SkillPilot Glossar

Gemeinsames Fachvokabular für Curricula, Skill-Graphen, Lernlaufzeit, Coach-Adapter und Qualitätssicherung

Zweck dieses Glossars

Dieses Dokument definiert die zentralen Fachbegriffe von SkillPilot und grenzt nah verwandte Konzepte präzise voneinander ab. Es bildet das gemeinsame Vokabular für Curriculum-Autorinnen und -Autoren, Curriculum Champions, Entwicklerinnen und Entwickler sowie KI-Coach-Adapter.

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

Verbindlich bleibt stets das beim jeweiligen Begriff verlinkte Fachdokument.

Wie dieses Glossar zu lesen ist

- **Definitionen sind bewusst knapp gehalten.** Jeder Eintrag benennt, was der Begriff bedeutet und – wo Verwechslungsgefahr besteht – was er *nicht* ist. Maßgeblich bleibt immer das verlinkte Fachdokument; dieses Glossar überschreibt keine Spezifikation.
- **EN** nennt jeweils das englische Äquivalent aus Code, englischer Dokumentation und Spezifikation. Feldnamen wie `contains`, `requires` oder `applicability` sind technische Bezeichner und bleiben in beiden Sprachen englisch.
- **Begriffe sind nach Schichten geordnet**, von den Curriculum-Quellen bis zur Qualitäts-Governance, da Missverständnisse meist durch das Vermischen von Schichten entstehen.

Zwei Grundregeln erklären einen großen Teil des Vokabulars:

- **Lernziele sind stabil, Struktur ist kontextabhängig.** Ein Lernziel ist eine erlernbare, überprüfbare Kompetenz mit stabiler Identität; Jahrgänge, Phasen, Kurse und Bundesländer bilden den Kontext darum herum.
- **So viel Semantik wie nötig, so wenig Ontologie wie möglich.** Eine neue Unterscheidung wird nur eingeführt, wenn sie ein konkretes Problem bei Modellierung, Migration, Validierung, Projektion oder Laufzeit löst.

Inhaltsverzeichnis

Wie dieses Glossar zu lesen ist 2

1 Curricula, Skill-Graphen und Herkunft 4

2 Im Inneren des Skill-Graphen 5

3 Knotentypen (Lernmodi) 9

4 Strukturschichten und Projektion 10

5 Lernlaufzeit 14

6 Coach, Session und Identität 17

7 Qualität, Review und Governance 19

8 Wichtige Abgrenzungen auf einen Blick 20

Pflegehinweise 21

1 Curricula, Skill-Graphen und Herkunft

Curriculum / Lehrplan

EN: Curriculum — siehe Whitepaper

Das externe, durch Bildungsbehörden oder Akkreditierung festgelegte Regelwerk, an das SkillPilot andockt: ein staatlicher Lehrplan, ein Modulhandbuch oder ein Standard wie der GER/CEFR. Das Curriculum bleibt die **normative Quelle**; SkillPilot erfindet keine Standards, sondern macht bestehende operational nutzbar.

Nicht identisch mit dem daraus abgeleiteten Skill-Graphen. Im Qualitäts-Dashboard bezeichnet „Curriculum“ enger gefasst eine einzelne kanonische Skill-Landschaftsdatei.

Skill-Graph

EN: Skill graph — siehe Graph-Definition

Das versionierte, maschinenlesbare **operative Modell**, das aus einem Curriculum abgeleitet wird: Lernziele, verbunden über `contains` und `requires`, zusammen mit ihrer Semantik, ihren Gültigkeitsbedingungen und der Frontier-Logik. Den Skill-Graphen baut, validiert und navigiert SkillPilot; das Curriculum bleibt die übergeordnete normative Quelle.

Skill-Landschaft

EN: Skill landscape — siehe Graph-Definition

Eine abgegrenzte, veröffentlichte Instanz eines Skill-Graphen: eine nach Fach und Schulform bestimmte Einheit mit stabiler `landscapeId`, betreut von Curriculum Champions, bewertet mit einem Reifegrad und ausgeliefert als Release-Paket. Eine Skill-Landschaft kann in mehrere Sprachdateien mit derselben ID serialisiert sein.

Skill-Graph bezeichnet das formale Modell, *Skill-Landschaft* die **konkrete** fachliche Instanz.

Kanonische Landschaft

EN: Canonical landscape — siehe Canonical Gymnasium Rollout Policy

Eine Skill-Landschaft, die einmalig für eine gesamte Schulform gepflegt und von allen Bundesländern gemeinsam genutzt wird, statt je Land einen eigenen Graphen zu duplizieren. Länderspezifische Unterschiede werden über Gültigkeit (`applicability`) und Kompositionssichten ausgedrückt, nicht durch Kopieren von Zielen.

Wurzelcurriculum und Curriculum-Manifest

EN: Root curriculum and curriculum manifest — siehe Levels of Personalization

Ein Wurzelcurriculum ist eine Skill-Landschaft, die in der Oberfläche direkt auswählbar und für die Champion-Registrierung zugelassen ist. Die Menge der Wurzelcurricula wird explizit in curricula/curriculum_manifest.json deklariert; die CI schlägt fehl, wenn Manifest und berechnete Wurzelmenge voneinander abweichen.

Quellen-Snapshot

EN: Source snapshot — siehe Curriculum-Qualität

Ein im Repository gespeicherter, maschinell auswertbarer Auszug eines offiziellen Curriculum-Dokuments samt offiziellem HTTP(S)-Verweis auf das Original. Abdeckungsnachweise werden ausschließlich gegen die im Repository vorliegenden Snapshots geprüft.

Originalziel und Source-Registry

EN: Source goal and source registry — siehe Curriculum-Qualität

Ein Originalziel (*Source-Ziel*) ist eine Kompetenzformulierung aus dem ursprünglichen Lehrplan. Die Source-Registry verzeichnet die Originalziele und bildet deren **atomare Quellenhülle** auf kanonische Lernziele ab. Die Abdeckung wird in beide Richtungen geprüft: Jedes sichtbare atomare Lernziel benötigt einen Quellennachweis, und jedes registrierte Originalziel muss vollständig in die Sicht abgebildet werden.

