

# Technische Spezifikation ALIS (Deep Dive Masterplan)

Dieses Dokument definiert die granularen Workflows, LLM-Schnittstellen und Datenstrukturen für das Adaptive Lern-App-System (ALIS).

## 1. Detaillierte Datenstrukturen (Firestore-Schema)

Alle Daten werden unter `/artifacts/{appId}/users/{userId}/...` gespeichert.

### 1.1 `config/profile` (Nutzerprofil)

Enthält aggregierte und persistente Lernpräferenzen.

```
{
  "userId": "unique-user-id-string",
  "stylePreference": "Analogien-basiert", // Aus P7 abgeleitet
  "complexityLevel": "Mittel", // Aus P7 abgeleitet
  "paceWPM": 180, // Wörter pro Minute, dynamisch aus P5 berechnet
  "P2Enabled": true, // Boolean: Vorwissenstest aktiv?
  "P6Enabled": true, // Boolean: Verständnisprüfung aktiv?
  "lastActiveGoalId": "G-20251127-12345"
}
```

### 1.2 `goals/{goalId}` (Lernziel-Vertrag)

Der von **Architekt (P1)** erstellte SMART-Vertrag.

```
{
  "goalId": "G-20251127-12345",
  "name": "Multi-Agenten-Systeme in der Logistik implementieren",
  "fachgebiet": "Künstliche Intelligenz",
  "targetDate": "2026-03-01",
  "bloomLevel": 5, // 5: Evaluieren
  "messMetrik": "90% korrekte Bewertung von 5 Fallstudien",
  "status": "In Arbeit" // In Arbeit, Abgeschlossen, Abgebrochen
}
```

### 1.3 `goals/{goalId}/path/structure` (Lernpfad-Struktur)

Die sequenzielle Gliederung, dynamisch änderbar (durch P5.5).

```
[
  {
    "id": "K1-Rekursion",
    "name": "Grundlagen der Rekursion (Eingefügte Lücke)",
    "status": "Offen", // Offen, Beherrscht, Übersprungen, Reaktiviert (durch P5.5)
    "estimatedTime": 25, // Minuten, aus P3/P7 berechnet
    "expertiseSource": "P5.5 Remediation", // P2 Vorwissen, P3 Experte, P5.5 Remediation
    "requiredBloomLevel": 3 // Das notwendige Bloom-Niveau für dieses Konzept
  },
  {
    "id": "K2-Agentenarchitektur",
    "name": "Agentenarchitekturen (Sense-Plan-Act)",
    "status": "Aktiv",
    "estimatedTime": 45,
    "expertiseSource": null,
  }
]
```

```

    "requiredBloomLevel": 4
  },
  // ... weitere Konzepte
]

```

#### 1.4 logs/{logId} (Lernprotokolle)

Detailliertes Log jeder Interaktion, essenziell für P7 und den Export.

```

{
  "timestamp": "2025-11-27T10:00:00Z",
  "eventType": "P5_Chat", // P1_Zielsetzung, P3_Skip, P4_Material, P5_Chat, P5.5_Lücke, P
  "conceptId": "K2-Agentenarchitektur",
  "textContent": "Da Sie nach einem einfacheren Beispiel fragten, hier die Analogie...",
  "emotionFeedback": "Frustration/Verwirrung", // Vom Tutor abgeleitet
  "testScore": null,
  "kognitiveDiskrepanz": null,
  "groundingSources": ["URL1", "URL2"]
}

```

## 2. Die LLM-Agenten und System Prompts

Die folgenden Prompts werden als `systemInstruction` in den Gemini API-Aufruf eingebettet.

### 2.1 Rolle 1: Der Architekt (P1, P3, P5.5)

**Aufgabe:** Strukturierung und Pfadverwaltung.

**\*\*System Prompt: ARCHITEKT\*\***

Du bist der **\*\*Architekt\*\*** im ALIS-System. Deine Aufgabe ist es, die Lernabsicht des Nutze

**\*\*Anweisungen:\*\***

- \*\*SMART-Standardisierung (P1):\*\*** Verhandle das Ziel, bis es messbar ist. Extrahiere `
- \*\*Pfad-Generierung (P3):\*\*** Erstelle eine Gliederung (5-10 Konzepte) in logischer Sequ
- \*\*Experten-Review (P3):\*\*** Präsentiere die Gliederung im Format der finalen Ausgabe. F
- \*\*Dynamische Korrektur (P5.5):\*\*** Wenn vom Tutor eine Lücke diagnostiziert wird, führe
  - \* Definiere das fehlende Konzept N1.
  - \* Füge N1 mit `status: Offen` und `expertiseSource: P5.5 Remediation` an die **\*\*Spitze**
  - \* Setze den Status des ursprünglich übersprungenen Konzepts (falls relevant) auf `Rea
- \*\*Output-Format:\*\*** Liefere das Ergebnis stets im folgenden Markdown-Format zurück.

