SkillPilot Architecture Handbook · Volume Ia

Skill Graph Architecture, Media, Goal Book, and Package Interchange

Layered Goal System, Node Types, View Projection, Source and Content Integration,

Visualizations, Learning-Goal Book Pipeline, Gymnasium Rollout, and Dual Package Releases

Scope of this volume

Complementing the formal Graph Definition (Volume I), Volume Ia consolidates the architectural, media, publication, and interchange contracts surrounding the skill graph: the layered general goal system and incremental migration, pedagogical node types, deterministic view projection and goal-placement semantics, source vs. external content-package separation (Issue #50), human-readable source rationales, atomic goal visualizations, the learning-goal book and multi-AI evidence-review pipeline, the canonical German Gymnasium rollout policy, and MEM/FWU ontology roundtrips plus dual curriculum package releases.

As of 11 November 2026 · License: Apache-2.0 (technical architecture and specification documentation; see LICENSING.md).

First-party skill landscapes and didactic media are licensed separately under CC-BY-4.0; external resources retain their upstream licenses.

Contents

1 General Goal System and Incremental Migration **3**
1.1 Four Semantic Layers Plus One Composition-View Layer . . . 3
1.2 Recursive Learning Units and Incremental Migration . . . 3

2 Node Types **5**
2.1 Outcome Contract and Compiled Applicability . . . 5
2.2 Orientation, Understanding, and Memorize Semantics . . . 5
2.3 Exam Nodes, Recognition Review, and Scoring . . . 6

3 View Projection and Goal Placement **7**
3.1 Four View Families and Five Non-Negotiable Projection Rules . . . 7
3.2 Prerequisite Closure, Personal Learning Scope, and Canonical Identity . . . 7
3.3 Learning Units in Projected Views . . . 8

4 Source and Resource Links **9**
4.1 Three-Way Purpose Split and Hard Canonical Prohibition . . . 9

5 Content Integration: Architectural Framework **10**
5.1 Three-Layer Separation and Personal Selection . . . 10
5.2 Physics Pilot and Four Acceptance Criteria . . . 10

6 Human-Readable Source Rationales **11**
6.1 Deterministic Explainability Envelope and Dual Evidence Routes . . . 11
6.2 Deterministic Ranking, Delivery Modes, and Copyright Discipline . . . 11

7 Atomic Goal Visualizations **12**
7.1 Canonical Link Schema and MCP App Presentation Contract . . . 12
7.2 Production Workspace, Format, and Provider Policy . . . 12
7.3 CLI Workflow and Two-Stage Review Gates . . . 12

8 Learning-Goal Book and Evidence-Review Pipeline **14**
8.1 Shared BookModel, One-Goal-Per-Page Atlas, and Evidence Profiles V2 . . . 14
8.2 Multi-AI Review Loop, Evidence Gates, and Moderated Feedback Custody . . . 14

9 Canonical Gymnasium Rollout Policy **16**
9.1 Nine Core Decisions and Scope Vocabulary . . . 16
9.2 Subtree Migration State Machine and Adoption Sequence . . . 16

10 MEM/FWU Roundtrip Plan **18**
10.1 Semantic Roundtrip Probe and Required Coverage . . . 18
10.2 Core-First FWU Alignment and Application Profile . . . 18

11 Dual Curriculum Packages: JSON Runtime and Curriculum Ontology **19**
11.1 Seven Binding Architectural Decisions and Release Artifacts . . . 19
11.2 Generic Runtime Decoupling, Versioning, and Phases 0–6 . . . 19

1 General Goal System and Incremental Migration

SkillPilot's long-term goal model follows two governing principles: **"As much semantics as necessary, as little ontology as possible"** and **"Goals are stable. Structure is contextual."** The formal mathematical graph core is specified in Graph Definition; projection rules live in 3, and runtime fallback mechanics live in Minimal Schema & Runtime Fallback.

1.1 Four Semantic Layers Plus One Composition-View Layer

Rather than mixing subject competencies, school-year containers, and process taxonomies into one undifferentiated hierarchy, the target architecture separates four semantic layers plus one learner-facing projection layer:

| Layer | JSON / Artifact Carrier | Role and Invariants | | — | — | — | | **Goal layer** | `goals[]`, `contains`, `requires` | Durable assessable atomic goals and subject-matter clusters; owns mastery, frontier eligibility, and canonical IDs. | | **Program layer** | `programUnits[]` (`id`, `kind`, `label`, `order`, `parentUnitId`) | Contextual structural units such as year, semester, module, phase, track, or exam; never mastered directly. | | **Placement layer** | `goalPlacements[]` (`goalId`, `unitId`, `relation`, `context`) | Attaches unchanged canonical goals (`primary`, `secondary`, `assessed`) to program units under explicit scope contexts. | | **Competency-axis layer** | `competencyCatalog[]` (`id`, `label`, `dimension`), `competencyRefs[]` | Orthogonal capability taxonomies (for example `K1`–`K6` process competencies); never main content parents unless assessable. | | **Composition-view layer** | `*.view.json` (`structure`, `canonicalSubtree`, `goalEntry`) | Scope-specific learner-facing upper trees referencing reviewed canonical subtrees without duplicating goal payloads. | {cols="22%,32%,46%"}

Interpretation invariants:

- Year, semester, module, and phase containers are **not** the durable semantic backbone of the skill graph; `phase` is retained as compatibility/badge metadata rather than canonical parentage.
- Broad capability families (`K1`–`K6`) are taxonomy axes, not program units; only concrete, assessable process skills become actual `LearningGoal` nodes.
- Initial learner narrowing (jurisdiction, school form, stage, duration model, course profile) acts as **entry scope** over program units, placements, composition views, and compiled `applicability` (Levels of Personalization), never as duplicated canonical goals.

