



# **Pflichtenheft**

## **eKI API (Arbeitstitel: eKI)**

Stand: 20.08.2025, Version 1



# 1. Management Summary

Die eKI-API liefert der Filmakademie eine automatisierte Sicherheitsprüfung für Drehbücher und ergänzt das eProjekt um konsistente Risikoanalysen mit konkreten Handlungsempfehlungen. Die Ergebnisse stehen unmittelbar allen relevanten Rollen (Produktion, Disposition, Sicherheit) zur Verfügung und beschleunigen die Sichtung deutlich.

Die eKI ist als Plattform ausgelegt und bildet den Grundstein für weitere KI-Module im eProjekt. In der ersten Ausbaustufe konzentriert sie sich auf die Sicherheitsprüfung physischer, psychischer und umgebungsbedingter Risiken mit Bewertung und Maßnahmenkatalog.

Datenschutz und Sicherheit sind durchgängig berücksichtigt: Inhalte werden nach Übergabe gelöscht, Übertragungen sind verschlüsselt und es existieren klare Lösch- und Aufbewahrungsregeln.

Der Projekterfolg wird daran gemessen, dass Berichte zuverlässig an Projekten hängen und die Durchsichtszeit der Fachrollen spürbar reduziert wird. Die fachlichen Inhalte werden vom Sicherheitsbeauftragten gepflegt, die IT stellt den Betrieb sicher.

Die Umsetzung erfolgt entlang einer zwölfmonatigen Roadmap mit monatlichen Ergebnispaketen. Die Vergütung beträgt 72.600 € zzgl. USt. (Festpreis), abgerechnet nach Meilensteinen. Risiken liegen vor allem in Schnittstellenänderungen und Abstimmungen, welche jedoch durch klare Prozesse abgesichert sind.

Das Projekt kann mit dem 01.01.2026 beginnen und wird voraussichtlich bis 31.01.2027 dauern.



## 2. Einleitung

Dieses Pflichtenheft definiert Ziele, Funktionen, Qualitätsmerkmale, Schnittstellen, Betriebs- und Sicherheitsanforderungen der eKI API. Die eKI ist eine modulare, serviceorientierte Backend-Schnittstelle, die Daten aus dem eProjekt entgegennimmt, KI-gestützt verarbeitet (Analyse, Validierung, Anreicherung) und transformierte Ergebnisse zurück an das eProjekt liefert. Der Schwerpunkt der Erstumsetzung liegt auf dem Modul „Sicherheitsprüfung“ für Drehbücher. Die Schnittstelle wird aber so aufgebaut, dass sie später beliebig weitere Module beherbergen kann. Mehr dazu unter §20.

**Auftraggeber:** Filmakademie Baden-Württemberg (eProjekt)

**Auftragnehmer:** Stefan Müller (Einzelunternehmer; Schulung, Beratung, Softwareentwicklung im KI-Umfeld)

**Adressaten dieses Dokuments:** Projektleitung, IT, Rechts-/DSB, Fachabteilungen (Produktion/Disposition), spätere Betriebsteams.

**Bezug & Standards:** OpenAPI 3.1.1 (Design-First), interne IT-Sicherheitsrichtlinien der Akademie, DSGVO, Stand der Technik für Web-APIs.



## 3. Zielbestimmung

### 3.1. Projektziele

Bau einer zentralen KI-API („eKI“) zur standardisierten Analyse von eProjekt-Artefakten im Intranet der Filmakademie BW.

Modul 1 in der KI-API ist die Sicherheitsprüfung und umfasst Identifikation potenziell gefährlicher oder psychisch belastender Szenen in Drehbüchern sowie die Ableitung konkreter Schutz- und Präventionsmaßnahmen nach festem Schema.

Rückkopplung zu eProjekt

Es werden strukturierte, maschinenlesbare Reports an das eProjekt (inkl. Schweregrad, Begründung, Maßnahmenvorschläge und Verantwortlichkeiten) zurückgeleitet und dort persistiert.

eKI wird gebaut mit dem Plattformgedanken und fokussiert auf Erweiterbarkeit. eKI bildet den Grundstein für weitere KI-Module zur Ergänzung der Projektdaten des ePro (z.B. Credits-Validierung, Genehmigungsvorcheck, DSGVO-Prüfungen) und kann modular nach diesem Projekt erweitert werden. Durch die künstliche Intelligenz von eKI werden Inhalte in ePro und ggf. anderen Software-Eigenentwicklungen selbständig ergänzt, um personelle Ressourcen zu schonen.