`sourceRef` vs. `resourceLinks`

EN: `sourceRef` vs `resourceLinks` — siehe Source And Resource Links

`sourceRef`, Zuordnungen und Herkunftsdaten beantworten die Frage „Warum existiert dieses Lernziel?“. Offizielle Lehrplanreferenzen sind Nachweise, kein optionales Lernmaterial.

Der Architekturrahmen für die Contentanbindung legt optionale externe Lernmaterialien in separate Pakete aus, die Lernziel-IDs referenzieren; Lernende wählen Pakete unabhängig vom Curriculum. `resourceLinks` bleibt das Kompatibilitätsfeld für Bestandsressourcen und die gesondert geprüften SkillPilot-eigenen Zielvisualisierungen. Direkte externe Material-Links in kanonischen Zielen gelten als Migrationsschuld, nicht als Zielmodell.

Lernmaterial darf niemals als Herkunftsnachweis getarnt werden, und Provenienz ist kein Studienlink.

Runtime-Paket und Ontologie-Paket

EN: Runtime package and ontology package — siehe Dual Curriculum Package Releases

Die zwei äquivalenten Release-Artefakte einer Curriculum-Veröffentlichung: ein JSON-Runtime-Paket für die Anwendung und ein FWU-kompatibles Ontologie-Paket für den semantischen Datenaustausch. Beide entstehen aus derselben geprüften Release-Einheit.

2 Im Inneren des Skill-Graphen

Lernziel

EN: Goal — siehe Graph-Definition

Ein Knoten des Skill-Graphen: eine erlernbare, überprüfbare Kompetenz, in der Regel formuliert als „Die lernende Person kann …“. Jedes Lernziel besitzt eine stabile UUID `id`, einen `title` und ein strikt positives Gewicht `weight`.

Ein Jahrgang, Halbjahr, Modul, Kurstyp oder eine Phase ist **kein** Lernziel.

Atomares Lernziel

EN: Atomic goal — siehe Graph-Definition

Ein Lernziel ohne `contains`-Kinder. Atomare Ziele bilden die überprüfbaren Grundeinheiten: Auf ihnen wird Mastery gespeichert, über ihnen wird die Frontier berechnet, und im Zielzustand verlaufen alle fachlichen `requires`-Kanten zwischen atomaren Zielen.

Die Eigenschaft ergibt sich aus der Graphstruktur, nicht aus freier Etikettierung; gespeicherte Klassifikationsmetadaten müssen mit der Struktur übereinstimmen.

Cluster-Ziel

EN: Cluster goal — siehe Graph-Definition

Ein Lernziel mit mindestens einem `contains`-Kind. Cluster dienen der Navigation, Filterung und aggregierten Fortschrittsanzeige. Ihre Erfüllung leitet sich aus ihren atomaren Nachfahren ab und wird niemals direkt als eigenes Lernergebnis gespeichert.

Atomare Basis

EN: Atomic basis — siehe Graph-Definition

Geschrieben Atoms(g): das Ziel selbst, falls es atomar ist, andernfalls die Menge seiner atomaren `contains`-Nachfahren. Ihre vollständige Beherrschung bezeugt die Erfüllung eines Clusters.

`contains`

EN: `contains` — siehe Graph-Definition

Die modellierte Hierarchierelation C: (p, c) bedeutet, dass Elternknoten p das Kind c **inhaltlich** unmittelbar enthält. Der Pfeil zeigt vom Elternknoten zum Kind. `contains` muss azyklisch sein, erlaubt aber mehrere Elternknoten (Polyhierarchie).

Vorfahren- und Nachfahrenbeziehungen ergeben sich über Pfade aus einer oder mehreren `contains`-Kanten. `contains` ist weder eine Voraussetzung noch eine Programmplatzierung.

`requires`

EN: `requires` — siehe Graph-Definition

Die modellierte Voraussetzungsrelation R: (g, p) bedeutet, dass Lernziel g die Voraussetzung p unmittelbar benötigt. Der Pfeil zeigt vom abhängigen Ziel zur Voraussetzung; p muss sitzen, bevor g bearbeitet wird. Die Relation muss azyklisch sein. Im verbindlichen Zielzustand verlaufen sämtliche modellierten `requires`-Kanten zwischen **atomaren** Zielen ($R \subseteq A \times A$); bestehende Voraussetzungen auf Clustern werden während der Migration über `contains`-Vererbung und atomare Expansion ausgewertet.

Sowohl `contains` als auch `requires` speichern direkte Kanten; abgeleitete Beziehungen werden berechnet und nicht als zusätzliche Kantenarten modelliert.

Vererbte Voraussetzungen

EN: Inherited prerequisites — siehe Graph-Definition

Zusätzliche Voraussetzungen, die aus den `requires`-Verweisen der `contains`-Vorfahren eines Ziels folgen und nicht bereits direkt am Ziel selbst stehen: $R_{\text{inherited}} = R_{\text{eff}} \setminus R$. Bei mehreren Elternknoten tragen alle Vorfahrenpfade bei. Vererbung entlang `contains` ist begrifflich strikt von mehrstufigen Voraussetzungsketten zu unterscheiden.

Effektive Voraussetzungen

EN: Effective prerequisites — siehe Graph-Definition

Die abgeleitete Voraussetzungsrelation $R_{\text{eff}} = R \cup R_{\text{inherited}}$ aus eigenen `requires`-Kanten und von `contains`-Vorfahren geerbten Voraussetzungen, auf atomare Basisziele expandiert (R_{at}). Im vollständig atomar modellierten Zielzustand fallen modellierte und effektive Voraussetzungen auf Atomebene zusammen.

Erreichbarkeit von Voraussetzungen

EN: Prerequisite reachability — siehe Graph-Definition

Ziele, die über eine oder mehrere Kanten der jeweiligen Voraussetzungsrelation verbunden sind, geschrieben R^+ bzw. R^+_{at}. Diese transitive Hülle umfasst Einzelkanten wie längere Ketten. Eine Lernfluss-Darstellung kehrt die Pfeilrichtung um und kennzeichnet dies ausdrücklich.