1.2 Recursive Learning Units and Incremental Migration

Recursive educational units (Graph Definition §11, Learning Units; §12, Optional Planning) compose over the same stable canonical goals:

1. Within one resolved course, each planned atomic acquisition has **one directly owning unit**; ancestor units aggregate child targets without re-owning them.
2. Competence mastered or recognized in a child unit is **globally effective** immediately, even while a broader parent integration exam remains open; local recognition applies only to directly owned ordinary targets inside units that satisfy the validity rules R1–R5 of 11 November 2026 (Graph Definition §13.3).
3. Passing a parent unit's integration exam never marks protected child exams or memorization decks as passed; explicit whole-unit progression (Unit Progression) is validated separately from atomic `requires` acyclicity.

[!NOTE] **Implementation boundary:** This chapter defines the target semantic split and incremental 6-phase migration path (additive optional schema fields → projection-first UI with `phase` fallback → pilot slice → subtree adoption → composition-view cutover → legacy retirement) without forcing a big-bang rewrite or claiming an already implemented `LearningUnit` runtime compiler.

2 Node Types

SkillPilot classifies learning goals into four pedagogical goal kinds that are orthogonal to the structural atomic/cluster distinction (Graph Definition) and to program-unit kinds (3).

2.1 Outcome Contract and Compiled Applicability

| Node Type | Pedagogical Purpose | Typical Content & Completion Basis | | — | — | — | | **Orientation** | Meaningful, low-pressure entry into a subject or phase | Concrete possibilities, personal interest follow-up, and explicit learner closure (`semanticKind: "orientation"`, rule `GVR-014`). | | **Understanding** | Conceptual or procedural subject mastery (default) | Assessable `"The learner can ..."` competencies governed by `requires`. | | **Memorize** | Durable recall of facts, vocabulary, and formulas | SM-2 spaced-repetition practice plus a mandatory coach-led **verified hard recall** check over all required cards. | | **Exam** | Realistic scored assessment simulation | Verbatim `examData` tasks graded strictly by `scoring.steps` under neutral AI Exam Mode supervision. | {cols="18%,34%,48%"}

Every atomic goal shares one binary outcome contract: **mastered or not mastered** (persisted as `1` / `1.0` for runtime compatibility; see Learning Workflow). Cluster progress is derived strictly from unique atomic descendants. Optional compiled `applicability` metadata (for example `{"jurisdiction": ["DE-HE", "DE-BY"]}`, with authoring exceptions in `extended Data.applicabilityOverrides`) filters visibility in scoped views without defining a node type or replacing semantic `tags`.

2.2 Orientation, Understanding, and Memorize Semantics

Orientation (`semanticKind: "orientation"`): Checked by `GVR-014` (Graph Validation Rules) to carry zero `requires` and no exam, memory, example, or assessment metadata. Orientation never tests prior knowledge or terminology. Completion follows a four-step dialogue: (1) present possibilities and invite an interest or question, (2) respond to a bare label with a tailored follow-up inviting a personal reaction, (3) summarize positively and offer closure, and (4) persist `1` **only after** the learner explicitly agrees to close (never exposing a graded `0/50/100` UI control). At genuine zero-progress entry into a scope, unresolved orientation goals form the sole selectable frontier; learners with existing progress are never reset to the orientation gate (see Graph Definition §8.2 and didactic routes in §3.3).

Understanding: The default node kind when no specialized payload is present; availability follows `requires` and optional `applicability`.

Memorize (SM-2 + Verified Recall): Each card maintains two process-data lanes that determine the single goal outcome:

- **SRS practice lane:** `interval` (days), `repetition`, `ef` (easiness factor, clamped to ≥ 1.3), and `nextReview` (ms). On review, `quality >= 3` increments `repetition`, sets `interval` to `1`, `6`, or `round(lastInterval * ef)`, updates `ef`, and schedules `now + interval * 24h`; `quality < 3` resets `repetition = 0` and `interval = 1` with unchanged `ef`. Due cards (`nextReview <= now` or unseen) are shuffled per session (up to 20 cards). Interactive MCP practice components translate **known** / **not known yet** into SM-2 updates; flipping cards never writes state, and completing an SRS practice queue **never** proves mastery.

- **Verified recall lane (`verifiedRecall`):** Tracks `status` (`passed` | `failed`), `attempts`, `failures`, `lastTestedAt`, `passedAt`, and `lastFailedAt`. Learners may trigger **Verify** at any time: the coach fetches only the prompt first, waits for an unassisted answer, then fetches the expected answer and records `passed` (updating SRS as a strong retrieval) or `failed` (re-queuing as due). Controlled recall of **all required cards** in a fully resolved, valid deck establishes permanent goal completion; subsequent SRS due dates never withdraw established goal mastery (the 29 October 2026 code inspection of `LearnerService.java` `isSrsMasteredByVerifiedRecall` and `useSrsMastery.ts` records current flag derivation without authorizing mastery withdrawal).

