Skill-Graph-Architektur, Medien, Lernzielbuch und Paketaustausch

Schichtenmodell, Zieltypen, Sichtenprojektion, Quellen- und Contentanbindung,

Visualisierungen, Lernzielbuch-Pipeline, Gymnasium-Rollout und Dual-Release-Pakete

Geltungsbereich dieses Bandes

Ergänzend zur formalen Graph-Definition (Band I) bündelt Band Ia die weiterführenden Architektur-, Medien-, Publikations- und Austauschverträge rund um den Skill-Graphen: das allgemeine Zielsystem und seine schrittweise Migration, die pädagogischen Knotentypen, die deterministische Sichtenprojektion und Goal-Placement-Semantik, Quellen- und Contentlink-Trennung (Issue #50), nachvollziehbare Quellenbegründungen, atomare Zielvisualisierungen, die Lernzielbuch- und Evidenz-Review-Pipeline, die kanonische Gymnasium-Rollout-Politik sowie den MEM/FWU-Ontologie-Roundtrip und duale Curriculum-Paket-Releases.

Stand: 11. November 2026 · Lizenz: Apache-2.0 (technische Architektur- und Spezifikationsdokumentation; siehe LICENSING.md).

Eigene Skill-Landschaften und didaktische Medien stehen separat unter CC-BY-4.0; externe Inhalte behalten ihre Originallizenzen.

Inhaltsverzeichnis

1 Allgemeines Zielsystem und schrittweise Migration 3
1.1 Vier semantische Schichten plus eine Composition-View-Schicht . . . 3
1.2 Rekursive Lerneinheiten und schrittweise Migration . . . 3

2 Knotentypen 5
2.1 Ergebnisvertrag und kompilierte Applicability . . . 5
2.2 Semantik von Orientation, Understanding und Memorize . . . 5
2.3 Prüfungsziele, Anerkennungsreview und Bewertung . . . 6

3 Sichtenprojektion und Goal-Placement 7
3.1 Vier Sichtenfamilien und fünf unverhandelbare Projektionsregeln . . . 7
3.2 Voraussetzungshülle, persönlicher Lernscope und kanonische Identität . . . 7
3.3 Lerneinheiten in projizierten Sichten . . . 8

4 Quellen- und Ressourcenlinks 9
4.1 Dreiteilung der Linkzwecke und hartes Kernverbot . . . 9

5 Contentanbindung: Architekturrahmen 10
5.1 Dreischichtige Entkopplung und persönliche Auswahl . . . 10
5.2 Physik-Pilot und vier Abnahmekriterien . . . 10

6 Nachvollziehbare Quellenbegründungen 11
6.1 Deterministischer Erklär-Umschlag und duale Nachweisrouten . . . 11
6.2 Deterministische Rangfolge, Ausspielmodi und Urheberrechtsdisziplin . . . 11

7 Visualisierungen für atomare Lernziele 12
7.1 Kanonisches Linkschema und MCP-App-Darstellungsvertrag . . . 12
7.2 Produktionsbereich, Format und Provider-Priorität . . . 12
7.3 CLI-Workflow und zweistufige Review-Gates . . . 12

8 Lernzielbuch und Evidenz-Review-Pipeline 14
8.1 Gemeinsames BookModel, Ein-Ziel-pro-Seite-Atlas und Evidenzprofile V2 . . . 14
8.2 Multi-KI-Review-Schleife, Evidenzgates und moderierter Feedback-Transfer . . . 14

9 Rollout-Politik für das kanonische Gymnasium 16
9.1 Neun Kernentscheidungen und Scope-Vokabular . . . 16
9.2 Teilbaum-Zustandsmodell und Einführungsreihenfolge . . . 16

10 MEM/FWU-Roundtrip-Plan 18
10.1 Semantischer Roundtrip-Test und geforderte Abdeckung . . . 18
10.2 Core-First-FWU-Abbildung und Anwendungsprofil . . . 18

11 Duale Curriculum-Pakete: JSON-Runtime und Lehrplan-Ontologie 19
11.1 Sieben verbindliche Architekturentscheidungen und Release-Artefakte . . . 19
11.2 Generische Runtime-Entkopplung, Versionierung und Phasen 0–6 . . . 19

1 Allgemeines Zielsystem und schrittweise Migration

Das langfristige Zielmodell von SkillPilot folgt zwei Leitprinzipien: **„So viel Semantik wie nötig, so wenig Ontologie wie möglich“** und **„Ziele sind stabil. Struktur ist kontextabhängig.“** Der formale mathematische Graphkern ist in Graph-Definition spezifiziert; Projektionsregeln stehen in 3, Laufzeit-Fallbacks in Minimales Schema & Laufzeit-Fallback.

1.1 Vier semantische Schichten plus eine Composition-View-Schicht

Statt Fachkompetenzen, Jahrgangscontainer und Prozesstaxonomien in einer undifferenzierten Hierarchie zu vermischen, trennt die Zielarchitektur vier semantische Schichten sowie eine lernendenseitige Projektionsschicht:

| Schicht | JSON- / Artefaktträger | Rolle und Invarianten | | — | — | — | | **Zielschicht (Goal Layer)** | `goals[]`, `contains`, `requires` | Stabile prüfbare atomare Lernziele und fachliche Cluster; trägt Mastery, Frontier-Berechtigung und kanonische IDs. | | **Programmschicht (Program Layer)** | `programUnits[]` (`id`, `kind`, `label`, `order`, `parentUnitId`) | Kontextuelle Struktureinheiten wie Jahrgang, Halbjahr, Modul, Phase, Zweig oder Prüfung; nie direkt beherrschbar. | | **Verortungsschicht (Placement Layer)** | `goalPlacements[]` (`goalId`, `unitId`, `relation`, `context`) | Hängt unveränderte kanonische Ziele (`primary`, `secondary`, `assessed`) unter expliziten Scope-Kontexten in Programmeinheiten ein. | | **Kompetenzachsenschicht** | `competencyCatalog[]` (`id`, `label`, `dimension`), `competencyRefs[]` | Orthogonale Kompetenztaxonomien (z. B. Prozesskompetenzen `K1`–`K6`); keine primären Inhaltseltern, sofern nicht eigenständig prüfbar. | | **Composition-View-Schicht** | `*.view.json` (`structure`, `canonicalSubtree`, `goalEntry`) | Scope-spezifische obere Lernbäume, die geprüfte kanonische Teilbäume ohne Duplizierung von Zielinhalten referenzieren. | {cols=”22%,32%,46%”}

Interpretationsinvarianten:

- Jahrgangs-, Halbjahres-, Modul- und Phasencontainer bilden **nicht** das dauerhafte semantische Rückgrat des Skill-Graphen; `phase` bleibt als Kompatibilitäts- und Badge-Metadatum erhalten, nicht als kanonische Elternkante.
- Breite Kompetenzfamilien (`K1`–`K6`) sind Taxonomieachsen und keine Programmeinheiten; nur konkrete, prüfbare Prozessfertigkeiten werden echte `LearningGoal`-Knoten.
- Die anfängliche Eingrenzung durch Lernende (Bundesland, Schulform, Stufe, Dauer, Kursprofil) wirkt als **Einstiegsscope** über Programmeinheiten, Placements, Composition Views und kompilierte `applicability` (Ebenen der Personalisierung), niemals über duplizierte kanonische Ziele.

