamp covering all 715 records (unprovable assets become `prohibited` and must be removed/replaced prior to public release).
- **`HR-003` (Non-binary artifact rights):** Only `software-contracts` (bound to the root `LICENSE` hash) clears automatically as Apache-2.0. Open classes are (1) `skillpilot-data`, (2) `official-source-evidence` (distinguishing bare facts/metadata, original paraphrase, and protected wording), and (3) `generated-metadata`.

- **HR-004 (479 source-text decisions):** Alongside 9,493 contiguous authored-passage matches and 5 ephemeral PDF-projection matches (9,498 technical matches that carry no legal approval), 479 records require human classification bound via reviewEvidenceSha256 to sourceText, passage, PDF, and locator (verified-verbatim, verified-normalized-transcription, reviewed-paraphrase, rejected; per-document breakdown in source-verification-de-gymnasium-mathematik-v1.md).

10.3 Ontology Sign-Off and Stable Promotion (HR-005 & HR-006)

- **HR-005 (Core-first ontology sign-off):** Hash-bound expert sign-off across the DPK-008c FWU-OWL ZIP (2,362,455,128 bytes, SHA-256 abab1d8aac3e9394af26c614bbf231954ba45ab11f725dd0f93f088820dc3f94, manifest 29f308424d1aeba9095f0e800253acadcdfaca0562dfa1fc37741c77c76023b3), reconstructed JSON ZIP (1,737,571,471 bytes, SHA-256 7dcd233dd495900f6d6bd971ff6e86bdcdcfe5701f3522ed480be8336de23195, manifest 8d7970435431ff78743d0bf413a54bbebee009bbaafe35f1283ebcb89b4f2ff0), DPK-008d reverse receipt (12,185 bytes, 94035f2d88e2c2b7588f13f01cb44b917c83900b9f839c66558ea7eae585884f), 20-file evidence manifest (3,483 bytes, 31c256ff612aa7a346fb0792c83b109d4bc01df52fdc36e98a8c717bf9b8eafc, inventory digest 8483f5ce428cc54cca1353668370ae09f32f51166af6a3235a2ff19ccb88ea1e), and 454-entry registry (2e536c3f8d63e2acf45690375ace69ec0c6a6e92787bc8a16957b80120c4ca48). Reviewers confirm semantic fidelity of every FWU Core term, strict minimality of extensions and JSON literals, clean layer separation, and license/Share-Alike duties (CC-BY-SA-4.0; until sign-off, the bundled Core file keeps review-required and licenseExpression: null).
- **HR-006 (Signature and stable promotion after DPK-011):** Approves the trusted signing identity and verifies that all prior human gates are closed on the bound hashes before atomically promoting the JSON/OWL release group from staging to stable.

11 MEM SPARQL Consistency Audit

This deliberately **non-blocking** QA lane compares live curriculum data from the external MEM/FWU ontology endpoint (`https://sparql.mem.edufeed.org/sparql`) against SkillPilot's retained `source-extraction` evidence. Because external endpoint coverage across German states remains incomplete, discrepancies act as structured triage signals rather than CI failures; MEM never replaces official source references or `CQR-*`, atomicity, memory-card, and composition-view validators.

11.1 Sources of Truth, Generated Artifacts, and Execution

- **Config & ledger (source of truth):** `curricula/DE/Gymnasium/quality/mem-sparql-consistency/canonical-math-poc.config.json` and `canonical-math-poc.review.jsonl` (generator: `app/scripts/generateMemSparqlConsistencyAudit.ts`).
- **Generated snapshots (never hand-edited):** mem-sparql-consistency-audit.md, `mem-sparql-consistency-audit.json`, mem-sparql-consistency-review-issues.md, and `mem-sparql-consistency-review-issues.json`.

```
cd app
npm run quality:mem-sparql-consistency
npm run quality:mem-sparql-consistency -- \
  --endpoint=https://sparql.mem.edufeed.org/sparql
MEM_CFG=curricula/DE/Gymnasium/quality/mem-sparql-consistency
npm run quality:mem-sparql-consistency -- \
  --config="$MEM_CFG/canonical-math-poc.config.json"
```

The command exits `0` even when open triage items exist, failing only on local script or configuration errors (initial PoC record: MEM SPARQL PoC, 2026-06-01; operational checklist: 12).

11.2 Fingerprint-Bound Review Ledger and Expansion Criteria

Every JSON review-queue entry carries an `itemFingerprint` computed from current MEM and local evidence:

```
{
  "schemaVersion": 1,
  "auditId": "canonical-math-poc",
  "itemId": "triage-a3e85d9ef77a",
  "itemFingerprint": "sha256:...",
  "status": "mem_feedback_candidate",
  "reviewedAt": "2026-06-01",
  "reviewer": "initials-or-tool",
  "reason": "Short evidence-based decision.",
  "followUpRef": "optional issue, PR, or MEM feedback reference"
}
```

- **Supported `status` values:** `not_an_issue`, `local_extraction_fix_needed`, `mem_feedback_candidate`, `source_version_gap`, `deferred`.
- **Per-run ledger validation:** Matching `itemId` + `itemFingerprint` annotates the entry as `decided` (except `deferred`, which stays open); drifted fingerprints surface as `stale_review`; unknown, duplicate, or malformed records appear under `reviewLedger.diagnostics`.