2.3 Exam Nodes, Recognition Review, and Scoring

Exam nodes embed `examData` (`taskContent`, optional `taskContentEn`, `solutionContent`, optional `solutionContentEn`, and `scoring: {maxPoints, passingPoints, steps: [{id, points, description}]}`). Learners may attempt exams directly without completing preparatory prerequisites first; failed attempts use `requires` for targeted remediation. Under the 4 November 2026 Assessment-Gated Completion concept (implementation adaptation pending), assessments that substantiate global competence recognition must undergo static authoring review proving that no candidate can pass via unrelated compensating points while omitting an essential recognized competence. Under the unit validity rules of 11 November 2026, an assessment claims coverage (`coveredGoalIds`) only for goals it requires directly or transitively (R3), every own ordinary goal of a unit lies on a prerequisite path from an own assessment of that unit (R1), and at least one such assessment carries reviewed evidence that its tasks and normal grading genuinely test the goal (R2); a missing path is a finding for subject review, never a reason to add a redundant edge (Graph Definition §13.3).

Scoring, persistence, and **Exam Mode** supervision follow strict rules:

- **Formulas:** total $= \min(\sum \text{stepPointsAwarded}, \text{scoring.maxPoints})$ and passed $= (\text{total} \geq \text{scoring.passingPoints})$ (logic errors yield `0` step points; arithmetic slips allow partial step points).
- **Binary persistence:** On `passed == true`, set mastery to `1.0` immediately after grading (no extra learner confirmation required; never claim persistence if the write fails). On `passed == false`, leave mastery unchanged and persist **no** failed attempt or score (unlimited retries remain available; revealing the solution afterward is a documented evidential trade-off).
- **AI Exam Supervisor workflow:** (1) Show a neutral 2–4 line exam header, deliver `examData.taskContent` **verbatim** without chunking or hints, and append one fixed submission line; (2) accept only readability clarifications until a single complete submission or explicit give-up arrives; (3) grade strictly against `scoring.steps`; (4) persist mastery on pass only; (5) report score, pass/fail, and step feedback, reveal `solutionContent`, switch back to learning-coach mode, and wait for an explicit learner choice before starting new content.

3 View Projection and Goal Placement

A careful reader must be able to reconstruct every visible learner-facing tree directly from documented JSON fields without inspecting runtime code or relying on title/`phase` heuristics.

3.1 Four View Families and Five Non-Negotiable Projection Rules

| View Family | Primary Question Answered | Exclusive Structural Source of Truth | | — | — | — | | **Content tree** | What belongs under what in subject-matter terms? | `goals` + explicit `contains` edges (ordered by `contains` array). | | **Program tree** | Where in the curriculum is a goal introduced, revisited, or assessed? | `programUnits` (`parentUnitId`, `order`) + `goalPlacements` (`primary` attaches; `secondary`/`assessed` render as badges/overlays, missing yields `unplacedInScope`). | | **Competency view** | Along which capability or process axis is a goal classified? | `competencyCatalog` + `competencyRefs` (orthogonal taxonomy projection). | | **Composition tree** | How does one resolved scope see the subject as one default tree? | Reviewed `*.view.json` (`structure` + `canonicalSubtree` / `goalEntry`) + canonical `contains` expansion. | {cols="20%,36%,44%"}

Five binding rules govern all projections:

1. **Rule 1 (No heuristic inference):** Never infer parent-child edges from titles, localized labels, `phase`, `dimensionTags.phase`, or string similarity.
2. **Rule 2 (Explicit edge provenance):** Every visible edge originates from exactly one documented field (`contains`, `parentUnitId`, `goalPlacements`, `competencyRefs`, or composition-view `children`).
3. **Rule 3 (Single relation family per view):** Enrich views with badges or filters, never by silently mixing parent hierarchies.
4. **Rule 4 (Reviewed composition view wins):** When a scope-specific `.view.json` exists, it defines the default learner-facing tree while `programUnits`/`goalPlacements` remain valid for program views and diagnostics (`ALL` is a query-only sentinel, never serialized in placements or applicability).
5. **Rule 5 (Single-occurrence tree per resolved scope):** Within one resolved scope (such as `DE-HE / Gymnasium / SekII / LK`), each visible goal appears **at most once** under **at most one** visible parent; pairwise overlapping `canonicalSubtree` expansions are compile-time errors.

3.2 Prerequisite Closure, Personal Learning Scope, and Canonical Identity

Under the Product Owner decisions of **9 November 2026**, **10 November 2026**, and **11 November 2026** (Graph Definition §§5.1–5.2):

- **Unpruned prerequisite closure (9 Nov 2026) & learner-read resilience (11 Nov 2026):** Selecting atomic goals must preserve their complete transitive prerequisite demand on the unpruned canonical graph (`A requires B requires C` retains `B` even if `A` and `C` alone matched initial scope filters). Unknown IDs, cycles, or unaccounted required atoms remain hard errors in validation, curriculum QA, and native acceptance; on learner reads, when a learning context's canonical graph is not strictly compilable, the runtime keeps the reviewed seed scope without the derived closure, claims no learning basis, offers no goal whose demand is broken, and logs the failure once per curriculum and error code (errors in the seed selection itself remain hard errors).