1.2 Rekursive Lerneinheiten und schrittweise Migration

Rekursive Lerneinheiten (Graph-Definition §11, Lerneinheiten; §12, Optionale Planung) setzen auf denselben stabilen kanonischen Zielen auf:

1. Innerhalb eines aufgelösten Kurses besitzt jeder geplante atomare Kompetenzerwerb **genau eine direkt besitzende Einheit**; übergeordnete Einheiten aggregieren Kindziele, ohne deren Erwerb erneut zu besitzen.
2. In einer Kindeinheit beherrschte oder anerkannte Kompetenz gilt sofort **global**, auch solange eine übergeordnete Integrationsprüfung noch offen ist; lokale Anerkennung gilt ausschließlich für direkt besessene gewöhnliche Ziele in Einheiten, die die Gültigkeitsregeln R1–R5 vom 11. November 2026 erfüllen (Graph-Definition §13.3).

3. Das Bestehen einer übergeordneten Integrationsprüfung besteht niemals geschützte Kindprüfungen oder Lernkartendecks mit; eine explizite Blockfolge ganzer Einheiten (Einheitsfolge) wird getrennt von der atomaren `requires`-Azyklizität geprüft.

> [!NOTE] **Implementierungsgrenze:** Dieses Kapitel definiert die fachliche Schichtentrennung und den sechsstufigen inkrementellen Migrationspfad (additive optionale Schemafelder → Projektions-UI mit `phase`-Fallback → Pilotschnitt → Teilbaumübernahme → Composition-View-Cutover → Altsystem-Stilllegung), ohne einen Big-Bang-Umbau zu erzwingen oder einen bereits implementierten `LearningUnit`-Laufzeitcompiler zu behaupten.

2 Knotentypen

SkillPilot gliedert Lernziele in vier pädagogische Zielarten, die orthogonal zur strukturellen Atomar-/Cluster-Unterscheidung (Graph-Definition) und zu Programmeinheitstypen (3) stehen.

2.1 Ergebnisvertrag und kompilierte Applicability

| Knotentyp | Pädagogischer Zweck | Typischer Inhalt & Abschlusskriterium | | — | — | — | | **Orientation** | Sinnstiftender, druckfreier Einstieg in Fach oder Phase | Konkrete Möglichkeiten, interessengeleitete Nachfrage und expliziter Abschlusswunsch (`semanticKind: örientation"`, Regel `GVR-014`). | | **Understanding** | Konzeptuelles oder prozedurales Fachverständnis (Standard) | Prüfbare „Die lernende Person kann …“-Kompetenzen, gesteuert über `requires`. | | **Memorize** | Dauerhaftes Abrufwissen (Fakten, Vokabeln, Formeln) | SM-2-Spaced-Repetition-Training plus verpflichtender Coach-geführter **harter Abruftest** über alle Pflichtkarten. | | **Exam** | Realistische Prüfungsaufgabe mit Bepunktung | Unveränderte `examData`-Aufgaben, streng nach `scoring.steps` im neutralen KI-Prüfungsmodus bewertet. | {cols=”18%,34%,48%”}

Jedes atomare Ziel besitzt denselben binären Ergebnisvertrag: **beherrscht oder nicht beherrscht** (zur Laufzeitkompatibilität als `1` / `1.0` gespeichert; siehe Operativer Lernablauf). Cluster-Fortschritt leitet sich ausschließlich aus eindeutigen atomaren Nachfahren ab. Optionale kompilierte `applicability`-Metadaten (z. B. `{"jurisdiction: ["DE-HE", "DE-BY"]}`, mit redaktionellen Ausnahmen unter `extendedData.applicabilityOverrides`) steuern die Sichtbarkeit in gefilterten Sichten, ohne einen Knotentyp zu bilden oder fachliche `tags` zu ersetzen.

2.2 Semantik von Orientation, Understanding und Memorize

Orientation (`semanticKind: örientation"`): Durch `GVR-014` (Graph- und View-Validierung) auf null `requires` sowie Freiheit von Prüfungs-, Karten-, Beispiel- und Bewertungsdaten geprüft. Orientierungsanker prüfen kein Vorwissen und keine Fachbegriffe. Der Abschluss folgt einem vierstufigen Dialog: (1) Möglichkeiten aufzeigen und Interesse oder Frage einladen, (2) bei bloßer Stichwortnennung gezielt nachfassen und eine persönliche Reaktion einladen, (3) positiv zusammenfassen und den Abschluss separat anbieten, (4) `1` **erst nach** ausdrücklicher Zustimmung speichern (ohne abgestuften `0/50/100`-UI-Regler). Beim echten Null-Fortschritts-Einstieg bilden offene Orientierungsziele die einzige wählbare Frontier; Lernende mit vorhandenem Fortschritt werden nie dorthin zurückgesetzt (siehe Graph-Definition §8.2 und didaktische Routen in §3.3).

Understanding: Standardknotenart ohne Sonderfelder; Verfügbarkeit folgt `requires` und optionaler `applicability`.

Memorize (SM-2 + Verified Recall): Jede Karte führt zwei Prozessspuren zur Bestimmung des einen Zielergebnisses:

- **SRS-Übungsspur:** `interval` (Tage), `repetition`, `ef` (Easiness-Faktor, begrenzt auf ≥ 1.3) und `nextReview` (ms). Bei `quality >= 3` steigt `repetition`, `interval` wird `1`, `6` bzw. `round(lastInterval * ef)`, `ef` wird aktualisiert und `nextReview = now + interval * 24h` gesetzt; bei `quality < 3` gilt `repetition = 0`, `interval = 1` bei unverändertem `ef`. Fällige Karten (`nextReview <= now` oder neu) werden pro Sitzung gemischt (bis zu 20 Karten). Interaktive MCP-Kartenkomponenten übersetzen **gewusst** / **noch nicht gewusst** in SM-2-Updates; bloßes Umdrehen schreibt keinen Zustand, und das Leeren der Übungswarteschlange beweist **niemals** Mastery.

- **Verifizierte Abrufspur (`verifiedRecall`):** Speichert `status` (`passed` | `failed`), `attempts`, `failures`, `lastTestedAt`, `passedAt` und `lastFailedAt`. Lernende können **Prüfen (Verify)** jederzeit starten: Der Coach lädt zuerst nur die Vorderseite, wartet auf eine eigenständige Antwort ohne Hilfe, lädt erst danach die Musterlösung und speichert `passed` (inkl. starkem SRS-Abruf) oder `failed` (erneut fällig). Der kontrollierte Abruf **aller Pflichtkarten** eines vollständig aufgelösten Decks begründet den dauerhaften Zielabschluss; spätere Fälligkeiten entziehen eine erreichte Ziel-Mastery nicht (die Codeprüfung vom 29. Oktober 2026 in `LearnerService.java` `isSrsMasteredByVerifiedRecall` und `useSrsMastery.ts` dokumentiert die Ist-Ableitung ohne Freigabe eines Mastery-Entzugs).