Gewicht (`weight`)

EN: Weight — siehe Graph-Definition und AGENTS.md

Eine strikt positive Zahl für den Anteil eines Lernziels am Fortschritt. Konvention in den kanonischen Landschaften: Atomare Ziele tragen `weight = 1`, Cluster die Anzahl ihrer *eindeutigen* atomaren Nachfahren, sodass mehrfach eingehängte Ziele nur einfach zählen.

Lernziel-ID und `shortKey`

EN: Goal ID and `shortKey` — siehe Graph-Definition

`id` ist die kanonische UUID eines Lernziels; sie darf Lernenden und Coaches angezeigt werden und ist kein geheimer Zugriffsschlüssel. `shortKey` ist ein optionaler stabiler ASCII-Bezeichner für Exporte, APIs und schichtenübergreifende Referenzen; er ist je logischer Landschaft eindeutig und ersetzt niemals `id`.

Graphgültigkeit

EN: Graph validity — siehe Graph-Definition

Ein Graph ist gültig genau dann, wenn alle IDs eindeutig sind, `contains`, `requires` und die effektive atomare Voraussetzungsrelation azyklisch sind und beide Minimalitätsregeln gelten:

- **Lokale Minimalität:** Eine bereits von einem `contains`-Vorfahren geerbte Voraussetzung darf am Kindknoten nicht erneut notiert werden.
- **Transitive Minimalität:** Eine Kante darf nicht gesetzt sein, wenn dieselbe Voraussetzung bereits über andere effektive Voraussetzungspfade folgt.

Motivationsanker

EN: Motivation anchor — siehe Context & Views

Ein atomares Orientierungsziel, das eine didaktische Route eröffnet und die Sinnfrage eines Fachs oder Themenbereichs aufschließt, etwa `Warum Physik? - Weltverständnis & Zukunft`. Routenprüfungen messen die Rückanbindung an solche Anker.

Terminales Autonomieziel

EN: Terminal autonomy goal — siehe Context & Views

Ein atomares Ziel, an dem Lernende selbstständig Probleme lösen – typischerweise eine Prüfungs- oder Übungsaufgabe wie die Einzelaufgaben unter `Übungen Q2`. Es bildet den Abschluss einer didaktischen Route.

Routenabdeckung

EN: Route coverage — siehe Context & Views

Die Eigenschaft, dass jedes routenrelevante atomare Lernziel auf mindestens einem Voraussetzungspfad zwischen einem Motivationsanker und einem terminalen Autonomieziel liegt. Die intendierte Lernrichtung verläuft von Orientierung über Verstehen (und begleitendes Memorieren bei Bedarf) zur selbstständigen Anwendung.

3 Knotentypen (Lernmodi)

Die pädagogischen Zielarten stehen orthogonal zur strukturellen Unterscheidung zwischen atomaren Zielen und Clustern. Siehe Node Types.

Verstehen (`Understanding`)

EN: Understanding

Der Standardtyp: begriffliche oder prozedurale Kompetenz, die im Lerndialog erarbeitet und geübt wird. Ein Ziel ohne spezielle Prüfungs- oder Memorierfelder ist ein Verständnisziel; Orientierungsziele sind gesondert gekennzeichnet und prüfen kein Fachwissen.

Sich merken (`Memorize`)

EN: Memorize

Ein Spaced-Repetition-Ziel für Fakten, Vokabeln, Notation oder ausgewählte Formeln. Es gilt erst dann als gemeistert, wenn keine Pflichtkarte fällig ist **und** jede Pflichtkarte einen ungeholfenen Abruftest (*Verified Recall*) bestanden hat.

Selbstständig Probleme lösen / Prüfungsknoten (`Exam`)

EN: Exam

Ein Ziel mit `examData`: eine eigenständig zu bearbeitende Prüfungs- oder Übungsaufgabe samt Musterlösung und Bewertungsschema. `requires` steuert die Erreichbarkeit und ermöglicht gezieltes Nacharbeiten nach Fehlversuchen.

`examData` und Bewertung

EN: `examData` and scoring — siehe Node Types

Aufgabentext, Musterlösung und ein Bewertungsobjekt mit `maxPoints`, `passingPoints` und Teilpunkten je Lösungsschritt: total $=$ $\min(\sum \text{Teilpunkte}, \text{maxPoints})$, bestanden genau für total $\geq$ passingPoints. Nur das Bestehen schreibt Mastery (`1.0`); ein nicht bestandener Versuch wird im Dialog erläutert, verändert aber den gespeicherten Lernzustand nicht, sodass die Aufgabe erneut bearbeitet werden kann.

Prüfungsmodus (`Exam Mode`)

EN: Exam Mode — siehe Node Types

Der Laufzeitmodus, in dem der KI-Coach als neutrale Prüfungsaufsicht agiert: Die Aufgabe wird unverändert gestellt, während der Bearbeitung werden keine inhaltlichen Hinweise gegeben, eine vollständige Abgabe wird Schritt für Schritt bewertet, und erst danach kehrt die begleitende Coach-Rolle zurück.

Kartei und Wiederholungsplanung (`SRS / SM-2`)

EN: SRS and SM-2 — siehe Node Types

Zeitgesteuerte Wiederholungsplanung nach dem SuperMemo-2-Algorithmus. Jede Karte führt `interval`, `repetition`, Leichtigkeit `ef` und `nextReview`. Eine Karte ist **fällig**, sobald `next Review <= now` gilt oder noch kein Kartenstatus vorliegt.

Geprüfter Abruf (`Verified recall`)

EN: Verified recall — siehe Node Types

Das verbindliche Abschluss-Gate für Memorierkarten: Der Coach zeigt ausschließlich die Vorderseite, die lernende Person antwortet ohne Hilfestellung, und erst danach ruft der Coach die erwartete Antwort ab und speichert `passed` oder `failed`. Reines Karteikartenüben im Cockpit ersetzt diesen Nachweis nicht.