### 2.2 Rolle 2: Der Kurator (P4, P6)

**Aufgabe:** Generierung von Material und Tests, Faktenprüfung.

**\*\*System Prompt: KURATOR\*\***

Du bist der **\*\*Kurator\*\*** im ALIS-System. Deine Aufgabe ist die Generierung von maßgeschnei

**\*\*Anweisungen:\*\***

- \*\*Kontext-Einbeziehung:\*\*** Nutze die Metriken aus dem `Nutzerprofil` (`stylePreference
- \*\*Inhaltsgenerierung (P4):\*\*** Generiere den Text. Der Stil muss zur Präferenz passen (
- \*\*Grounding:\*\*** Führe **\*\*Google Search\*\*** durch, um alle Fakten zu untermauern und die R
- \*\*Test-Generierung (P6):\*\*** Generiere 3-5 Testfragen (MC, Freitext, etc.), deren Frage
- \*\*Output-Format:\*\*** Liefere das Ergebnis stets im folgenden Markdown-Format zurück.

## 2.3 Rolle 3: Der Tutor (P5, P5.5, P7)

Aufgabe: Motivation, Diagnose, Ad-hoc-Hilfe.

**\*\*System Prompt: TUTOR\*\***

Du bist der **\*\*Tutor\*\*** im ALIS-System. Deine Priorität liegt in der **\*\*affektiven Steuerung**

**\*\*Anweisungen:\*\***

1. **\*\*Affektive Analyse (P7):\*\*** Bewerte jeden Nutzereintrag auf emotionale Indikatoren (F
2. **\*\*Ad-hoc-Hilfe (P5):\*\*** Reagiere auf Fragen, indem du die im 'Lernprotokoll' gespeiche
3. **\*\*Lücken-Diagnose (P5.5):\*\*** Wenn der Nutzer den Indikator **\*\*'Fundament fehlt'\*\*** auslö  
\* Wechsle in den Diagnosemodus.  
\* Frage den Nutzer: **\*\*Welches Schlüsselkonzept\*\*** fehlt Ihnen genau?"  
\* Sobald das Konzept identifiziert ist, **\*\*delegiere die Aufgabe, das neue Kapitel zu**
4. **\*\*Output-Format:\*\*** Dein Output ist **\*\*immer\*\*** ein natürlicher, konversationeller Chat-

## 3. Detaillierter Workflow mit Entscheidungspunkten

Der Prozess ist ein dynamischer Loop, der auf Triggern basiert.

### 3.1 P3: Pfad-Erstellung & Experten-Review

- **Trigger:** Abschluss der P1-Zielverhandlung.
- **LLM-Rolle:** Architekt.
- **Entscheidungspunkt: Nutzer-Input (Manuelle Skips):** Der Architekt generiert die Gliederung (Pfad). Das System präsentiert dem Nutzer die Gliederung.
  - **IF** Konzept von Nutzer als **'Bekannt'** markiert: Setze `status` auf `Übersprungen` und `expertiseSource` auf `P3 Experte` in der Datenbank.
  - **ELSE IF** Konzept durch P2 auf `Beherrscht`: Setze `status` auf `Beherrscht`.
  - **ELSE:** Setze `status` auf `Offen`.

### 3.2 P5.5: Dynamische Remediation (Der Lücken-Loop)

- **Trigger:** Nutzer klickt den Button "Fundament fehlt / Lücke melden" während P5.
- **LLM-Rollen:** Tutor (Diagnose) -> Architekt (Korrektur).
- **Workflow:**
  1. Tutor führt die **Lücken-Diagnose** durch (Konzept-Name N1 identifizieren).
  2. Tutor delegiert an den Architekten.
  3. **Entscheidungspunkt (Architekt-Logik):**
    - Architekt prüft, ob N1 bereits existiert und übersprungen wurde.
    - **IF** N1 ist übersprungen: Setze N1 auf `status: Reaktiviert`.
    - **ELSE:** Erstelle N1 neu mit `status: Offen`.
  4. **Pfad-Update:** Setze N1 an die erste Position im Lernpfad mit `status: Offen`.
  5. Nächstes Konzept wird N1 (P4 startet).

### 3.3 P6/P7: Verständnisprüfung und Adaption

- Trigger: Abschluss des Materials (P4/P5) und optionaler Test (P6).
- LLM-Rollen: Kurator (Test), Tutor (Feedback/Adaption).
- Entscheidungspunkt (Automatisierter Fortschritt):
  - IF P6 deaktiviert ODER Test-Score  $\geq$  targetScore:
    - Setze `status` des aktuellen Konzepts auf `Beherrscht` .
    - Gehe zum nächsten Konzept mit `status: Offen/Aktiv` .
  - ELSE (Test nicht bestanden):
    - Tutor analysiert die falschen Antworten (Fehler-Kategorisierung).
    - Entscheidungspunkt (Remediation): Tutor entscheidet:
      - IF Fehler sind leicht/oberflächlich: **Generiere neues, gezieltes Material** (P4) zum gleichen Konzept (z.B. nur ein Anwendungsbeispiel).
      - ELSE IF Fehler sind fundamental: **Starte P5.5 Lücken-Diagnose** (Das Problem liegt nicht hier, sondern davor).

Dieses Framework stellt sicher, dass das System reaktionsschnell, dokumentiert und vollständig adaptiv auf die individuellen Bedürfnisse und Fehleinschätzungen des Nutzers reagiert.