### 3.2 Lieferumfang

Lieferumfang v1.0

Sicherheitsprüfung inkl. Parser, Regelwerk, Scoring, Report-Erzeugung (JSON/PDF), Outbound-Adapter (Push/Pull, Idempotenz, Retries), Write-Through-Löschlogik, LLM-Adapter (Mistral API und lokales Mistral), KB-Grundlage (Dokumentenaufnahme, Embeddings), OpenAPI-Doku, Container-Setup, Monitoring-Dashboards, Betriebsdoku.

Meilensteine (High-Level, siehe Meilensteinplan für Details):

- Architektur & OpenAPI
- Parser & Szenenmodell

StefanAI - Research & Development  
Graeffstr. 22, 50823 Köln  
Mobil: +491775228242  
E-Mail: [info@stefanai.de](mailto:info@stefanai.de)



- Regelwerk & Scoring
- Outbound-Integration (Push/Pull)
- Stage-Integration (Mistral API)
- Lokale Mistral-Anbindung (Prod)
- Großdokument-Optimierung
- Sicherheit/Observability/UAT & Go-Live

### 3.3. Abgrenzung

Dieses Projekt hat folgende Themen nicht im Fokus:

- Kein Ersatz für Fachrollen wie Stunt-/Sicherheitskoordination oder psychologische Betreuung; die API liefert Hinweise, keine rechtsverbindlichen Gutachten.
- Keine Endnutzer-Oberfläche (außer Entwickler-Doku/Swagger).
- Keine Durchführung physischer Gefährdungsbeurteilungen am Set.
- Keine Rechtsberatung.
- Kein Aufbau von KI-Server Kapazitäten oder Beschaffung dieser.
- Keine Anpassungen an eProjekt oder den eProjekt API-Endpunkten.

### 3.4. Erfolgskriterien

- Report-Verfügbarkeit: Durch die eKI erzeugte Sicherheitsberichte sind an ePro-Projekten angehängt und im eProjekt einsehbar.
- Prozessnutzen: Bei der Durchsicht von Projekten führt der Einsatz der eKI-Berichte zu deutlich reduzierter Sichtungszeit der Fachrollen (interner Vergleich vor/nach Einführung). Das heißt Sicherheitsrisiken werden in Filmprojekten automatisiert erkannt und in ihren Auswirkungen adressiert.
- Technische Qualität: Abnahmetests bestanden (Push/Pull-Flows, Delete-on-Delivery, Großdokumente, Idempotenz, Log-Hygiene).



## 4. Produktumgebung

### 4.1 System- und Softwareumgebung

Das Projekt wird auf folgendem Techstack gebaut. Die technischen Komponenten haben sich hinsichtlich Performance, Erweiterbarkeit und Robustheit als Best Practices bewiesen.

- Backend/Framework: Python FastAPI (ASGI) – OpenAPI 3.1.1 als Single Source of Truth.
- Server: Uvicorn + Gunicorn.
- Datenbank (nur Metadaten/KB): PostgreSQL mit PgBouncer (Pooling); pgvector für Embeddings (Wissensspeicher).
- Cache/Queue: Redis (Caching, Rate Limiting, Task-Queues).
- Background Processing: Celery (Jobs), Flower (Monitoring).
- Containerisierung: Docker.
- Observability: Prometheus/Grafana (Metriken), strukturierte JSON-Logs, OpenTelemetry (Tracing).
- CI/CD: Build, Tests, Linting, Security-Scans, Deployment (Design-First-Workflow).
- Lokales KI-System: Mistral Small/Medium als lokaler Inferenzdienst (z. B. via Ollama/llama.cpp HTTP-Endpoint). Kein externer Cloud-Zugriff erforderlich.
- Wissensspeicher (KB, optional/gezielt): Verschlüsselter Objekt-Store (für freigegebene Unterlagen/Kataloge), Embeddings in PostgreSQL/pgvector. Keine dauerhafte Speicherung von Reports oder Drehbüchern (außer falls ePro dies explizit verlangt und freigibt; Default = transient).