2.3 Prüfungsziele, Anerkennungsreview und Bewertung

Prüfungsknoten ergänzen `examData` (`taskContent`, optional `taskContentEn`, `solutionContent`, optional `solutionContentEn` sowie `scoring: {maxPoints, passingPoints, steps: [{id, points, description}]}`). Lernende dürfen Prüfungen direkt ohne vorherigen Abschluss der Vorbereitung versuchen; bei Nichtbestehen steuert `requires` die gezielte Nacharbeit. Gemäß dem korrigierten Konzept zur impliziten Beherrschung vom 4. November 2026 (Laufzeitanpassung ausstehend) müssen Prüfungen mit Anerkennungswirkung statisch darauf geprüft werden, dass keine Antwort durch kompensierende Nebenpunkte bestehen kann, obwohl eine anzuerkennende Kernkompetenz ausgelassen wurde. Nach den Gültigkeitsregeln für Lerneinheiten vom 11. November 2026 beansprucht ein Assessment Abdeckung (`coveredGoalIds`) nur für direkt oder transitiv vorausgesetzte Ziele (R3), jedes eigene gewöhnliche Ziel einer Einheit liegt auf einem Voraussetzungspfad eines eigenen Assessments dieser Einheit (R1), und mindestens ein solches Assessment trägt geprüfte Evidenz, dass Aufgaben und reguläre Bewertung das Ziel tatsächlich prüfen (R2); ein fehlender Pfad ist ein Befund für das Fachreview und niemals Anlass für eine redundante Kante (Graph-Definition §13.3).

Bewertung, Persistenz und **Prüfungsmodus (Exam Mode)** folgen festen Regeln:

- **Formeln:** total $= \min(\sum \text{stepPointsAwarded}, \text{scoring.maxPoints})$ und passed $= (\text{total} \geq \text{scoring.passingPoints})$ (Denkfehler ergeben `0` Schrittpunkte; reine Rechenfehler erlauben Teilpunkte).
- **Binäre Speicherung:** Bei `passed == true` wird Mastery unmittelbar nach der Bewertung auf `1.0` gesetzt (ohne zusätzliche Rückfrage; bei Schreibfehler darf keine Speicherung behauptet werden). Bei `passed == false` bleibt Mastery unverändert und Fehlversuch bzw. Punktzahl werden **nicht** gespeichert (unbegrenzte Wiederholungen bleiben möglich; die anschließende Lösungsanzeige ist eine bekannte Evidenzgrenze).
- **Ablauf der KI-Prüfungsaufsicht:** (1) Neutralen 2–4-zeiligen Prüfungskopf zeigen, `examData.taskContent` **wörtlich** ohne Aufteilung oder Hinweise ausgeben und eine feste Abgabezeile anhängen; (2) bis zur vollständigen Einreichung oder Aufgabe nur Lesbarkeitsrückfragen zulassen; (3) streng entlang `scoring.steps` bewerten; (4) Mastery nur bei Bestehen speichern; (5) Punkte, Ergebnis und Schrittfeedback nennen, `solutionContent` zeigen, in den Lerncoach-Modus wechseln und vor neuen Inhalten die ausdrückliche Wahl abwarten.

3 Sichtenprojektion und Goal-Placement

Eine sorgfältig lesende Person muss jeden sichtbaren Lernbaum allein aus dokumentierten JSON-Feldern rekonstruieren können – ohne Laufzeitcode zu lesen oder auf Titel- bzw. `phase`-Heuristiken zurückzugreifen.

3.1 Vier Sichtenfamilien und fünf unverhandelbare Projektionsregeln

| Sichtenfamilie | Beantwortete Leitfrage | Ausschließliche strukturelle Wahrheitsquelle | | — | — | — | | **Inhaltsbaum (Content Tree)** | Was gehört fachlich worunter? | `goals` + explizite `contains`-Kanten (geordnet nach `contains`-Array). | | **Programmbaum (Program Tree)** | Wo im Bildungsgang wird ein Ziel eingeführt, vertieft oder geprüft? | `programUnits` (`parentUnitId`, `order`) + `goalPlacements` (`primary` hängt ein; `secondary`/`assessed` als Badges/Overlays, fehlend liefert `unplacedInScope`). | | **Kompetenzsicht** | Entlang welcher Prozess- oder Kompetenzachse ist ein Ziel eingeordnet? | `competencyCatalog` + `competencyRefs` (orthogonale Taxonomieprojektion). | | **Composition-Baum** | Wie sieht ein aufgelöster Scope das Fach als einen Default-Baum? | Geprüfte `*.view.json` (`structure` + `canonicalSubtree` / `goalEntry`) + kanonische `contains`-Expansion. | {cols="20%,36%,44%"}

Fünf verbindliche Regeln gelten für alle Projektionen:

1. **Regel 1 (Keine Heuristik):** Niemals Eltern-Kind-Beziehungen aus Titeln, lokalisierten Labels, `phase`, `dimensionTags.phase` oder Namensähnlichkeit ableiten.
2. **Regel 2 (Explizite Kantenherkunft):** Jede sichtbare Baumkante stammt aus genau einem dokumentierten Feld (`contains`, `parentUnitId`, `goalPlacements`, `competencyRefs` oder `children` einer Composition View).
3. **Regel 3 (Eine Relationsfamilie je Default-Sicht):** Sichten dürfen Badges oder Filter ergänzen, aber niemals stillschweigend Elternhierarchien mischen.
4. **Regel 4 (Geprüfte Composition View hat Vorrang):** Existiert für einen Scope eine `.view.json`, definiert sie den lernendenseitigen Default-Baum, während `programUnits`/`goalPlacements` für Programmsichten und Diagnostik gültig bleiben (`ALL` ist ein reiner Query-Sentinel und wird nie in Placements oder Applicability serialisiert).
5. **Regel 5 (Einmalvorkommen je aufgelöstem Scope):** Innerhalb eines aufgelösten Scopes (z. B. `DE-HE / Gymnasium / SekII / LK`) erscheint jedes sichtbare Ziel **höchstens einmal** unter **höchstens einem** sichtbaren Elternknoten; überlappende `canonicalSubtree`-Expansionen sind Kompilierfehler.

3.2 Voraussetzungshülle, persönlicher Lernscope und kanonische Identität

Gemäß den Product-Owner-Entscheidungen vom **9. November 2026**, **10. November 2026** und **11. November 2026** (Graph-Definition §§5.1–5.2):

- **Unbeschnittene Voraussetzungshülle (09.11.2026) & robuste Lern-Lesezugriffe (11.11.2026):** Die Auswahl atomarer Ziele muss deren vollständigen transitiven Voraussetzungsbedarf auf dem unbeschnittenen kanonischen Graphen erhalten (`A requires B requires C` behält `B`, selbst wenn anfangs nur `A` und `C` dem Scope-Filter entsprachen). Unbekannte IDs, Zyklen oder nicht ausgewiesene Pflichtatome bleiben harte Fehler in Validierung, Curriculum-QS und nativer Abnahme; bei Lern-Lesezugriffen behält die Laufzeit im Fall eines nicht streng kompilierbaren kanonischen Kontexts den geprüften Seed-Scope ohne abgeleitete Hülle bei, beansprucht keine Lernbasis, bietet kein Ziel mit defektem Voraussetzungsbedarf an

und protokolliert den Fehler einmal je Curriculum und Fehlercode (Fehler in der Seed-Auswahl selbst bleiben harte Fehler).