- **Expansion criteria:** Add a scope only when MEM exposes concrete plan data (beyond vocabulary), graph and class IRIs are known, SkillPilot holds a retained `source-extraction` artifact for the same scope, and stable issue IDs can be triaged without blocking `CQR-*` readiness rules.

12 MEM SPARQL Consistency Runbook

This runbook provides the operational checklist for executing and triaging the MEM SPARQL consistency audit defined in 11 (initial PoC investigation: MEM SPARQL PoC, 2026-06-01).

12.1 Running the Audit and Inspecting the Queue

```
cd app
npm run quality:mem-sparql-consistency
cd ..
node - <<'NODE'
const q = require('./docs/qa-ci/status/mem-sparql-consistency-review-issues.json')
console.log(q.summary)
console.log(q.reviewLedger)
console.log(q.highestSignalChecks.map((i) => ({
  id: i.id, status: i.status, category: i.category,
  side: i.side, mem: i.memText, local: i.localText,
})))
NODE
```

For human scanning, inspect mem-sparql-consistency-review-issues.md (queue) and mem-sparql-consistency-audit.md (jurisdiction availability and counts).

12.2 Recording Decisions, Interpreting Diagnostics, and Verifying Locally

1. Pick an item from `highestSignalChecks` or `textTriageQueue` and copy its `id` and `itemFingerprint`.
2. Inspect `memText`/`memRef`, `localText`/`localRef`, `rationale`, and any `reviewNote`; fix local extraction defects directly in the `source-extraction` file.
3. Choose a status (`not_an_issue`, `local_extraction_fix_needed`, `mem_feedback_candidate`, `source_version_gap`, `deferred`), append a single-line JSONL object to `curricula/DE/Gymnasium/quality/mem-sparql-consistency/canonical-math-poc.review.jsonl`, and re-run the audit.

Interpreting `reviewLedger.diagnostics` (clean before a MEM-team feedback pass; use `tmp/` for synthetic ledger tests):

- `stale_review`: Item still exists, but its current evidence fingerprint drifted → re-inspect before reusing.
- `unknownItemRecords`: Ledger references an `itemId` absent from the current queue.
- `duplicateItemRecords`: Multiple records target the same `itemId` (last record wins; deduplicate).

Local verification after editing the generator, config, extraction files, or docs:

```
cd app
npx eslint scripts/generateMemSparqlConsistencyAudit.ts
npm run quality:source-coverage-audit:check
cd ..
git diff --check
MEM_CFG=curricula/DE/Gymnasium/quality/mem-sparql-consistency
node -e "for (const f of [
  '$MEM_CFG/canonical-math-poc.config.json',
```

```
  'docs/qa-ci/status/mem-sparql-consistency-audit.json',
  'docs/qa-ci/status/mem-sparql-consistency-review-issues.json'
]) JSON.parse(require('fs').readFileSync(f, 'utf8')); console.log('json ok')"
```

13 Goal Source Rationales Runbook

This runbook documents generation, runtime serving, and verification of human-readable source rationales (`generateGoalSourceRationales.ts`) along the reviewed classic chain `canonical goal -> mapping review -> source extraction -> official source document` (concept specification in `docs/concept/skill-graph/human-readable-source-rationales.md`; status registry in status/README.md).

13.1 PoC, Custom Goal Sets, and MEM/FWU Showcase Commands

Run from `app/` for the Bavaria Mathematics PoC (goal-source-rationales-poc.md + `.json`), custom goal selections, and the Bavaria LehrplanPLUS live MEM/FWU showcases (`:plain` replaces UUIDs with readable SkillPilot paths and embeds concrete SPARQL queries):

```
cd app
npm run quality:goal-source-rationales:poc
npx tsx scripts/generateGoalSourceRationales.ts \
  --jurisdiction=DE-BY --goals=<goal-id>,<another-goal-id>
npm run quality:goal-source-rationales:mem-examples
npm run quality:goal-source-rationales:mem-examples:plain
```

Key CLI flags: `--landscape=<path>`, `--mapping-root=<path>`, `--output-json=<path>`, `--output-md=<path>`, `--no-md`, `--goals=source-backed | source-backed-relevant-leaves | all-relevant-leaves`, `--jurisdiction=all`, `--include-mem`, and `--mem-config=<path>`.