4 Strukturschichten und Projektion

Schichtenmodell

EN: Layer model — siehe General Goal System and Migration

SkillPilot trennt fünf Ebenen, die in älteren Bestandsgraphen teilweise noch vermischt waren:

Schicht	Inhalt	Beispiel
Zielschicht (*Goal layer*)	überprüfbare Kompetenzen und fachliche Cluster	„lineare Funktionen verknüpfen“
Programmschicht (*Program layer*)	strukturelle Einheiten	`Jahrgang 8`, `E-Phase`, Modul, Kursprofil, Prüfung
Platzierungsschicht (*Placement layer*)	Zuordnungen von Zielen zu Programmeinheiten	„dieses Ziel gehört in Q1“
Kompetenzachsen (*Competency-axis layer*)	Taxonomieeinträge	`K1 Mathematisch argumentieren`
Kompositionssicht (*Composition-view layer*)	kontextspezifische Lernbäume	`DE-BY / Gymnasium / SekII / Mathematik / LK`

Programmeinheit

EN: Program unit — siehe View Projection and Goal Placement

Ein expliziter Strukturcontainer (Jahrgang, Halbjahr, Modul, Phase, Kursprofil, Prüfungsabschnitt), der über `parentUnitId` und `order` einen geordneten Baum bildet. Ein fachliches Lernziel darf nicht als Ersatz für eine Programmeinheit zweckentfremdet werden.

Lerneinheit (`LearningUnit`)

EN: Learning unit — siehe Lerneinheiten und Optionale Planung (§12)

Ein rekursiv aufgebautes pädagogisches Ganzes mit explizitem Lernversprechen, motivierendem Einstieg, zugehörigen Zielen oder Untereinheiten und geprüfter Abschlussregel. Ein Jahrgang, Halbjahr oder bloßer Navigationscluster wird nicht allein durch Gruppierung zur Lerneinheit.

Jedes Ziel besitzt in der aufgelösten Einheitenhierarchie genau einen direkten Erwerbseigentümer; übergeordnete Einheiten aggregieren ihre Nachfahren. Der Abschluss folgt aus Pflichtkindern und eigenen Zielpflichten ohne gesondert gespeichertes Einheits-Mastery-Ergebnis. Eine offene Elternprüfung öffnet bereits erreichte Kindkompetenzen nicht wieder, und ein bestandener Elterntest ersetzt niemals geschützte Kindprüfungen oder Memorierkarten. Wirksame Kompetenz bleibt global, und einzelne Anerkennungen können dem vollständigen Einheitsabschluss vorausgehen; tragfähig ist eine solche Anerkennung nur in Einheiten, die die Gültigkeitsregeln R1–R5 vom 11. November 2026 erfüllen (Graph-Definition §13.3). Dieses weiterentwickelte Konzept befindet sich noch in der Umsetzung; siehe auch Einheitsfolge.

Zielplatzierung (`Goal placement`)

EN: Goal placement — siehe View Projection and Goal Placement

Ein Eintrag, der ein unverändertes Lernziel unter einem geprüften Kontext an eine Programmeinheit anbindet (`primary`, `secondary`, `assessed`). Eine Platzierung hängt den fachlichen `contains`-Baum niemals um und erzeugt kein doppeltes Vorkommen im Standardbaum.

Kompetenzachse

EN: Competency axis — siehe View Projection and Goal Placement

Eine orthogonale Taxonomiesicht aus `competencyCatalog` und `competencyRefs`, etwa Prozesskompetenzen `K1`–`K6`. Solche übergreifenden Kompetenzfamilien sind Taxonomieeinträge, weder Lernziele noch Programmeinheiten.

Kompositionssicht (`Composition view`)

EN: Composition view — siehe View Projection and Goal Placement

Eine explizite, geprüfte Definition des lernendenseitigen oberen Baums für genau einen aufgelösten Kontext: Strukturknoten, Bezeichnungen, Reihenfolge und Referenzen auf kanonische Teilbaumwurzeln. Sie enthält weder neu verfasste atomare Ziele noch neue `requires`-Kanten und ist – wo vorhanden – die maßgebliche Quelle für den Standardbaum dieses Kontexts.

Kanonische Teilbaumreferenz

EN: Canonical subtree reference — siehe View Projection and Goal Placement

Ein Verweis aus einer Kompositionssicht auf eine geprüfte Cluster-Wurzel des kanonischen Graphen; die Inhalte bleiben im kanonischen Modell. Innerhalb eines kompilierten Sichtbaums müssen expandierte Teilbaumreferenzen paarweise disjunkt sein.

Projektionsrolle (`projectionRole`)

EN: Projection role — siehe View Projection and Goal Placement

Pro Referenz in einer Kompositionssicht deklarierte Rolle (Standardwert `target`):

- `target` — eigenes Lernziel dieses Umfangs; zählt für Navigation, Frontier, Fortschritt und Abschluss.
- `prerequisiteOnly` — reine Voraussetzung, deren globaler Lernstand Ziele freischalten oder blockieren kann, die hier aber nicht selbst als Ziel ausgewählt wird.

Rollen werden niemals implizit aus `requires`, `phase`, Jahrgang oder Schulstufe abgeleitet. Bei Überlappung entscheidet die Spezifität (direkter `goalEntry` vor Teilbaumrolle, tieferer Teilbaum vor breiterem; erst bei gleicher Spezifität gewinnt `target`).

Gültigkeit / Anwendbarkeit (`Applicability`)

EN: Applicability — siehe Node Types und Context & Views

Optionale kompilierte Zielmetadaten, die Filterdimensionen (`jurisdiction`, `schoolForm`, `stage`, `durationModel`, `courseProfile`, …) auf die zulässigen Werte abbilden. Eine nicht gesetzte Dimension gilt als unbeschränkt. `applicability` ist kein Knotentyp und erzeugt niemals Elternkanten; sie blendet Ziele in gefilterten Sichten lediglich ein oder aus.