Einbettung & Schnittstellen (high level)

- eProjekt-Technologie (gegeben): PHP-Anwendung mit SQL Server und Apache.
- Stage-Umgebung: Das eProjekt-Team stellt eine Stage des ePro mit Testfilmprojekt(en) und ggf. neuen/erweiterten Endpunkten bereit. Die eKI wird dort integriert und gegen Stage-Endpunkte getestet.
- Ablauf (Benutzer-unsichtbar): Beim Upload eines Drehbuchs in eProjekt löst dieses im Hintergrund den Aufruf von POST /v1/security/check(:async) der eKI aus (Service-to-Service).
- Liefermodell: Ergebnismrückgabe entweder
  - (a) Push: eKI postet den Report an die eProjekt-API (Write-Through) oder
  - (b) Pull: eProjekt pollt den Report einmalig ab. In Pull-Fällen hält eKI den



Report nur kurzzeitig ( $TTL \leq 6h$ , verschlüsselt) vor; nach erfolgreichem Abruf Sofort-Löschung.

- Nutzerbezug: eProjekt übergibt bei Start den Akteur-Kontext (X-Actor-User-Id, optional X-Actor-Project-Id). Die eKI verarbeitet on-behalf-of dieses Nutzers; der gesamte Prozess bleibt UI-unsichtbar.
- Kompatibilität/Minimalinvasivität: Keine direkten DB-Zugriffe auf SQL Server; Integration ausschließlich über REST-Endpunkte des eProjekt-Teams.
- AuthN/Z: Service-Account (Client-Credentials/API-Key) zwischen eProjekt↔eKI; optional Nutzer-Impersonation via Header (nur für Audit).
- Provider-Abstraktion: eKI unterstützt `LLM_PROVIDER=mistral_cloud|local_mistral` (Umschaltbar per Konfiguration).

### Nutzergruppen & Rollen

- System-Integratoren (eProjekt-Team): Anbindung, Bereitstellung der REST-Endpunkte inkl. technischem User.
- Fachbereiche: Arbeiten ausschließlich in eProjekt; sehen Status/Ergebnisse dort.
- Administratoren (ops): Schlüssel/Secrets, Monitoring, Audits (inhaltsarm)

## 4. Produktfunktionen

### 4.1 Modul – Sicherheitsprüfung

Das Modul der Sicherheitsprüfung hat folgenden Zweck. KI-gestützte Analyse von Drehbüchern (FDX bevorzugt, PDF mit OCR-Fallback) auf physische Gefährdungen und psychische Belastungen. Die eKI fungiert als durchleitende Verarbeitungsschicht (Processing-Only): Ergebnisse werden sofort an das ePro zurückgespielt; keine doppelte Datenhaltung in der eKI.

Das Modul hat folgende Kernfunktionen:

#### 1. Ingest & Parsing

- Upload/Referenz von FDX/PDF; robustes Parsing (Szenen, Locations, Tageszeit, Charaktere).
- Normalisierung: einheitliche ID-Vergabe (Script-ID, Scene-ID, Character-ID).



- Temporäre Verarbeitung in Arbeitsspeicher/temporärem Objekt-Store (verschlüsselt), ohne dauerhafte Speicherung in der DB.

## **2. Erkennung riskanter Inhalte (Taxonomie)**

- Physisch: Stunts, Stürze, Kämpfe, Waffen/Requisiten, Fahrzeuge, Höhe, Wasser/Eis, Feuer/Explosionen/SFX, Elektrik, Tiere, Wetter/Hitze/Kälte, Nacht-/Übermüdung, Massen-/Gedränge.
- Umgebung: gefährliche Locations (Dach, Baustelle, Straße, Gewässer), begrenzte Räume, Rauch/Staub, Lärm.
- Psychisch: Gewalt-/Missbrauchsdarstellungen, Tod/Trauer, traumareaktive Inhalte, sexualisierte Inhalte (alters-/jugendschutzrelevant), belastende Sprache/Diskriminierung.