- **Personal learning-scope basis L (10 Nov 2026):** Necessary prerequisites belong to learning, progress, and completion, not merely hidden lookup support. Given valid reviewed atomic seed targets S, compile the fixed global target basis $L = S \cup \{p \in A \mid \exists t \in S : (t, p) \in R^{+}_{\text{at}}\}$ on the unpruned canonical graph. L depends solely on personal configuration and curriculum/projection revision, **never** on current mastery. Mastered atoms remain in L as already satisfied; open prerequisites receive navigable acquisition targets in the full Personal Curriculum with explainable `required by` provenance (without inventing new source mappings, enrolling unrelated siblings, or mutating mastery). Level 3 focus equals the selected roots' descendant union intersected with L (normal mode masks checks to focus *after* effective transitive reachability on canonical `requires`, without erasing paths through outside-focus intermediates).
- **Subject-local `projectionRole` specificity:** Authored view references carry `projectionRole: "target" | "prerequisiteOnly"` (default `"target"`). Specificity resolves overlaps via (1) direct `goalEntry` over `canonicalSubtree`, (2) deeper/nearest `canonicalSubtree` over broader ancestor subtrees, and (3) `"target"` on equal specificity. A goal marked `prerequisiteOnly` in one subject (for example Hessen Sek-II-LK Mathematics bridge/formula goals or Physics math support) still requires an acquisition target somewhere in the closed Personal Curriculum L if genuinely required.
- **Preserved canonical atomicity:** Collapsing or referencing a canonical cluster never creates an opaque substitute atom and never transfers cluster results to children (see also the historical `DPK-006c-c` boundary in Dual Package Implementation Status and tooling split in Curriculum Authoring Editors).

3.3 Learning Units in Projected Views

Projected Learning Units (target concept; runtime compiler pending) bind to the reviewed composition view or program projection:

1. **Single direct course owner:** Every planned atomic acquisition has exactly one directly owning unit in the resolved course; ancestors aggregate child obligations without duplicating ownership.
2. **Cross-subject identity:** Shared canonical goals (for example `target` in Mathematics and `prerequisiteOnly` in Physics) count once in the joint plan while retaining subject-local roles.
3. **Full prerequisite accounting:** Every required atom must be covered by the explicit entry basis or a planned acquisition no later than its dependent target.
4. **Local recognition & block ordering:** Implicit recognition evaluates only against each directly owning unit's reviewed contract; parent exams neither reopen completed child units nor pass protected child exams/cards, and whole-unit progression remains a separate explicit course policy.

4 Source and Resource Links

SkillPilot enforces a strict inversion principle for external learning media: **"Content references the curriculum; the curriculum does not select providers."**

4.1 Three-Way Purpose Split and Hard Canonical Prohibition

| Mechanism | Carrier | Permitted Purpose & Constraints | | — | — | — | | **Official curriculum provenance** | `sourceRef` and curriculum-typed `resourceLinks` (`type: "curriculum"`) | Trace canonical goals to official state/institutional syllabi, statutes, and page locators (6). | | **First-party goal visualizations** | `resourceLinks` with `type: "goal-visualization"` | Optional curated 16:9 concept illustrations owned and reviewed inside SkillPilot (7). | | **External learning materials** | Optional packages under `content/` (5) | Provider-specific textbooks, simulations, videos, and exercise collections referencing stable canonical `goalId`s from outside the core graph. | {cols="24%,30%,46%"}

Normative rules:

- **No new external provider links on canonical goals:** Canonical `SkillLandscape` JSON files must not add third-party textbook, video, or practice-platform links in `resourceLinks`. Earlier practice-link migration work remains historical provenance evidence only and is never a model for re-attaching external providers to canonical nodes.
- **Privacy-safe plain URLs & honest coach boundary:** Any surfaced link must be a plain, tracking-free URL without learner or session identifiers. AI coaches may mention offered links as optional reading, but must **never** claim to have fetched or read external web pages they did not inspect.

5 Content Integration: Architectural Framework

This chapter establishes the binding v1.1 architectural framework (Product Owner decision of **21 September 2026**, GitHub Issue `#50`; PoC details in Physik-Libre PoC, PoC Operations, and Issue #54 Content Packages).

5.1 Three-Layer Separation and Personal Selection

```
Canonical Skill Graph (provider-neutral goalIds)
        ^
        |  referenced from outside via content/ packages
Optional Content-Link Packages (textbooks, simulations, tasks)
        ^
        |  activated per learner via ordinary SkillPilot ID
Personal Material Selection (Cockpit & Coach, default opt-in/off)
```

Core architectural invariants:

- **Provider-neutral canonical core:** Canonical landscapes never hardcode or privilege external textbook, video, or publisher URLs; content packages under `content/` point inward to stable canonical `goalIds`.
- **Ordinary SkillPilot-ID personalization:** Learners select or deselect optional content packages directly in their ordinary Cockpit profile without special activation keys, teacher accounts, or third-party logins.
- **Operational kill-switch & privacy:** Every external content package can be disabled centrally via an operational kill-switch without touching canonical landscapes or learner mastery. Outbound links remain plain, parameter-free URLs without SkillPilot IDs or tracking tokens; SkillPilot relicenses no third-party content.

5.2 Physics Pilot and Four Acceptance Criteria

The initial reference pilot binds open-licensed physics textbook sections (`https://physikbuch.schule/`) after Physics `M7` stabilization and expands to Mathematics and LEIFIphysik under Issue `#54`. Every content-package integration is judged against four binding acceptance criteria:

| Criterion | Required Proof | | — | — | | **1. Unchanged core** | Disabling or removing all `content/` packages leaves canonical graphs, views, frontier logic, and mastery 100% intact. | | **2. Personal selection** | Learners control package visibility through their ordinary SkillPilot profile without special keys or accounts. | | **3. Concrete utility** | Active links resolve to goal-specific reading or practice sections and appear cleanly in Cockpit and Coach contexts without fabricated page summaries. | | **4. General viability** | Adding a second provider or subject requires only a declarative `content/` package and validation pass, zero core schema changes. | {cols="26%,74%"}

6 Human-Readable Source Rationales

Every canonical learning goal in a SkillPilot skill landscape should be explainable back to official curricular evidence without requiring teachers or reviewers to inspect raw JSON ledgers or SPARQL outputs (operational commands live in Source Rationales Runbook).