Gültigkeits-Override (`Applicability override`)

EN: Applicability override — siehe Context & Views

Prüfmetadaten unter `extendedData.applicabilityOverrides`, die dokumentieren, welche aktiven Gültigkeitswerte durch eine ausdrücklich begründete Ausnahme statt durch gewöhnliche Quellenableitung ergänzt wurden. Sie dienen dem Audit; zur Laufzeit zählt weiterhin die kompilierte `applicability`.

Einstiegskontext (`Entry scope`)

EN: Entry scope — siehe General Goal System and Migration

Die kleine Menge kontextueller Eingrenzungen vor der Inhaltsnavigation: Schulform, Stufe, Bundesland, Dauermodell, Kursprofil, aktueller Jahrgang oder Phase. Der Einstiegskontext wirkt als Abfrage- und Navigationsfilter und erzwingt niemals eine Duplizierung von Lernzielen.

Aufgelöster Kontext (`Resolved scope`)

EN: Resolved scope — siehe View Projection and Goal Placement

Eine vollständig bestimmte Kombination wie `DE-HE / Gymnasium / SekII / Mathematik / LK`. Standardbäume, Einmalvorkommen und Abdeckungsprüfungen beziehen sich stets auf genau einen aufgelösten Kontext.

Standardbaum (`Default tree`)

EN: Default tree — siehe View Projection and Goal Placement

Der eindeutige Baum, den Lernende für einen aufgelösten Kontext sehen. Jede sichtbare Kante entstammt genau einer expliziten Quelle (`contains`, `parentUnitId`, `goalPlacements` oder `competencyRefs`), niemals einer Heuristik über Titel oder `phase`. **Einmalvorkommen:** In einem aufgelösten Kontext erscheint jedes sichtbare Ziel höchstens einmal und unter höchstens einem sichtbaren Elternknoten.

Inhaltsbaum, Programmbaum, Kompetenzsicht

EN: Content tree, program tree, competency view — siehe View Projection and Goal Placement

Die drei klassischen Sichtfamilien mit jeweils eindeutiger Strukturquelle (`contains` für Inhalt, `programUnits` + `goalPlacements` für Programm, `competencyCatalog` + `competencyRefs` für Kompetenzachsen), ergänzt um die kontextspezifische Kompositionssicht als bevorzugte lernendenseitige Standardsicht.

`phase`

EN: `phase` — siehe Graph-Definition

Kompatibilitätsmetadaten (`E`, `Q1`–`Q4`, …) für Badges, Filter oder Migrationswerkzeuge. `phase` ist ausdrücklich **keine** kanonische Graphsemantik und begründet niemals eine Elternschaft.

`ALL`-Platzhalter

EN: `ALL` — siehe Context & Views

Ein reiner Abfrage-Platzhalter mit der Bedeutung „diese Dimension bei der Kontextauflösung nicht einschränken“. `ALL` darf niemals als gespeicherter Gültigkeits- oder Platzierungswert serialisiert werden.

5 Lernlaufzeit

Personalisierungsstufen (`Levels of personalization`)

EN: Levels of personalization — siehe Levels of Personalization

Die vierstufige Eingrenzung vom Curriculum bis zum individuellen Lernstand: **1** Basiscurriculum → **2** bestätigtes persönliches Curriculum → **3a** aktueller Fokus plus **3b** genau ein aktives atomares Lernziel → **4** globale Mastery. Katalognavigation wie Schule, Hochschule oder Sprachen liegt vor Level 1 und ist kein Lernumfang.

Basiscurriculum (`Base curriculum`)

EN: Base curriculum — siehe Learning Workflow

Die vor Lernbeginn gewählte Skill-Landschaft; sie spannt das gesamte Zieluniversum für diese lernende Person auf.

Persönliches Curriculum (`Personal curriculum`)

EN: Personal curriculum — siehe Learning Workflow

Der bestätigte, längerfristig gültige Ausschnitt des Basiscurriculums für eine lernende Person (Bundesland oder kanonische Sicht, Dauermodell, Schulstufe, Fächer und Kursprofile je Fach). Er bestimmt die lernendenseitige Kompositionssicht und bildet den Kandidatenraum für Fokus, Frontier, Fortschritt und Abschluss, während das Backend bei strikten Prüfungen weiterhin den Gesamtgraphen heranziehen kann.

Dieser Zustand gehört verbindlich dem Backend und wird ausschließlich über die SkillPilot-Weboberfläche bzw. das Cockpit bearbeitet – der Chat-Coach fragt ihn weder ab noch verändert er ihn.

Personalisierung (`Personalization`)

EN: Personalization — siehe Verhaltensintegration des MCP-Lerncoaches

Die von der lernenden Person gesetzten Gültigkeitsentscheidungen wie Bundesland oder Kursprofil. Personalisierung wird niemals aus `contains` oder `requires` erraten. Auswählbare Optionen kommen stets als opake IDs aus dem aktuellsten Backendzustand. Kursprofile gelten je Fach, sodass Mathematik LK und Physik GK gleichzeitig geführt werden können.

Lernumfang und Fokus (`Learning scope and focus`)

EN: Learning scope and focus — siehe Verhaltensintegration des MCP-Lerncoaches

Zwei Ebenen, die der Coach sauber trennen muss:

- **Lernumfang (*Learning scope*):** das bestätigte persönliche Curriculum dieser lernenden Person.

- **Fokus (*Level 3a*):** der aktuell ausgewählte Teilbaum oder Lernkorridor (`plannedGoals`), aus dem Frontier- und Zieloptionen stammen. Sobald das persönliche Curriculum vollständig festgelegt ist, bleibt der Fokus niemals leer: Ohne gültige explizite Auswahl greift deterministisch die oberste erste sichtbare Wurzel. Ein Jahrgang oder eine Phase kann den Fokus begrenzen, ohne selbst ein Lernziel zu sein.

Frontier und Zielempfehlungen werden erst angeboten, wenn Umfang und Fokus aufgelöst sind.

Aktives Ziel (`Active goal`)

EN: Active goal — siehe Learning Workflow

Genau ein vom Backend bestätigtes atomares Lernziel (**Level 3b**), das aktuell in Oberfläche und Coach bearbeitet wird. Im Regelfall entstammt es der Frontier; bei einer Auswahl außerhalb der Frontier weist das System die fehlenden Voraussetzungen aus, statt sie stillschweigend zu übergehen.