Diese Taxonomie wird im Rahmen der Projektarbeit noch erweitert oder reduziert. Sollte aber bereits jetzt einen Mehrheit der notwendigen Informationen abdecken.

## **3. Risikobewertung & Scoring**

- Likelihood × Impact je Szene (Skala 1–5) → Severity (Low/Med/High/Critical).
- Berücksichtigung von Verletzlichkeit (Kinder, Tiere, Stunts-Double), Komplexität (Extras, Mehrkameras, Wasser/Feuer), Dauer der Exposition.

## **4. Maßnahmenableitung (Bibliothek)**

- Technische/organisatorische Maßnahmen: z. B. Rutschschutz, Matten, Rigging, SFX-Freigaben, Sicherheitsabstände, Stuntproben, Intimacy-Coordination, Closed-Set, Wärme-/Kälteschutz, Gehörschutz, Psy-Briefing/Debriefing, Notfall-/Rettungskette.
- Rollen/Verantwortung: Produktionsleitung, Stunt-/Intimacy-Coordination, Set-Sicherheit inkl. Fristen.

## **5. Report-Erzeugung & Rückgabe (Write-Through)**

- Maschinenlesbar (JSON): je Szene Findings, Severity, Evidenz (Rule-ID, Textauszug), Maßnahmenvorschläge, Verantwortliche, Fälligkeiten.
- Human-Report (PDF/HTML): Executive Summary, Szenenliste, To-Dos.
- Sofortige Rückübertragung an ePro (s. 4.2) — keine persistente Speicherung des Reports in eKI.





## **6. Audit & Nachvollziehbarkeit (inhaltsarm)**

- Nur Metadaten: job\_id, script\_id, Zeitstempel, Größe, Dauer, Status, ePro-Responsecode, Trace-ID. Kein inhaltlicher Report/PII in Logs oder DB.

## **4.2 Outbound-Integration ins ePro (Write-Back-Kontrakt)**

Das Ziel ist doppelte Datenhaltung zu vermeiden. eProjekt ist System of Record.

Es gibt zwei mögliche Lieferwege von eKI zu ePro. Welcher verwendet wird, muss im Projekt beschlossen werden.

Möglichkeit 1 „Push“: eKI postet Report an eProjekt (POST {EPRO\_BASE\_URL}/reports/security). Bei 201-Rückmeldung Sofort-Löschung. Sollte ePro nicht verfügbar sein, dann Retry in definierter Dauer und Häufigkeit.

Möglichkeit 2 „Pull“ (One-Shot): eProjekt ruft GET /v1/security/reports/{report\_id} ab; nach erstem 2xx-Abruf löscht eKI den Inhalt sofort. Report verschlüsselt ≤ 6 h verfügbar; danach Auto-Löschung.



## 4.3 Lokale KI-Integration & Wissensspeicher (KB)

Ziel: eKI arbeitet ohne externe Cloud, nur mit lokalem Mistral Small/Medium. In der Entwicklung hingegen wird zur Reduktion der Kosten ausschließlich Mistral als externer Cloud-Dienst per API eingesetzt. Zudem müssen zur Analyse der Drehbücher auch interne Dokumente in einem Wissensspeicher abgelegt werden.

Funktionsweise (intern):

- Inferenzdienst lokal (Ollama/llama.cpp HTTP-API).
- Kontextzugriff auf freigegebene Unterlagen/Kataloge (z. B. Sicherheitsleitfäden, Stunt-SOPs, Checklisten, regulatorische Vorgaben).
- Retriever-Pipeline (Vektor-Speicher): Embedding-Erstellung (pgvector), Chunking (ca. 800–1500 Tokens), Top-k-Suche; nur freigegebene Quellen.
- Sicherheitsmaßnahmen: Verschlüsselte Speicherung, Zugriffskontrolle, Quellen-Tagging, Zugriffsprotokoll, TTL-Jobs

Fachliche Inhalte & Pflege:

Die Dokumente zur Einsortierung der Sicherheitsrisiken werden vom Content-Owner bereitgestellt. Beim Auftraggeber ist das der Sicherheitsbeauftragte, welcher fachliche Inhalte anfertigt, liefert und pflegt (Leitfäden, Kataloge, Maßnahmenbibliothek). Diese sind für das Modul „Sicherheitsprüfung“ essentiell.