- **Persönliche Lernscope-Basis L (10.11.2026):** Notwendige Voraussetzungen gehören zu Lernen, Fortschritt und Abschluss, nicht bloß zur unsichtbaren Nachschlagehülle. Aus den gültigen geprüften atomaren Seed-Zielen S wird auf dem unbeschnittenen kanonischen Graphen die feste globale Zielbasis $L = S \cup \{p \in A \mid \exists t \in S : (t, p) \in R_{\text{at}}^{+}\}$ kompiliert. L hängt ausschließlich von persönlicher Konfiguration und Curriculum-/Projektionsstand ab, **niemals** vom aktuellen Mastery-Stand. Beherrschte Atome bleiben als erfüllt in L; offene Voraussetzungen erhalten im vollständigen Persönlichen Curriculum navigierbare Erwerbsziele mit nachvollziehbarer `required by`-Provenienz (ohne neue Quellenmappings zu erfinden, Geschwisterknoten mitzuziehen oder Mastery zu schreiben). Der Level-3-Fokus ist der Schnitt der Nachfahren ausgewählter Wurzeln mit L (der Normalmodus blendet Prüfungen erst *nach* Bestimmung effektiver transitiver Erreichbarkeit auf den Fokus ab, ohne Pfade über Zwischenziele außerhalb des Fokus zu löschen).
- **Fachlokale `projectionRole`-Spezifität:** View-Referenzen tragen `projectionRole: "target "prerequisiteOnly"` (Standard `"target"`). Überlappungen lösen nach Spezifität auf: (1) direktes `goalEntry` vor `canonicalSubtree`, (2) tieferer/näherer `canonicalSubtree` vor breiterem Oberbaum, (3) `"target"` bei gleicher Spezifität. Ein fachlokal als `prerequisiteOnly` markiertes Ziel (etwa Brücken-/Formelziele in Hessen Sek-II-LK Mathematik oder Mathe-Stützziele in Physik) benötigt bei echtem Bedarf dennoch irgendwo im geschlossenen Persönlichen Curriculum L ein Erwerbsziel.
- **Erhaltene kanonische Atomizität:** Das Einklappen oder Referenzieren eines kanonischen Clusters erzeugt niemals ein opakes Ersatzatom und überträgt keine Cluster-Ergebnisse auf Kinder (siehe auch die historische `DPK-006c-c`-Grenze in Umsetzungsstand Duale Pakete sowie den Werkzeugschnitt in Curriculum-Autoreneditoren).

3.3 Lerneinheiten in projizierten Sichten

Projizierte Lerneinheiten (Zielkonzept; Laufzeitcompiler offen) binden an die geprüfte Composition View oder Programmprojektion:

1. **Ein direkter Kursbesitzer:** Jeder geplante atomare Kompetenzerwerb hat im aufgelösten Kurs genau eine direkt besitzende Einheit; Vorfahren aggregieren Kindpflichten ohne Doppelbesitz.
2. **Fachübergreifende Identität:** Gemeinsame kanonische Ziele (z. B. `target` in Mathematik und `prerequisiteOnly` in Physik) zählen im Gesamtplan einmal bei erhaltener Fachrolle.
3. **Vollständige Voraussetzungsbilanz:** Jedes benötigte Atom muss spätestens beim abhängigen Ziel durch die explizite Eingangsbasis oder einen geplanten Erwerb gedeckt sein.
4. **Lokale Anerkennung & Blockfolge:** Implizite Anerkennung gilt ausschließlich gemäß dem geprüften Vertrag der direkt besitzenden Einheit; Elternprüfungen öffnen weder abgeschlossene Kindeinheiten neu noch bestehen sie geschützte Kindprüfungen/Karten, und eine Blockfolge ganzer Einheiten bleibt eine separate explizite Kurspolitik.

4 Quellen- und Ressourcenlinks

SkillPilot setzt für externe Lernmaterialien ein striktes Umkehrprinzip durch: **„Content referenziert das Curriculum; das Curriculum wählt keine Anbieter aus."**

4.1 Dreiteilung der Linkzwecke und hartes Kernverbot

| Mechanismus | Träger | Zulässiger Zweck & Grenzen | | — | — | — | | **Amtliche Curriculum-Provenienz** | `sourceRef` sowie `resourceLinks` vom Typ `curriculum` | Rückverfolgung kanonischer Ziele auf amtliche Lehrpläne, Verordnungen und Fundstellen (6). | | **Eigene Zielvisualisierungen** | `resourceLinks` mit `type: "goal-visualization"` | Optionale kuratierte 16:9-Begriffsillustrationen in eigener SkillPilot-Verantwortung (7). | | **Externe Lernmaterialien** | Optionale Pakete unter `content/` (5) | Anbieterspezifische Lehrwerke, Simulationen, Videos und Aufgabensammlungen, die stabile kanonische `goalId`s von außen referenzieren. | {cols="24%,30%,46%"}

Verbindliche Regeln:

- **Keine neuen Drittanbieter-Links an kanonischen Zielen:** Kanonische `SkillLandscape`-JSON-Dateien dürfen keine externen Lehrbuch-, Video- oder Übungsplattform-Links in `resourceLinks` aufnehmen. Frühere Migrationsstände zu Übungslinks dienen ausschließlich als historische Provenienzevidenz und nicht als Vorlage für neue Anbieterlinks im Kerngraphen.
- **Datenschutzsichere Plain-URLs & ehrliche Coach-Grenze:** Ausgespielte Links müssen trackingfreie Plain-URLs ohne Lernenden- oder Sitzungs-IDs sein. KI-Coaches dürfen angebotene Links als optionale Vertiefung nennen, aber **niemals** behaupten, nicht abgerufene externe Webseiten gelesen zu haben.

5 Contentanbindung: Architekturrahmen

Dieses Kapitel definiert den verbindlichen Architekturrahmen v1.1 (Product-Owner-Entscheidung vom **21. September 2026**, GitHub Issue `#50`; PoC- und Ausbaudetails unter Physik-Libre-PoC, PoC-Betrieb und Content-Pakete zu Issue #54).

5.1 Dreischichtige Entkopplung und persönliche Auswahl

```
Kanonischer Skill-Graph (anbieterneutrale goalIds)
        ^
        |  von aussen referenziert ueber content/-Pakete
Optionale Contentlink-Pakete (Lehrwerke, Simulationen, Aufgaben)
        ^
        |  pro Lernendem ueber normale SkillPilot-ID gewaehlt
Persoenliche Materialauswahl (Cockpit & Coach, jederzeit abschaltbar)
```

Verbindliche Architekturprinzipien:

- **Anbieterneutraler Kern:** Kanonische Landschaften enthalten und bevorzugen keine externen Lehrbuch-, Video- oder Verlags-URLs; Contentpakete unter `content/` verweisen von außen auf stabile kanonische `goalIds`.
- **Personalisierung über die normale SkillPilot-ID:** Lernende aktivieren oder deaktivieren optionale Materialpakete im regulären Cockpit-Profil ohne Sonder-Keys, Lehrer-Accounts oder Drittanbieter-Logins.
- **Betriebsschalter (Kill-Switch) & Datenschutz:** Jedes externe Contentpaket lässt sich zentral per Betriebsschalter deaktivieren, ohne kanonische Graphen oder Lernstände zu verändern. Ausgespielte Links sind parameterfreie Plain-URLs ohne SkillPilot-ID oder Tracking-Token; Drittinhalte werden nicht umlizenziert.