6.1 Deterministic Explainability Envelope and Dual Evidence Routes

`generateGoalSourceRationales.ts` compiles a deterministic `schemaVersion: 1` JSON envelope per goal combining two complementary evidence routes:

- **Status classification (`sourceRationaleStatus`):** `classic_source_reviewed` (backed by reviewed official source extraction and mapping), `partial` (incomplete or broader topic-level coverage), or `gap` (missing source proof, surfaced honestly under `limitations`).
- **Classic source route (`classicSourceRoute`):** Binds reviewed `source-to-canonical` mappings (`exact` or `partial`), official syllabus documents, section breadcrumbs, and short statutory locator excerpts.
- **MEM SPARQL route (`memSparqlRoute`):** Cross-references external Metadata for Educational Media (`MEM`) / FWU ontology alignments (MEM SPARQL Audit) where available.

6.2 Deterministic Ranking, Delivery Modes, and Copyright Discipline

Candidate source records are ranked deterministically:

1. `exact` mappings precede `partial` mappings;
2. active/current state syllabi precede retired legacy snapshots;
3. sources matching the learner's selected jurisdiction precede nationwide or cross-state backup evidence.

In standalone curriculum package mode (11), the runtime serves generation-bound evidence via Catalog 1.2 discovery rather than hardcoded repository paths, failing closed without falling back to stale static files. Official syllabus quotes remain strictly bounded short citation locators (never full-document reproductions) to respect statutory/publisher copyright boundaries.

7 Atomic Goal Visualizations

Optional concept illustrations help learners grasp the core idea of an atomic learning goal visually without replacing independent reasoning or task solving.

7.1 Canonical Link Schema and MCP App Presentation Contract

Reviewed visualizations attach to canonical atomic goals via `resourceLinks`:

```
{
  "type": "goal-visualization",
  "resourceType": "image",
  "role": "primary",
  "skillpilotId": "<exact-canonical-goal-id>",
  "url": "/assets/goal-visualizations/<subject>/<id>/<id>.png"
}
```

In MCP-capable hosts such as OpenAI ChatGPT, `render_skillpilot_goal_visualization` serves a surface-neutral `text/html;profile=mcp-app` widget governed by a one-shot `(goalId, stateVersion)` presentation authorization and the learner toggle `showGoalVisualizations InChat`. The widget remains collapsed until the image loads cleanly, accepts bounded MCP result envelopes while using persisted widget state only as a bootstrap fallback, and retains all previously advertised content-addressed UI resource hashes unchanged.

7.2 Production Workspace, Format, and Provider Policy

Authoring prompts, candidate generation, and QA ledgers reside under `curricula/DE/Gymnasium/visualizations/` (`prompt.de.md`, `image-reconstruction-prompt.de.md`, published via `npm run deploy:assets` at 16:9 `1600x900` PNG and inspected at both `360px` mobile and `680px` desktop widths, calibrated on pilot goal `502ecaa7-cca6-5c51-a1cc-da09a7b2382c`).

Under the binding **Product Owner decision of 13 September 2026**:

- **Default `KEEP`:** Retain good existing Mathematics and Physics images regardless of which provider generated them; never regenerate retro-actively merely to unify providers.
- **Provider priority for new or defective images:** Prefer **ChatGPT/Codex image generation** in SkillPilot's friendly, abstract, clear, comic-like raster style when subject-matter accuracy and representation fit. **Nano Banana Pro** (`gemini-3-pro-image`) is an allowed alternative whenever it better serves the didactic representation — no prior failed attempt with the other provider is required.
- **Narrow SVG/programmatic exception:** Programmatically plotted diagrams or SVGs remain narrow, explicitly documented exceptions justified by exact geometric/function-plot precision, never a blanket style replacement.

7.3 CLI Workflow and Two-Stage Review Gates

Helper scripts in `app/` (`visualization:generate:nano-banana`, `visualization:plan-batch`, `visualization:check-prompt-appends`, `visualization:prepare`, `visualization:import`) separate asset generation from approval:

1. **AI QA ledger (`aiApprovedAssetSha256` / Gate V):** Records structured review labels (`accepted_pilot`, `accepted_pilot_after_regeneration`, `rejected_regenerate`, `deferred_provider_limitation`, `needs_external_review`) bound to exact file SHA-256 digests.
2. **Human publication gate (`humanApproved`, `HR-003`):** Public Standalone releases and `public`-mode GoalBooks (8, Human Review Gates) require explicit human content and rights clearance (`CC-BY-4.0` vs. provenance); generation or AI review alone never grants publication approval.

8 Learning-Goal Book and Evidence-Review Pipeline

The learning-goal book (`Lernzielbuch`) and positive understanding-evidence pipeline make every curricular-atomic goal inspectable by humans and multi-AI reviewers without bloating learner-facing goal descriptions (teacher timeline planning on top of the GoalBook is specified in Goal-Book Time Planning).