Frontier

EN: Frontier — siehe Graph-Definition

Die Menge der noch nicht gemeisterten **atomaren** Lernziele, deren effektive Voraussetzungen sämtlich erfüllt sind – die didaktische Zone der nächsten Entwicklung, deterministisch aus dem Graphen berechnet statt durch die KI geschätzt.

Erfüllung (`Satisfaction`)

EN: Satisfaction — siehe Graph-Definition

$\mathrm{Sat}(g)$ gilt genau dann, wenn die atomare Basis von g nichtleer und vollständig gemeistert ist. Ein atomares Lernziel ist genau dann erfüllt, wenn es gemeistert ist; ein Cluster genau dann, wenn alle seine atomaren Nachfahren gemeistert sind.

Optimistischer Modus (`Optimistic mode`)

EN: Optimistic mode — siehe Context & Views

Erst auf den gewählten Sichtumfang filtern (einschließlich expliziter `prerequisiteOnly`-Grundlagen), dann Voraussetzungen prüfen. Lernende können direkt im gewählten Umfang starten, ohne pauschal durch ältere Jahrgänge blockiert zu werden. Dies ist der Standardmodus.

Strikter / pessimistischer Modus (`Strict mode`)

EN: Strict mode (pessimistic mode) — siehe Context & Views

Kandidaten stammen weiterhin aus dem gefilterten Umfang, Voraussetzungen werden jedoch global über den gesamten Graphen geprüft. Dient der gezielten Diagnose fehlender Grundlagen außerhalb des aktuellen Filters (`strictMode`).

Fehlende Voraussetzungen (`Missing prerequisites`)

EN: Missing prerequisites — siehe Context & Views

Die noch unerfüllten effektiven Voraussetzungen eines Ziels, getrennt nach Lücken innerhalb und außerhalb des aktiven Filters – die präzise Antwort auf „Warum ist dieses Ziel noch gesperrt?“.

Beherrschung (`Mastery`)

EN: Mastery — siehe Learning Workflow und Implizite Beherrschung

Das einzige gespeicherte Ergebnis für ein atomares Lernziel: **gemeistert oder noch nicht gemeistert** (kodierbar als `0` und `1`, ohne Zwischennoten oder zweite Qualifikationsstufe). Mastery hängt an stabilen Lernziel-IDs und gilt global über alle Kompositionssichten hinweg. Cluster- oder Fachfortschritt ist der gewichtete Anteil eindeutiger gemeisterter atomarer Ziele.

Das verbindliche Konzept vom 4. November 2026 ergänzt die **effektive Beherrschung** $M_{\text{eff}} = M \cup I$: Dabei bezeichnet I die abgeleitete Anerkennung gewöhnlicher Kompetenzen durch selbstständige Anwendung in gültigen lokalen Prüfungsverträgen. Prüfungs- und Memorierziele gelangen niemals nach I. Die Laufzeitanpassung an diesen erweiterten Vertrag steht noch aus.

Curriculum-Zeitachse und Stoffverteilungsplan

EN: Curriculum timeline and pacing plan — siehe Curriculum-Zeitachse, Pufferplanung und Soll-Ist-Lerntempo

Ein versioniertes Ausführungsplan-Overlay, das aufgelöste Curriculum-Cluster konkreten lokalen Zeiträumen und Unterrichtskapazitäten einer Klasse oder lernenden Person zuordnet. Es verwaltet Kalender, Puffer, Planblöcke und Revisionen, schreibt aber niemals Termine in kanonische Ziele, Programmeinheiten, Platzierungen oder Kompositionssichten.

Planungseinheit (`Planning unit`)

EN: Planning unit — siehe Curriculum-Zeitachse, Pufferplanung und Soll-Ist-Lerntempo

Eine planlokale Einheit zur Verteilung des erwarteten Lernfortschritts auf die verfügbare Zeitkapazität (transparenter Grundwert: eine Einheit je einplanbarem atomarem Ziel). Sie ist weder eine Note noch das kanonische `weight`; Vorwissen vor Planbeginn zählt nicht als neu gelernte Planungseinheit.

Planmeilenstein (`Plan milestone`)

EN: Plan milestone — siehe Curriculum-Zeitachse, Pufferplanung und Soll-Ist-Lerntempo

Eine datierte Zielzusage wie „ausgewählte Mathematik-Abituraufgaben bis 15. März abschließen“. Sie trennt fachliche Art von Terminhärte sowie eingeplanten Arbeitsumfang von erreichtem Mastery-Abschluss, steuert datierte Zielliste, Prognose und Umplanung, vergibt aber selbst keine Mastery.

Kurs-Planlage (`Course plan status snapshot`)

EN: Course plan status snapshot — siehe Curriculum-Zeitachse, Pufferplanung und Soll-Ist-Lerntempo

Ein prüfsummengebundenes Stichtags-Lesemodell einer Planrevision mit textuellem Kursplan, geplantem Blockfortschritt, bestätigter Unterrichtsabdeckung und gesondert ausgewiesener Mastery-Verteilung samt Eingangsdatenqualität. Es erzeugt keine eigene Datenquelle neben Plan und Mastery.

Lerngeschwindigkeit (`Learning velocity`)

EN: Learning velocity — siehe Learning Velocity

Die beobachtete Rate abgeschlossener atomarer Lernziele pro Zeitraum. Das Cockpit-Diagramm zählt dokumentierte Abschlüsse je Kalenderwoche; ein gebundener Zeitplan vergleicht neu erreichte Planungseinheiten je effektiver Unterrichtswoche mit seiner Sollkurve. Beide Größen sind Verlaufsindikatoren, weder Noten noch Ranglisten.

Autopilot

EN: Autopilot — siehe Frontier Goal Selection Strategies & Autopilot

Ein optionaler Modus, der nach dem Meistern des aktiven Ziels automatisch das erste Ziel der gemäß Auswahlstrategie geordneten nächsten Schritte aktiviert.