5.2 Physik-Pilot und vier Abnahmekriterien

Der erste Referenzpilot bindet offen lizenzierte Physikbuch-Abschnitte (`https://physikbuch.schule/`) nach der Physik-`M7`-Stabilisierung an und wird in Issue `#54` auf Mathematik und LEIFIphysik erweitert. Jede Contentanbindung wird an vier verbindlichen Kriterien gemessen:

| Kriterium | Geforderter Nachweis | | — | — | | **1. Unveränderter Kern** | Das Abschalten oder Entfernen aller `content/`-Pakete lässt kanonische Graphen, Views, Frontier-Logik und Mastery zu 100 % unberührt. | | **2. Persönliche Auswahl** | Lernende steuern die Sichtbarkeit über ihr normales SkillPilot-Profil ohne Sonder-Keys oder Zusatzkonten. | | **3. Konkrete Funktion** | Aktive Links führen zielgenau auf passende Kapitel/Aufgaben und erscheinen sauber in Cockpit und Coach ohne erfundene Seiteninhalte. | | **4. Allgemeine Tragfähigkeit** | Ein zweiter Anbieter oder ein weiteres Fach benötigt nur ein deklaratives `content/`-Paket samt Validierung, keine Kernschema-Änderung. | {cols="26%,74%"}

6 Nachvollziehbare Quellenbegründungen

Jedes kanonische Lernziel einer SkillPilot-Skill-Landschaft muss nachvollziehbar auf amtliche Lehrplanevidenz zurückgeführt werden können, ohne dass Lehrkräfte oder Prüfende rohe JSON-Ledger oder SPARQL-Ausgaben lesen müssen (operative Befehle im Quellenbegründungen-Runbook).

6.1 Deterministischer Erklär-Umschlag und duale Nachweisrouten

`generateGoalSourceRationales.ts` erzeugt pro Lernziel einen deterministischen JSON-Umschlag (`schemaVersion: 1`), der zwei komplementäre Nachweisrouten zusammenführt:

- **Statusklassifikation (`sourceRationaleStatus`):** `classic_source_reviewed` (durch geprüfte amtliche Quellenextraktion und Mapping belegt), `partial` (unvollständige oder breitere Themenfeldabdeckung) oder `gap` (fehlender Nachweis, ehrlich unter `limitations` ausgewiesen).
- **Klassische Quellenroute (`classicSourceRoute`):** Bindet geprüfte `source-to-canonical`-Mappings (`exact` oder `partial`), amtliche Lehrplandokumente, Kapitelpfade und kurze Fundstellenauszüge.
- **MEM-SPARQL-Route (`memSparqlRoute`):** Ergänzt Abgleiche gegen externe Metadata-for-Educational-Media-(`MEM`-) bzw. FWU-Ontologie-Bestände (MEM-SPARQL-Audit), soweit vorhanden.

6.2 Deterministische Rangfolge, Ausspielmodi und Urheberrechtsdisziplin

Kandidatenbelege werden deterministisch sortiert:

1. `exact`-Mappings stehen vor `partial`-Mappings;
2. gültige aktuelle Landeslehrpläne stehen vor archivierten Altständen;
3. Belege des gewählten Bundeslands stehen vor bundesweiten oder länderübergreifenden Ersatzbelegen.

Im Standalone-Paketmodus (11) liefert die Laufzeit generationsgebundene Evidenz über Catalog-1.2-Discovery statt über fest verdrahtete Repositorypfade und fällt bei Fehlern nicht auf statische Altdateien zurück. Lehrplanzitate bleiben auf knappe Fundstellennachweise beschränkt (keine Volltextübernahme amtlicher Dokumente), um Urheber- und Verwertungsrechte strikt zu wahren.

7 Visualisierungen für atomare Lernziele

Optionale Begriffsillustrationen unterstützen Lernende dabei, die Kernidee eines atomaren Lernziels visuell zu erfassen, ohne eigenes Nachdenken oder Aufgabenlösen vorwegzunehmen.

7.1 Kanonisches Linkschema und MCP-App-Darstellungsvertrag

Geprüfte Visualisierungen werden über `resourceLinks` am atomaren Lernziel referenziert:

```
{
  "type: "goal-visualization",
  "resourceType: ïmage",
  "role: "primary",
  ßkillpilotId: «exakte-kanonische-goal-id>",
  ürl: /assets/goal-visualizations/<subject>/<id>/<id>.png"
}
```

In MCP-fähigen Hosts wie OpenAI ChatGPT liefert `render_skillpilot_goal_visualization` ein oberflächenneutrales `text/html;profile=mcp-app`-Widget mit einmaliger `(goalId, state Version)`-Autorisierung und dem Schalter `showGoalVisualizationsInChat`. Das Widget bleibt bis zum erfolgreichen Bildladen eingeklappt, wertet begrenzte MCP-Ergebnisumschläge aus (mit gespeichertem Widget-State nur als Bootstrap-Fallback) und bewahrt alle zuvor veröffentlichten inhaltsadressierten UI-Ressourcen-Hashes unverändert.

7.2 Produktionsbereich, Format und Provider-Priorität

Prompts, Kandidatenerzeugung und QA-Ledger liegen unter `curricula/DE/Gymnasium/visualizations/` (`prompt.de.md`, `image-reconstruction-prompt.de.md`, Ausspielung per `npm run deploy:assets` als 16:9-PNG um `1600x900`, geprüft auf `360px` Mobil- und `680px` Desktopbreite, kalibriert am Pilotziel `502ecaa7-cca6-5c51-a1cc-da09a7b2382c`).

Gemäß verbindlicher **Product-Owner-Entscheidung vom 13. September 2026** gilt:

- **Standard `KEEP`:** Gute bestehende Mathematik- und Physikbilder bleiben unabhängig vom ursprünglichen Generator erhalten; es erfolgt keine rückwirkende Konvertierung zur Provider-Vereinheitlichung.
- **Provider-Priorität bei Neu- oder Korrekturbildern:** Bevorzugt wird **ChatGPT/Codex-Bildgenerierung** in der freundlichen, abstrakten, klaren, comicartigen Raster-Bildsprache von SkillPilot, sofern Fachrichtigkeit und Darstellung passen. **Nano Banana Pro** (`gemini-3-pro-image`) ist ohne vorherigen Fehlversuch als reguläre Alternative zulässig, wenn es die fachliche Darstellung besser trifft.
- **Enge SVG-/Plot-Ausnahme:** Programmatisch gezeichnete Diagramme/SVGs bleiben eng begrenzte, fachlich begründete Ausnahmen für exakte Geometrie-/Funktionsplots und kein allgemeiner Stilwechsel.