8.1 Shared BookModel, One-Goal-Per-Page Atlas, and Evidence Profiles V2

One deterministic `BookModel` (`contracts/goal-book/v1/goal-book-model-1.1.schema.json`) drives both the read-only `/lernzielbuch` WebGUI and the A4 portrait PDF:

- **Current nationwide review editions (30 Aug 2026): 792** unique curricular-atomic pages for Gymnasium Mathematics and **461** pages for Physics across all 16 German states (`publicationMode: review`, carrying QA-ledger candidate images without claiming human image approval; coach runtime and Mastery writes remain unchanged).
- **Fourteen binding invariants:** Concise canonical `title`/`description`; separate fingerprint-bound V2 evidence profiles (`contracts/goal-evidence/v2/goal-evidence-profile.schema.json` capturing `expectations`, `coverageExpectations`, `variationAxes`, and `applicationCaseBriefs`; V1 retained as archived audit history); **one physical page per ordinary curricular-atomic target goal** displaying full **Lernziel-ID**, chapter path (`canonical-composition-view-v1` + **Bundeslandspezifische Ergänzungen** with gap-free `treeOrder`), stable Kahn topological order from `requires`, clickable named PDF destinations (`goal-<uuid>`), fail-closed overflow/legibility checks, incremental Skia PDF 1.4 bookmark outlines, state-primary applicability tuples (`Bundesland -> [{stage, durationModel, courseProfile}]` — never Cartesian `G8`/`G9` cross-products), and on-demand `/lernzielbuch/<bookId>.original-sources.json` supplements.
- **Application-based recognition review (4 Nov 2026):** Assessments supporting global competence recognition (Assessment-Gated Completion) require static review against compensating-point bypasses before release.

8.2 Multi-AI Review Loop, Evidence Gates, and Moderated Feedback Custody

Quality iteration combines a fast blind multi-AI loop with strict generalization gates and a slow, privacy-minimized public feedback channel:

| Scale | Levels & Meaning | | — | — | | **Evidence (`E0`–`E5` + `NORMATIVE`)** | `E0` unanchored opinion → `E1` anchored observation → `E2` reproducible minimal counterexample → `E3` independent multi-model/prompt triangulation → `E4` controlled field evidence → `E5` replicated cohort evidence. | | **Claim scope (`G0`–`G4`)** | `G0` single case → `G1` one goal → `G2` goal archetype (requires ≥ `E3` across ≥ 2 chapters) → `G3` subject/stage (stratified sample + replication) → `G4` global Coach/Mastery rule (Math + Physics + field canary). | {cols="30%,70%"}

- **Two-round V2 campaign:** Calibrated on 16 Mathematics goals (`5`–`10`, `E`–`Q4`, pilot goal `8dd9f210-2683-5902-acab-e3be22725232`) before the 792-goal batch: (1) freeze Atlas V1, (2) blind Codex V1 review (`keep` | `revise` | `split_review` | `block`), (3) deterministic check + goal-by-goal human adjudication, (4) controlled bilingual V2 authoring + downstream lane re-runs, (5) blind external-provider V2 review, (6) dissent-preserving synthesis. Neither AI agreement nor silence ever auto-approves canonical changes.

- **Two-phase `/lernziel-feedback` custody transfer:** Default-off in production behind `SKILLPILOT_GOAL_FEEDBACK_OPERATOR_TOKEN`; accepts privacy-minimized V2 envelopes without learner/SkillPilot IDs, contact fields, or attachments. `POST /api/operations/goal-feedback/v1/export-batches` exports deterministic batches + server-derived publication snapshots into owner-only `tmp/goal-feedback/inbox/` (`trusted-context.jsonl` separated from `untrusted-feedback.jsonl`); only after local byte/digest verification does idempotent `DELETE .../{exportId}` with `If-Match` clear live rows while retaining a content-free batch receipt.

```
npm --prefix app run quality:goal-evidence:check
npm --prefix app run build:goal-book-model -- \
  scripts/config/goal-books/de-gym-math-national-atlas.json
npm --prefix app run build:goal-book-original-sources
npm --prefix app run check:goal-book-original-sources
```

9 Canonical Gymnasium Rollout Policy

This chapter specifies the durable target architecture and rollout rules for converging German Gymnasium curricula across all 16 Bundesländer into one shared canonical skill graph per subject without duplicating competencies per state (execution plans and migration ledgers live in Canonical Gymnasium Implementation Plan and Migration Status).