13.2 Runtime Public Index (Repository vs. Package Mode) & Deployment Smoke

- **Repository mode:** The Explorer source-rationale button lazily fetches two compatibility indexes kept outside the JS bundle and service-worker precache: `/data/goal-source-rationales-math-public.json` (preferring `DE-BY` so aligned MEM/FWU examples stay visible) and `/data/goal-source-rationales-physics-public.json` (never intercepted by the generic deck route).
- **Package mode:** `/data/**` is disabled; the UI discovers bindings via Catalog API 1.2 (`/api/ui/curriculum-catalog`) and fetches generation-bound per-goal evidence from `/api/ui/curriculum-source-evidence/packages/{packageId}/{packageVersion}/goals/{goalId}` with zero repository/classpath fallback.

Regeneration, build-artifact verification, and live deployment smoke checks (invoked automatically by `./deploy_skillpilot.sh` after service restart, auto-detecting `404` repository mode vs. Catalog API `1.2` package mode):

```
cd app
npm run quality:goal-source-rationales:public
npm run check:goal-source-rationales:math-public
npm run check:goal-source-rationales:build-artifact
npm run smoke:goal-source-rationales:deployment
npm run smoke:goal-source-rationales:deployment -- --base-url=http://127.0.0.1:8080
cd ..
SKILLPILOT_BASE_URL=https://staging.example.org \
  ./deploy_skillpilot.sh --coach-variant openai-mcp
```

13.3 Review Checklist and Follow-Up Issues

Inspect five items on every rendered goal:

1. `sourceRationaleStatus:` `classic_source_reviewed` (ready for public index), `classic_source_partial` (usable only when the rationale explains the partial fit), or `classic_source_gap` (open a non-blocking follow-up issue).
2. **Reaching the original source:** Official URL or retained source path plus a concrete section/search locator is present.
3. **Extracted source evidence:** Document metadata resolves, `sourceGoalId` and reference match the mapping decision, and the Markdown excerpt stays concise (full text remains in JSON).
4. **Why the SkillPilot goal is justified:** Understandable mapping rationale that never overstates `partial`, `split`, or `aggregate` mappings.
5. **MEM/FWU SPARQL route:** `mem_sparql_unavailable` for classic-only PoC runs and `mem_sparql_consistent` for aligned Bavaria showcase items (treated strictly as a secondary route until a scope is approved as primary-ready; file a manual review issue on disagreements or missing metadata).

14 Abi 2026 Mathematics DeepLink MVP (BlueSky)

This chapter specifies the URL contract, automatic Cockpit provisioning flow, campaign telemetry events, and QA test links for the Hessen 2026 Mathematics Abitur deep-link MVP.

14.1 URL Contract and Cockpit Pre-Provisioning

The landing route `/start/abi26-he-mathe-k1` accepts `source` / `utm_source`, `campaign` / `utm_campaign`, `medium` / `utm_medium`, and `track` / `courseLevel` / `f` (`GK` or `LK`, defaulting to `GK`):

```
/start/abi26-he-mathe-k1?source=bluesky&campaign=abi26-mathe-k1&medium=social&track=GK
```

Clicking `Kostenlose SkillPilot-ID erstellen` executes four steps:

1. Creates a fresh pseudonymous SkillPilot ID (`POST /api/ui/learners`).
2. Configures the learner's curriculum for Hessen upper-secondary Mathematics (`PUT /api/ui/learners/{id}/curriculum`).
3. Sets the initial scope to `Abiturprüfung Mathematik` (`9ad83149-3cb7-5b87-a617-3eae3715a50c`) for `GK` or `Abiturprüfung Mathematik (LK)` (`464a6024-a2f8-53b4-84e0-d7b9df22a0b1`) for `LK`.
4. Opens the Cockpit carrying campaign parameters (`l` targeting `Gymnasium (DE)`, state filter `f = DE-HE`, and course level `GK` or `LK`).

14.2 Campaign Events and QA Verification (3 Test Links)

Campaign events are buffered client-side in `localStorage` (`skillpilot_campaign_events`) and posted to `POST /api/ui/events`: `page_view`, `id_created`, `cockpit_opened`, `gpt_prompt_copied`, and `gpt_start_clicked`.

Verify all three QA test links:

```
1. GK:       /start/abi26-he-mathe-k1
               ?utm_source=bluesky&utm_campaign=abi26-he-mathe-k1
               &utm_medium=social&track=GK
2. LK:       /start/abi26-he-mathe-k1
               ?utm_source=bluesky&utm_campaign=abi26-he-mathe-k1
               &utm_medium=social&track=LK
3. Fallback: /start/abi26-he-mathe-k1
               ?source=bluesky&campaign=abi26-he-mathe-k1
               &medium=social&track=foo
```

Expected behavior: invalid `track=foo` visibly falls back to `GK`, pseudonymous ID creation succeeds without registration, the Cockpit opens with pre-selected scope and visible coach/GPT launch card, and the feedback notice is visible on both landing page and Cockpit.