7.3 CLI-Workflow und zweistufige Review-Gates

Die Hilfsskripte unter `app/` (`visualization:generate:nano-banana`, `visualization:plan-batch`, `visualization:check-prompt-appends`, `visualization:prepare`, `visualization:import`) trennen Generierung strikt von Freigabe:

1. **KI-QA-Ledger (`aiApprovedAssetSha256` / Gate V):** Speichert strukturierte Prüflabels (`accepted_pilot`, `accepted_pilot_after_regeneration`, `rejected_regenerate`, `deferred_provider_limitation`, `needs_external_review`), gebunden an den exakten Datei-SHA-256.
2. **Menschliches Publikationsgate (`humanApproved`, `HR-003`):** Öffentliche Standalone-Releases und `public`-Lernzielbücher (8, Menschliche Review-Gates) verlangen eine ausdrückliche menschliche Fach- und Rechtefreigabe (`CC-BY-4.0` getrennt von Provenienz); bloße Generierung oder KI-Prüfung ersetzt keine Publikationsfreigabe.

8 Lernzielbuch und Evidenz-Review-Pipeline

Das **Lernzielbuch** und die Pipeline für positive Verständnis-Evidenzprofile machen jedes curriculare atomare Lernziel für Menschen und Multi-KI-Reviews prüfbar, ohne die lernendenseitigen Zieltexte aufzublähen (die darauf aufbauende Unterrichts-Zeitplanung für Lehrkräfte ist in Lernzielbuch-Zeitplanung beschrieben).

8.1 Gemeinsames BookModel, Ein-Ziel-pro-Seite-Atlas und Evidenzprofile V2

Ein deterministisches `BookModel` (`contracts/goal-book/v1/goal-book-model-1.1.schema.json`) speist identisch die schreibgeschützte `/lernzielbuch`-WebGUI und das A4-Hochformat-PDF:

- **Aktuelle bundesweite Review-Ausgaben (30.08.2026): 792** eindeutige curricular-atomare Zielseiten für Gymnasium Mathematik und **461** Seiten für Physik über alle 16 Bundesländer (`publicationMode: review`, mit QA-Ledger-gebundenen Kandidatenbildern ohne Behauptung menschlicher Bildfreigabe; Coach-Runtime und Mastery-Schreibpfad bleiben unverändert).
- **Vierzehn verbindliche Invarianten:** Knappe kanonische `title`/`description`; separate fingerprint-gebundene V2-Evidenzprofile (`contracts/goal-evidence/v2/goal-evidence-profile.schema.json` mit `expectations`, `coverageExpectations`, `variationAxes` und `applicationCaseBriefs`; V1 bleibt als Archivhistorie erhalten); **genau eine physische Seite je regulärem atomarem Ziel** mit vollständiger **Lernziel-ID**, Kapitelpfad (`canonical-composition-view-v1` + **Bundeslandspezifische Ergänzungen** mit lückenloser `treeOrder`), stabiler Kahn-Topologiesortierung entlang `requires`, klickbaren PDF-Sprungzielen (`goal-<uuid>`), Fail-Closed-Überlauf-/Lesbarkeitsprüfung, inkrementellen Skia-PDF-1.4-Lesezeichen, bundeslandprimären Geltungstupeln (`Bundesland -> [{stage, durationModel, courseProfile}]` – niemals kartesische `G8`/`G9`-Kreuzprodukte) sowie nachgeladenen `/lernzielbuch/<bookId>.original-sources.json`-Quellenmatrizen.
- **Anwendungsbasierte Anerkennungsprüfung (04.11.2026):** Prüfungen mit globaler Kompetenzanerkennung (Implizite Beherrschung) verlangen vor Freigabe ein statisches Review gegen kompensierende Nebenpunkte.

8.2 Multi-KI-Review-Schleife, Evidenzgates und moderierter Feedback-Transfer

Die Qualitätssicherung verbindet eine schnelle blinde Multi-KI-Schleife mit strengen Verallgemeinerungsgates und einem langsamen, datensparsamen öffentlichen Feedbackkanal:

| Skala | Stufen & Bedeutung | | — | — | | **Evidenz (`E0`–`E5` + `NORMATIVE`)** | `E0` unbelegte Meinung → `E1` verankerte Beobachtung → `E2` reproduzierbares Minimal-Gegenbeispiel → `E3` unabhängige Multi-Modell-/Prompt-Triangulation → `E4` kontrollierte Feldevidenz → `E5` replizierte Kohortenevidenz. | | **Geltungsumfang (`G0`–`G4`)** | `G0` Einzelfall → `G1` einzelnes Ziel → `G2` Zielarchetyp (erfordert ≥ `E3` über ≥ 2 Kapitel) → `G3` Fach/Stufe (geschichtete Stichprobe + Replikation) → `G4` globale Coach-/Mastery-Regel (Mathe + Physik + Feld-Canary). | {cols="30%,70%"}

- **Zweistufige V2-Kampagne:** Kalibriert an 16 Mathematikzielen (`5`–`10`, `E`–`Q4`, Pilotziel `8dd9f210-2683-5902-acab-e3be22725232`) vor dem 792-Ziele-Lauf: (1) Atlas-V1 einfrieren, (2) blindes Codex-V1-Review (`keep` | `revise` | `split_review` | `block`), (3) deterministischer Check + einzelzielweise menschliche Entscheidung, (4) kontrollierte zweisprachige V2-Pflege + Folgelane-Prüfung, (5) blindes Zweitanbieter-V2-Review, (6) dissenzwahrende Synthese. Weder Modellmehrheit noch Ausbleiben von Kritik geben Änderungen automatisch frei.