Auswahlpriorisierung (`Selection strategy`)

EN: Selection strategy — siehe Frontier Goal Selection Strategies & Autopilot

Die Sortierregel über der berechneten Frontier: `Random` für Abwechslung über Themenbereiche hinweg, `Sequential` für ein lineares Vorgehen entlang der Curriculumstruktur. Gespeichert je Profil zusammen mit Autopilot und `strictMode`.

6 Coach, Session und Identität

SkillPilot-ID

EN: SkillPilot ID — siehe Schnellstart

Der dauerhafte, pseudonyme Zugriffsschlüssel einer lernenden Person zu ihrem Lernprofil – ohne Registrierung mit Klarnamen oder E-Mail-Adresse. Die permanente SkillPilot-ID verbleibt strikt im Backend und erscheint niemals als Tool-Parameter, Tool-Ergebnis oder Chatinhalt.

Lernsession (`Learning session`)

EN: Learning session — siehe OAuth-Appbindung und 24h-Lernsession

Ein opaker, zufällig erzeugter Schlüssel mit absoluter Gültigkeit von 24 Stunden, der genau den über **Lernen starten** bestätigten Lernkontext adressiert. Jeder Start erzeugt eine frische Session; der Adapter übergibt sie unverändert bei jedem lernendenbezogenen Werkzeugaufruf.

OAuth-Appbindung (`OAuth app binding`)

EN: OAuth app binding — siehe OpenAI MCP Client Binding

Die widerrufbare OAuth-Autorisierung, mit der sich die zugelassene Provider-Integration gegenüber dem Backend authentisiert. Sie weist die **Integration** aus, nicht die lernende Person: OAuth allein wählt kein Lernprofil aus und erzeugt keine Lernsession.

SkillPilot Lerncoach

EN: SkillPilot Lerncoach — siehe Kommunikationsvertrag ChatClient/Backend

Die KI-seitige Dialogschicht, die den Backendzustand in ein Lerngespräch übersetzt. Der Coach führt das Gespräch; das Backend entscheidet verbindlich über Lernzustand, aktive Filter, zulässige Übergänge und erreichbare nächste Schritte.

Cockpit

EN: Cockpit — siehe Schnellstart

Die SkillPilot-Weboberfläche für Lernlandkarte, Fortschritt, Lernplanstand und SRS-Karteikartentraining. Cockpit und Coach arbeiten auf demselben Backend-Lernzustand; den ungeholfenen Abruftest (*Verified Recall*) bewertet nicht das Cockpit, sondern der Coach-Ablauf.

Sichere Zustandsprojektion (`Safe state projection`)

EN: Safe state projection — siehe Kommunikationsvertrag ChatClient/Backend

Der Grundsatz, dass ausschließlich eine explizit freigegebene, datensparsame Sicht des Lernzustands die Providergrenze überschreitet. Interne Identitäten, Geheimnisse und private Dateipfade werden vor jeder Übermittlung an Modell, Chat oder Widget entfernt.

Providerneutrale Grenze (`Provider-neutral boundary`)

EN: Provider-neutral boundary — siehe Kommunikationsvertrag ChatClient/Backend

Die strikte Trennung zwischen gemeinsamem fachlichem SkillPilot-Kern und providerspezifischen Adaptern. Domänenregeln, Prüfungsautorisierung und Zustandsprojektion liegen im gemeinsamen Kern, sodass ein neuer oder geänderter KI-Anbieter niemals die Fachlogik aufspaltet.

7 Qualität, Review und Governance

Curriculum Champion

EN: Curriculum Champion — siehe Champion Guide

Eine fachkundige Person, die Verantwortung für die praktische Erprobung eines Curriculums oder eines klar abgegrenzten Themen-Scopes übernimmt: Ziele selbst durchlernen, Stolperstellen am betreffenden Lernziel oder über GitHub zurückmelden und die fachliche Weiterentwicklung begleiten.

Vier-Stufen-Champion-Modell

EN: Four-level champion model — siehe Four-Level Champion Model

Die QS-Kaskade zwischen maschineller Prüfung und menschlicher Erprobung: `1a` automatisierte CI-Pipeline (aktiv), `1b` statischer KI-Review-Agent und `1c` dynamischer KI-Agent sowie `2` die unabhängige menschliche Champion-Erprobung.

QA-Scope

EN: QA scope — siehe Curriculum-Qualität

Ein explizit konfiguriertes Prüfprofil für einen lernendenseitigen Curriculumabschnitt, bestimmt durch Curriculum, Bezeichnung (z. B. `Sekundarstufe I`), Motivationsanker, ausgewählte atomare Ziele, terminale Autonomiebereiche und Cluster-Selektor. Ein QA-Scope zählt registrierte Prüfprofile – weder Einzelpfade im Graphen noch Aufgabenanzahlen. Die frühere Bezeichnung „Routen“ wurde abgelöst.

CQR-Regel

EN: CQR rule — siehe Curriculum-Qualität

Eine *Curriculum-Quality-Review*-Regel mit stabiler ID im Qualitäts-Dashboard, z. B. `CQR-003` für bidirektionale Bundesland-Abdeckung oder `CQR-101` für geschlossene Lernrouten. CQR-Regeln bewerten die inhaltliche Curriculum-Qualität.

GVR-Regel

EN: GVR rule — siehe Graph Validation Rules

Eine *Graph-Validation-Rollout*-Regel mit stabiler ID im CI-Validator, z. B. `GVR-001` (kein Ziel darf seinen eigenen `contains`-Vorfahren voraussetzen). GVR-Regeln sichern die strukturelle Graphintegrität ab.

Reifegrad (`Maturity level`)

EN: Maturity level — siehe Curriculum-Qualität und Qualitäts- und Erprobungskonzept

Eine kumulative, konservative Einstufung (`M0`–`M7`) für einen exakt benannten Umfang und Inhaltsstand. Ein einzelner QA-Scope erreicht höchstens `M3`; ein Curriculum kann `M0` bis `M7` erreichen. `M5` steht für den abgeschlossenen Kern-QS-Stand (Quellen erfasst, Ländersichten bidirektional belegt, Pflicht-QA-Scopes routen- und prüfungsvollständig). `M6` ergänzt die geprüfte Memory-Schicht; `M7` ist der vollständig abgeschlossene maschinelle QS-Stand über alle fünf Einzelziel-Gates (`D` Beschreibungen, `P` Verständnisprofile, `A` Atomizität, `M` Memory, `V` Visualisierungen).