9.1 Nine Core Decisions and Scope Vocabulary

1. **Canonical subject layer first:** One continuous `SkillLandscape` per subject under `curricula/DE/Gymnasium/canonical/` spanning Sekundarstufe I through Abitur readiness.
2. **Legacy continuity during transition:** Retired state trees survive as state-scoped audit/source inputs under `curricula/DE/Gymnasium/input/<STATE>/` (enforced by retirement validators in Graph Validation Rules) while compatibility mappings bridge learner progress.
3. **Minimal mapping layer:** Explicit `legacyGoalId -> canonicalGoalId` mapping files (`matchType: "exact" | "partial"`) instead of a second content universe.
4. **ISO 3166-2 state identifiers:** Metadata and filters standardize on `DE-HE`, `DE-BY`, ..., `DE-TH` (`ALL` is a query-only sentinel, never serialized).
5. **Stable coach/MCP APIs:** Backend/runtime code hides legacy-vs-canonical translation so AI coaches always see one learner state, one frontier, and one mastery model.
6. **Controlled cross-subject `requires`:** Canonical subject landscapes may author cross-subject edges directly (for example Physics goals requiring Mathematics atoms by their stable IDs) without cloning foreign subtrees; graph validators resolve these references across loaded landscapes, and `CQR-001` includes required foreign goals and their inheriting ancestors in its atomic-normal-form and transitive-minimality audit.
7. **Shared `Gymnasium (DE)` root & reviewed scope vocabulary:** One DE-level composition root exposes child subject landscapes governed by `jurisdiction` (`DE-*`), `schoolForm` (`Gymnasium`), `stage` (`SekI`, `SekII`), `durationModel` (`G8`, `G9`), and `courseProfile` (`GK`, `LK`, `GK+LK` union without duplicated IDs), backed by explicit state-scoped `.view.json` files (3).
8. **G9 Sek-I reference grid:** Canonical Sek-I authoring normalizes onto the shared `Jahrgang 5`–`Jahrgang 10` grid while preserving `G8`/`G9` source truth in provenance and view bindings.
9. **State-scoped retained inputs:** Non-canonical source extracts, Abitur blueprints (`curricula/DE/Gymnasium/input/HE/abi/`), and audit records remain strictly state-scoped under `input/<STATE>/`.

9.2 Subtree Migration State Machine and Adoption Sequence

The unit of convergence is a **didactically closed subtree** (never a forced big-bang rewrite), progressing monotonically through four states:

| State | Meaning & Learner-State Contract | | — | — | | `1. legacy_frozen` | Existing state curriculum JSON is read-only source input; runtime/mapping fixes allowed without rewriting legacy trees. | | `2. subtree_adopted` | Closed `contains` subtree reconstructed in `canonical/` with explicit `exact`/`partial` mappings and bound prerequisites. | | `3. cutover_ready` | Learner-state continuity, frontier coherence, and mapping coverage verified so canonical navigation can become default. | | `4. legacy_view_retained` | Canonical subtree is the primary learner-facing route; compatibility views remain temporarily until formal retirement. | {cols="26%,74%"}

Strategic adoption sequence: preserve Hessen learner continuity → establish Mathematics across Sek I and Sek II as the primary prerequisite spine → fold Bavaria and remaining states

into canonical Mathematics via mappings and `.view.json` projections → widen multi-subject canonical navigation and cross-subject prerequisites.

10 MEM/FWU Roundtrip Plan

SkillPilot treats Metadata for Educational Media (`MEM`) and `FWU-DE/lehrplan-ontologie` as an adjacent semantic metadata and provenance layer rather than a competing model, acting as a practical open-source test bench for traceable AI learning-step navigation (see the operational pipeline in MEM/FWU Roundtrip Pipeline and reference paper <https://aifyer.com/ki-lernbegleitung/versions/v1.0/schulisch-verantwortete-ki-lernbegleitung-v1.0.pdf>).

10.1 Semantic Roundtrip Probe and Required Coverage

```
SkillPilot skill landscape (JSON export ZIP)
  -> MEM/FWU-first RDF/OWL bundle + hashed binary sidecars
  -> Reconstructed SkillPilot skill-landscape data (no source JSON access)
  -> Deterministic SkillPilot graph, view, and asset validation
```

Without embedding opaque JSON blobs or Base64 media literals, the semantic bundle preserves:

- Goal identity (`id`, `shortKey`, `title`, `description`), atomic/cluster classification, `weight`, `core`, and dimension metadata;
- Direct `contains` composition and didactic `requires` prerequisites;
- Program structure (`programUnits`), `goalPlacements`, competency axes (`competencyCatalog`, `competencyRefs`), and learner-facing composition views;
- Official source/provenance references plus `goal-visualization` references (`schema:ImageObject`) paired with byte-identical SHA-256 image sidecars (learner mastery state stays strictly outside public curriculum bundles).

10.2 Core-First FWU Alignment and Application Profile

| Relation / Concept | Core-First RDF/OWL Representation & Constraints | | — | — | | **Didactic `requires`** | Specialized reified `LP_0000554` resource attached via `LP_0030071` (`hat Verweis`) and pointing to the prerequisite via `LP_0030072` (`verweist auf`, introduced in `FWU-DE/lehrplan-ontologie#9`). Generic `CE-Verweis` links must **never** be misread as prerequisites. | | **Curricular `contains`** | `BFO_0000051` (`hat Teil`) only for genuine curricular part-whole relations, paired with `skillpilot:containsGoal` on every authored direct edge so transitive BFO reasoner inferences never obscure direct edges (mixed practice/exam/memory/navigation edges use only `skillpilot:containsGoal`). | | **Minimal application terms** | `skillpilot:{LearningGoal,AtomicGoal,ClusterGoal,containsGoal,weight,core,shortKey,ProgramUnit,GoalPlacement,CompositionView,GoalVisualizationReference}` plus `schema:ImageObject` for binary sidecars. | {cols="28%,72%"}

Success requires full SkillPilot validator conformance (acyclic `requires`/`contains`, referential integrity, valid composition views), deterministic field equivalence, OWL 2 DL / structural core validation, and byte-identical image SHA-256 verification without hidden access to the original JSON input.

11 Dual Curriculum Packages: JSON Runtime and Curriculum Ontology

Every versioned curriculum release publishes **one exact subject-matter state** in two semantically equivalent representations — without dual authoring and without requiring a SkillPilot deployment to read the `curricula/` source tree (current checkpoint: Dual Package Implementation Status; operational contracts: Package Readiness, Release-Model Conformance, Dual Release Contracts, and Package Provisioning).