- **Zweiphasiger `/lernziel-feedback`-Transfer:** In Produktion standardmäßig deaktiviert und durch `SKILLPILOT_GOAL_FEEDBACK_OPERATOR_TOKEN` geschützt; verarbeitet datensparsame V2-Umschläge ohne Lernenden-/SkillPilot-ID, Kontaktfeld oder Dateiupload. `POST /api/operations/goal-feedback/v1/export-batches` exportiert deterministische Batches samt serverseitigem Publikationssnapshot nach `tmp/goal-feedback/inbox/` (`trusted-context.jsonl` strikt getrennt von `untrusted-feedback.jsonl`); erst nach lokaler Byte-/Digest-Prüfung löscht ein idempotentes `DELETE .../{exportId}` mit `If-Match` die Live-Zeilen unter Erhalt einer inhaltsfreien Löschquittung.

```
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 Rollout-Politik für das kanonische Gymnasium

Dieses Kapitel legt die dauerhafte Zielarchitektur und Rollout-Politik fest, nach der die deutschen Gymnasiallehrpläne aller 16 Bundesländer ohne bundeslandspezifische Zielduplizierung in je einen kanonischen Skill-Graphen pro Fach zusammengeführt werden (Umsetzungspläne und Migrationsstände liegen unter Implementierungsplan Kanonisches Gymnasium und Migrationsstatus).

9.1 Neun Kernentscheidungen und Scope-Vokabular

1. **Kanonische Fachebene zuerst:** Eine durchgängige `SkillLandscape` je Fach unter `curricula/DE/Gymnasium/canonical/` von Sekundarstufe I bis zur Abiturreife.
2. **Kontinuität in der Übergangsphase:** Abgelöste Landesbäume bleiben als bundeslandgebundene Quellen- und Auditeingaben unter `curricula/DE/Gymnasium/input/<STATE>/` erhalten (abgesichert durch die Stilllegungsvalidatoren in Graph- und View-Validierung), während Mappings bestehenden Lernfortschritt überführen.
3. **Schlanke Mapping-Schicht:** Explizite `legacyGoalId -> canonicalGoalId`-Mappings (`matchType: exact "partial"`) statt eines zweiten Inhaltsuniversums.
4. **ISO-3166-2-Ländercodes:** Metadaten und Filter nutzen durchgängig `DE-HE`, `DE-BY`, …, `DE-TH` (`ALL` ist ein reiner Query-Sentinel und wird nie serialisiert).
5. **Stabile Coach-/MCP-Schnittstellen:** Backend und Laufzeit kapseln die Übersetzung zwischen Alt- und Kanon-Schicht, sodass KI-Coaches stets genau einen Lernstand, eine Frontier und ein Mastery-Modell sehen.
6. **Kontrollierte fachübergreifende `requires`:** Kanonische Fachlandschaften dürfen fachübergreifende Kanten direkt auf stabile kanonische IDs setzen (z. B. Physikziele mit Mathematik-Voraussetzungen), ohne fremde Teilbäume zu kopieren; die Graphvalidatoren lösen diese Verweise über die geladenen Landschaften auf, und `CQR-001` bezieht benötigte fachfremde Ziele samt vererbender Vorfahren in die Prüfung auf atomare Normalform und transitive Minimalität ein.
7. **Gemeinsame Wurzel `Gymnasium (DE)` & geprüftes Scope-Vokabular:** Ein bundesweiter Einstiegsknoten bündelt Fachlandschaften entlang `jurisdiction` (`DE-*`), `schoolForm` (`Gymnasium`), `stage` (`SekI`, `SekII`), `durationModel` (`G8`, `G9`) und `courseProfile` (`GK`, `LK`, `GK+LK` ohne doppelte IDs) auf Basis expliziter `.view.json`-Dateien (3).
8. **G9-Referenzraster für Sek I:** Kanonische Sek-I-Landschaften orientieren sich am gemeinsamen Raster `Jahrgang 5` bis `Jahrgang 10`, während `G8`/`G9`-Quellwahrheit in Provenienz und Views erhalten bleibt.
9. **Bundeslandgebundene Quell-Assets:** Nicht-kanonische Quellenextrakte, Abitur-Blueprints (`curricula/DE/Gymnasium/input/HE/abi/`) und Auditbelege bleiben strikt ländergetrennt unter `input/<STATE>/`.

9.2 Teilbaum-Zustandsmodell und Einführungsreihenfolge

Migrationseinheit ist stets ein **didaktisch abgeschlossener Teilbaum** (kein Big-Bang-Umbau) entlang vier geordneter Zustände:

Zustand	Bedeutung & Lernstandsvertrag
`1. legacy_frozen`	Bestehende Landes-JSONs dienen schreibgeschützt als Quelle; Laufzeit-/Mappingkorrekturen ohne Umbau des Altbaums erlaubt.
`2. subtree_adopted`	Abgeschlossener `contains`-Teilbaum unter `canonical/` mit expliziten `exact`/`partial`-Mappings und gebundenen Voraussetzungen angelegt.
`3. cutover_ready`	Lernstandskontinuität, Frontier-Stimmigkeit und Mappingabdeckung belegt, sodass der kanonische Pfad Standard werden kann.
`4. legacy_view_retained`	

Kanonischer Teilbaum ist primärer Lernpfad; Kompatibilitätssichten bleiben vorübergehend bis zur regulären Stilllegung verfügbar. | {cols="26%,74%"}

Strategische Reihenfolge: hessische Lernstandskontinuität sichern → Mathematik über Sek I und Sek II als zentrales Voraussetzungsrückgrat aufbauen → Bayern und weitere Länder per Mapping und `.view.json`-Projektionen einbinden → mehrfächige kanonische Navigation und gezielte Fach-zu-Fach-Voraussetzungen ausbauen.

10 MEM/FWU-Roundtrip-Plan

SkillPilot versteht Metadata for Educational Media (`MEM`) und `FWU-DE/lehrplan-ontologie` als benachbarte semantische Metadaten- und Provenienzschicht statt als Gegenmodell und dient als offener Prüfstand für nachvollziehbare KI-Lernschritt-Navigation (operative Pipeline in MEM/FWU-Roundtrip-Pipeline; Referenzpapier <https://aifyer.com/ki-lernbegleitung/versions/v1.0/schulisch-verantwortete-ki-lernbegleitung-v1.0.pdf>).

10.1 Semantischer Roundtrip-Test und geforderte Abdeckung

```
SkillPilot-Skill-Landschaft (JSON-Export-ZIP)
  -> MEM/FWU-first-RDF/OWL-Bundle + gehashte Binaer-Sidecars
  -> Rekonstruierte SkillPilot-Skill-Landschaft (ohne Quell-JSON-Zugriff)
  -> Deterministische Graph-, View- und Asset-Validierung
```

Ohne opake JSON-Blobs oder Base64-Medienliterale bewahrt das semantische Bundle:

- Zielidentität (`id`, `shortKey`, `title`, `description`), Atomar-/Cluster-Klassifikation, `weight`, `core` und Dimensionsmetadaten;
- direkte `contains`-Komposition und didaktische `requires`-Voraussetzungen;
- Programmstruktur (`programUnits`), `goalPlacements`, Kompetenzachsen (`competency Catalog`, `competencyRefs`) und lernendenseitige Composition Views;
- amtliche Quellen-/Provenienzreferenzen sowie `goal-visualization`-Verweise (`schema:Image Object`) samt bytegleichen SHA-256-Bild-Sidecars (persönliche Lernstände bleiben außerhalb öffentlicher Pakete).

10.2 Core-First-FWU-Abbildung und Anwendungsprofil

| Relation / Konzept | Core-First-RDF/OWL-Darstellung & Modellgrenzen | | — | — | | **Didaktisches `requires`** | Spezialisierte reifizierte `LP_0000554`-Ressource über `LP_0030071` (`hat Verweis`) und `LP_0030072` (`verweist auf`, eingeführt in `FWU-DE/lehrplan-ontologie#9`). Ein generischer `CE-Verweis` darf **niemals** pauschal als Voraussetzung gedeutet werden. | | **Curriculares `contains`** | `BFO_0000051` (`hat Teil`) nur bei echter fachlicher Teil-Ganzes-Beziehung, zusätzlich flankiert durch `skillpilot:containsGoal` auf jeder direkten Autorenkante, damit transitive BFO-Inferenzen die Direktkanten nicht verwischen (Übungs-/Prüfungs-/Karten-/Navigationskanten nutzen nur `skillpilot:containsGoal`). | | **Minimale Anwendungsterme** | `skillpilot:{LearningGoal,AtomicGoal,ClusterGoal,containsGoal,weight,core,shortKey,ProgramUnit,GoalPlacement,CompositionView,GoalVisualizationReference}` sowie `schema:ImageObject` für Binär-Sidecars. | {cols="28%,72%"}

Erfolgskriterien sind vollständige SkillPilot-Validierung (zyklenfreie `requires`/`contains`, referenzielle Integrität, gültige Composition Views), deterministische Feldäquivalenz, OWL-2-DL-/Core-Strukturprüfung und byteidentische Bild-SHA-256-Prüfung ohne versteckten Rückgriff auf das Ursprungs-JSON.

11 Duale Curriculum-Pakete: JSON-Runtime und Lehrplan-Ontologie

Jeder versionierte Curriculum-Release veröffentlicht **genau einen fachlichen Stand** in zwei semantisch gleichwertigen Darstellungen – ohne doppelte Fachpflege und ohne Zugriff einer Skill Pilot-Installation auf den `curricula/`-Quellbaum (aktueller Checkpoint: Umsetzungsstand Duale Pakete; operative Verträge: Paketbereitschaft, Release-Modell-Konformität, Duale Release-Verträge und Paketprovisionierung).

11.1 Sieben verbindliche Architekturentscheidungen und Release-Artefakte