Menschliche Erprobung durch Curriculum Champions wird als eigenes Merkmal separat ausgewiesen (kein `M8`); `M7` allein bescheinigt keine menschliche Freigabe oder Unterrichtswirksamkeit.

Semantische Atomarität (`Semantic atomicity`)

EN: Semantic atomicity — siehe Semantic Atomicity Review

Die fachliche Prüffrage, ob ein Lernziel tatsächlich genau eine überprüfbare Kompetenz beschreibt statt mehrere Fähigkeiten zu bündeln. Strukturelle Atomarität (keine `contains`-Kinder) ist dafür notwendig, aber nicht hinreichend.

Generiertes Artefakt (`Generated artifact`)

EN: Generated artifact — siehe Documentation Guidelines

Ein durch ein Skript erzeugter Bericht, Status-Snapshot oder Dokumentations-Build mit explizitem Hinweis auf Generator, Regenerierbefehl und maßgebliche Quelle (*Source of truth*). Inhaltliche Entscheidungen gehören stets in die Quelldateien, niemals als manuelle Nachbearbeitung in das generierte Artefakt.

8 Wichtige Abgrenzungen auf einen Blick

Nicht dasselbe	Warum die Trennung zählt
`contains` vs. `requires`	`contains` bündelt Inhalte hierarchisch; `requires` ordnet Lernvoraussetzungen. Nur `requires` steuert die Frontier.
`contains` vs. `goalPlacement`	Fachliche Zusammensetzung vs. Verortung im Programmbaum. Eine Platzierung hängt niemals den Inhaltsbaum um.
`requires` vs. effektive Voraussetzungen	`requires` speichert direkte Kanten; effektive Voraussetzungen umfassen während der Migration auch von Vorfahren geerbte Voraussetzungen.
Atomares Lernziel vs. Cluster-Ziel	Mastery, Frontier und Abdeckungszählungen sind auf atomaren Zielen definiert; Cluster aggregieren ausschließlich.
Strukturelle vs. semantische Atomarität	Ein Blattknoten ohne `contains`-Kinder kann inhaltlich trotzdem mehrere Teilkompetenzen vermischen.
`applicability` vs. `projection Role`	`applicability` regelt, *ob* ein Ziel im Kontext sichtbar ist; `projectionRole` regelt, ob es eigenes Ziel (`target`) oder reine Voraussetzung (`prerequisiteOnly`) ist.
`applicability` vs. Knotentyp	`applicability` ist kontextuelle Sichtmetadaten, kein vierter Lernmodus.

Nicht dasselbe	Warum die Trennung zählt
Einstiegskontext vs. persönliches Curriculum	Der Einstiegskontext filtert eine Abfrage; das persönliche Curriculum ist der dauerhaft bestätigte Lernumfang.
Lernumfang vs. Fokus	Der Umfang legt fest, was insgesamt zählt; der Fokus wählt den aktuellen Arbeitskorridor.
Frontier vs. Empfehlung	Die Frontier ist mathematisch aus Graph und Lernstand bestimmt; Auswahlstrategie und Autopilot ordnen nur innerhalb der Frontier.
Optimistischer vs. strikter Modus	Gleiche Zielkandidaten, unterschiedliche Voraussetzungstiefe – strikter Modus ist Diagnose, kein strengerer Lehrplan.
SRS-Üben vs. Verified Recall	Karteikartenüben plant Wiederholungen; erst der ungeholfene Abruftest im Coach schließt ein Memorierziel ab.
SkillPilot-ID vs. Lernsession	Dauerhafte pseudonyme Backend-Identität vs. frischer 24-Stunden-Schlüssel für die Provider-Anbindung.
OAuth-Bindung vs. Lernsession	OAuth authentisiert die App/Integration; die Lernsession adressiert den Lernkontext.
`phase` vs. Programmeinheit	`phase` ist Kompatibilitätsmetadaten; nur explizite Programmeinheiten bilden Strukturknoten.
Lerneinheit vs. Personalisierungsstufe	Rekursive pädagogische Einheit mit abgeleitetem Abschluss vs. eine der vier Umfangs-/Zustandsebenen.
Lerneinheit vs. Zielschwierigkeit	Erwerbs- und Abschlusseinheit vs. Anspruchsniveau eines einzelnen Ziels (`level`).
Einheitsabschluss vs. Zielverfügbarkeit	Erfüllte Kind- und Eigenpflichten vs. Startvoraussetzungen; ein technisches Ganz-Einheiten-Gate erfordert eine separate Policy.
QA-Scope vs. Lernroute	Ein QA-Scope ist ein registriertes Prüfprofil, kein einzelner Lernpfad.
CQR- vs. GVR-Regeln	Inhaltliche Curriculum-Qualitätssicherung vs. strukturelle Graphvalidierung.
Curriculum vs. Skill-Graph	Normatives externes Dokument vs. versioniertes operatives Graphmodell.
Skill-Graph vs. Skill-Landschaft	Formales Modell und Semantik vs. konkrete veröffentlichte Fachinstanz mit `landscapeId`, Champion und Reifegrad.

Pflegehinweise

- `cd app && npm run check:terminology` weist abgelöste Synonyme zurück; bei Konsolidierung eines Begriffs wird dort eine entsprechende Prüfregel ergänzt.
- Neue Begriffe gehören hierher, sobald sie dokumentübergreifend verwendet werden und ihre Bedeutung nicht unmittelbar aus dem Code hervorgeht.
- Jeder Eintrag beschränkt sich auf Definition und knappe Abgrenzung; Einzelheiten bleiben im verlinkten normativen Fachdokument.
- Bei Abweichungen zwischen diesem Glossar und einem verlinkten Fachdokument ist das Fachdokument maßgeblich und dieser Eintrag nachzuziehen.