11.1 Seven Binding Architectural Decisions and Release Artifacts

```
Authoring sources -> Release compiler -> Normalized release model
  -> JSON runtime ZIP (<id>-<ver>.json.zip)
       -> Core-first ontology exporter -> FWU/OWL ZIP (<id>-<ver>.fwu-owl.zip)
            -> Isolated reverse compiler (bubblewrap, no source JSON)
                 -> Reconstructed JSON model -> Semantic equivalence gate
```

1. **No dual authoring:** Both ZIPs derive from one source state; fixes happen only in authoring sources.
2. **JSON ZIP is the normative runtime contract (`full-standalone-v1`):** Contains `metadata/manifest.json` (explicit artifact roles, offline `schemas/catalog.json`), `data/runtime/{catalog.json,dependency-closure.json,migration-aliases.json}`, compiled `applicability`, `canonical/`, `views/`, `cards/`, `resources/`, `mappings/`, `sources/`, `metadata/quality/`, `assets/`, and `licenses/`, operable with zero `curricula/` or `app/public` fallbacks and strict ZIP32 bounds (`3.5 GB` archive, `60,000` entries, `64 MiB` JSON/image, `128` JSON depth, `100:1`/`25:1` compression ratios).
3. **Ontology ZIP derives strictly from the finished JSON ZIP:** Contains `rdf/{declarations,runtime,landscape,views,mappings,sources,cards,assets,bundle}.nt`, `skillpilot-curriculum-profile.ttl`, `catalog-v001.xml`, pinned `ontology/{lehrplan-core.owl,shapes.ttl}`, and identical binary sidecars (validated for RDF syntax, SHACL, OWL 2 DL via ROBOT, and HermiT consistency).
4. **Semantic equivalence & shared `contentDigest`:** Governed by a closed **454-rule field-semantics registry** (`323` runtime + `130` publication normal forms + `1` legacy entry; `scalar`, `ordered-list`, `set`, `map`, `binary-reference`, `generated-non-semantic`; mapped via `fwu-core`, `skillpilot-profile`, bounded `registered-canonical-json-literal`, or `excluded-generated`). Unregistered fields fail the build; placement identities use `goalId@unitId@index`.
5. **Core-first, not Core-only:** Declarative subject profiles map required `semanticKind` values (`curricularAtomic`, `curricularArea`, `programStructure`, `practiceAssessment`, `memory`, `orientation`, `runtimeSupport`) without title guessing.
6. **Zero hidden JSON carrier:** Isolated netless `bubblewrap --clearenv` reverse compilation (`DPK-008d`) proves lossless reconstruction from RDF + sidecars alone.
7. **External release index & signing projection:** `<id>-<ver>.release.json`, `.equivalence.json`, `.provenance.json`, `.SHA256SUMS`, and `.release.json.sig` (stripping `authentication.releaseIndexSignature` before canonical hash binding) plus a signed release catalog (`staging` → `stable`, never unpinned `latest`).

11.2 Generic Runtime Decoupling, Versioning, and Phases 0–6

Four generic components decouple SkillPilot from repository paths: `CurriculumPackageRepository` (read-only lock/CAS reader), `JsonCurriculumPackageLoader` (schema/hash-verified domain builder), `CurriculumResourceResolver` (versioned `(packageId, deckId,`

`locale)` and `(packageId, resourceId)` hrefs), and `CurriculumActivationService` (quarantine extraction, dry-run validation, atomic CAS lock switch, rollback). Version axes stay orthogonal (`packageVersion`, `curriculumEdition`, `packageFormatVersion`, `runtimeContractVersion`, `supportedSkillpilotSoftware` currently `>=0.1.0 <1.0.0` for consumer `0.1.0`, `ontologyProfileVersion`, and pinned FWU Core commit/SHA), while `migration-aliases.json` distinguishes `renamed` (auto-transfers mastery) from `replacedBy`, `splitInto`, `mergedInto`, and `removed`. Licensing separates `licenseExpression`, `provenanceClass`, and `redistributionStatus` (`review-required` keeps staging candidates `not-ready-incomplete`).

| Phase | Scope & Recorded Milestone Status | | — | — | | **Phase 0** | **Technically complete (`DPK-004 / DPK-007a`):** Frozen Mathematics `0.1.0-conformance.3` (`914` ZIP entries, `757` images, 454-rule registry; intentionally `not-ready-incomplete` pending human image/rights/source-text gates). | | **Phase 1** | **Technically complete (`DPK-005b`–`DPK-007`):** Hermetic `bubblewrap` + Corretto + Playwright consumer verified (`1,079` goals, `33,382` source mappings, `88` views, `12` decks, `757` images, zero repository-poison reads under `strace`). | | **Phase 2** | **Technically complete (`DPK-008a`–`DPK-008d`):** Reproducible FWU-OWL bundle (`824,452` triples, SHACL/ROBOT/HermiT green) and isolated byte-identical JSON ZIP reconstruction (`911` entries, `111` logical artifacts). | | **Phase 3** | **Next planned (`DPK-009, not_started`):** Unified downstream equivalence oracle, dual hermetic consumer smoke tests, isolated `JSON -> FWU` pair reproducibility, and signed external `release.json`. | | **Phases 4–6** | Multi-subject generalization (Math → Physics embedded closure → languages → Chemistry) → multi-package production catalog/upgrades → repository split between curriculum tooling and SkillPilot software. | {cols="18%,82%"}