```
Authoring-Quellen -> Release-Compiler -> Normalisiertes Release-Modell
  -> JSON-Runtime-ZIP (<id>-<ver>.json.zip)
       -> Core-first-Ontologie-Exporter -> FWU/OWL-ZIP (<id>-<ver>.fwu-owl.zip)
            -> Isolierter Reverse-Compiler (bubblewrap, kein Quell-JSON)
                 -> Rekonstruiertes JSON-Modell -> Semantisches Aequivalenz-Gate
```

1. **Keine doppelte Fachpflege:** Beide ZIPs entstehen aus demselben Quellstand; Korrekturen erfolgen ausschließlich in den Authoring-Quellen.
2. **Das JSON-ZIP ist die normative Runtime-Referenz (`full-standalone-v1`):** Enthält `metadata/manifest.json` (explizite Artefaktrollen, Offline-`schemas/catalog.json`), `data/runtime/{catalog.json,dependency-closure.json,migration-aliases.json}`, fertig kompilierte `applicability`, `canonical/`, `views/`, `cards/`, `resources/`, `mappings/`, `sources/`, `metadata/quality/`, `assets/` und `licenses/`, lauffähig ohne `curricula/`- oder `app/public`-Fallbacks unter festen ZIP32-Grenzen (`3,5 GB` Archiv, `60.000` Einträge, `64 MiB` je JSON/Bild, JSON-Tiefe `128`, Kompressionsfaktor `100:1/25:1`).
3. **Das Ontologie-ZIP wird strikt aus dem fertigen JSON-ZIP abgeleitet:** Enthält `rdf/{declarations,runtime,landscape,views,mappings,sources,cards,assets,bundle}.nt`, `skillpilot-curriculum-profile.ttl`, `catalog-v001.xml`, gepinnte `ontology/{lehrplan-core.owl,shapes.ttl}` sowie identische Binär-Sidecars (geprüft auf RDF-Syntax, SHACL, OWL 2 DL via ROBOT und HermiT-Konsistenz).
4. **Semantische Äquivalenz & gemeinsamer `contentDigest`:** Gesteuert durch eine geschlossene **454-Regeln-Feldsemantik-Registry** (`323` Runtime-Felder + `130` Publikationsnormalformen + `1` Historieneintrag; `scalar`, `ordered-list`, `set`, `map`, `binary-reference`, `generated-non-semantic`; abgebildet über `fwu-core`, `skillpilot-profile`, eng begrenztes `registered-canonical-json-literal` oder `excluded-generated`). Unregistrierte Felder blockieren den Build; Placement-Identitäten lauten `goalId@unitId@index`.
5. **Core-first statt Core-only:** Deklarative Fachprofile bilden das verpflichtende `semanticKind` (`curricularAtomic`, `curricularArea`, `programStructure`, `practiceAssessment`, `memory`, `orientation`, `runtimeSupport`) ohne Titelraten ab.
6. **Keine versteckte JSON-Vollkopie:** Zwei netzlose `bubblewrap --clearenv`-Läufe (`DPK-008d`) beweisen die verlustfreie Rekonstruktion allein aus RDF und Sidecars.
7. **Externer Release-Index & Signing-Projection:** `<id>-<ver>.release.json`, `.equivalence.json`, `.provenance.json`, `.SHA256SUMS` und `.release.json.sig` (Entfernung von `authentication.releaseIndexSignature` vor Kanonisierung/Hash) samt signiertem Veröffentlichungskatalog (`staging` → `stable`, niemals ungepinntes `latest`).

11.2 Generische Runtime-Entkopplung, Versionierung und Phasen 0–6

Vier generische Bausteine lösen SkillPilot von Repositorypfaden: `CurriculumPackageRepository` (Read-only-Lock-/CAS-Leser), `JsonCurriculumPackageLoader` (schema- und hashprüfender

Modellaufbau), `CurriculumResourceResolver` (versionsgebundene Hrefs für `(packageId, deckId, locale)` und `(packageId, resourceId)`) sowie `CurriculumActivationService` (Quarantäne-Extraktion, Dry-Run-Prüfung, atomarer CAS-Lock-Wechsel, Rollback). Versionsachsen bleiben orthogonal (`packageVersion`, `curriculumEdition`, `packageFormatVersion`, `runtimeContractVersion`, `supportedSkillpilotSoftware` aktuell `>=0.1.0 <1.0.0` für Consumer `0.1.0`, `ontologyProfileVersion` sowie gepinnter FWU-Core-Commit/SHA), während `migration-aliases.json renamed` (automatische Mastery-Übernahme) von `replacedBy`, `splitInto`, `mergedInto` und `removed` trennt. Lizenzen führen `licenseExpression`, `provenanceClass` und `redistributionStatus` getrennt (`review-required` hält Staging-Kandidaten ehrlich auf `not-ready-incomplete`).

| Phase | Umfang & erreichter Meilensteinstand | | — | — | | **Phase 0** | **Technisch abgeschlossen (`DPK-004` / `DPK-007a`):** Eingefrorener Mathematikstand `0.1.0-conformance.3` (`914` ZIP-Einträge, `757` Bilder, 454-Regeln-Registry; bewusst `not-ready-incomplete` bis zum Abschluss menschlicher Bild-/Rechte-/Quellentext-Gates). | | **Phase 1** | **Technisch abgeschlossen (`DPK-005b`–`DPK-007`):** Hermetischer `bubblewrap`- + Corretto- + Playwright-Consumer verifiziert (`1.079` Ziele, `33.382` Source-Mappings, `88` Views, `12` Decks, `757` Bilder, null Repository-Zugriffe unter `strace`). | | **Phase 2** | **Technisch abgeschlossen (`DPK-008a`–`DPK-008d`):** Reproduzierbares FWU-OWL-Paket (`824.452` Tripel, SHACL/ROBOT/HermiT grün) und isolierte bytegleiche JSON-ZIP-Rekonstruktion (`911` Einträge, `111` logische Artefakte). | | **Phase 3** | **Nächster Schritt (`DPK-009`, `not_started`):** Nachgelagerter Äquivalenz-Oracle-Abgleich, dualer hermetischer Consumer-Smoke-Test, isolierte `JSON -> FWU`-Paarreproduzierbarkeit und signierter externer `release.json`. | | **Phasen 4–6** | Mehr-Fächer-Ausbau (Mathe → Physik mit eingebetteter Closure → Sprachen → Chemie) → produktiver Mehrpaket-/Upgrade-Betrieb → Repository-Trennung zwischen Curriculum-Tooling und SkillPilot-Software. | {cols="18%,82%"}