## 5. Qualitätsanforderungen

**Sicherheit von Anfang an:** Alle Schnittstellen sind durchgehend verschlüsselt (TLS). Der Zugriff erfolgt mit sicheren Tokens (OAuth2/Bearer) oder API-Schlüsseln; zwischen eProjekt und eKI wird in der Regel ein eigener Service-Account verwendet. Aufrufe werden automatisch begrenzt (Rate Limiting), und alle Vorgänge werden nachvollziehbar protokolliert – ohne Inhalte aus Drehbüchern oder Berichten in die Logs zu schreiben.

**Datenschutz/DSGVO:** Wir verarbeiten nur die Daten, die wirklich nötig sind (Datenminimierung) und ausschließlich für den vorgesehenen Zweck. Dateien und Zwischenergebnisse sind während der Übertragung und – wo technisch erforderlich – im Speicher verschlüsselt. Inhalte werden nach erfolgreicher Übergabe an das eProjekt sofort gelöscht; in Fehlerfällen gibt es eine kurze,



automatisch endende Aufbewahrung (max. 6 Stunden). Dauerhaft gespeichert werden lediglich technische Metadaten zur Nachvollziehbarkeit.

**Leistung & Skalierung (Zielbild):** Kleine Prüfungen (bis ca. 50 Szenen) beantworten wir in der Regel in unter 20 Minuten. Ein 120-seitiges Drehbuch wird asynchron in höchstens 60 Minuten verarbeitet; sehr große Drehbücher mit 300–350 Seiten (Text-PDF) benötigen bis zu 3 Stunden. Bei Bedarf kann die Verarbeitung horizontal skaliert werden (zusätzliche Worker). Sie hängt allerdings nicht unwesentlich von der Höhe des eingesetzten Grafikkartenspeichers ab.

**Zuverlässigkeit:** Schlägt die Übergabe an das eProjekt einmal fehl, startet das System automatisch neue Zustellversuche mit wachsendem Abstand abhängig von der vereinbarten Zustellmethode – insgesamt höchstens über 6 Stunden. Doppelte Zustellungen werden technisch vermieden (idempotentes Design). Fehlgelaufene Aufträge landen in einer sicheren Warteschlange und werden ausgewertet.

**Wartbarkeit & Qualität:** Der Code ist modular aufgebaut; Schnittstellen sind klar getrennt und dokumentiert. Einheitliche Qualitätsprüfungen (Linting) sind Teil des Build-Prozesses; in der Kernlogik streben wir mindestens 80 % Testabdeckung an.

**Entwickler-Erlebnis & Doku:** Die API wird mit einer interaktiven Oberfläche (Swagger/Redoc) ausgeliefert – inklusive Beispielanfragen, Postman-Sammlung und einem versionierten Änderungsprotokoll.

**Integration mit eProjekt:** Die Anbindung erfolgt ausschließlich über die bereitgestellten REST-Endpunkte. Es gibt keine direkten Zugriffe auf den SQL-Server des eProjekts. Nötige Anpassungen beschränken sich auf neue oder erweiterte Endpunkte.

## 6. Datenmodell (Auszug / Beispiele)

### Script (transient)

```
{  
  "script_id": "uuid",  
  "title": "string",  
  "format": "FDX|PDF",
```



```
"scenes": [{
  "scene_id": "uuid",
  "number": "string",
  "location": "string",
  "time_of_day": "DAY|NIGHT|INT|EXT|MIXED",
  "characters": ["string"],
  "text": "string"
}]
}
```

## **SecurityFinding (transient)**

```
{
  "scene_id": "uuid",
  "category": "PHYSICAL|ENVIRONMENTAL|PSYCHOLOGICAL",
  "class": "FIRE|WATER|HEIGHT|VEHICLE|WEAPON|INTIMACY|...",
  "severity": "LOW|MEDIUM|HIGH|CRITICAL",
  "likelihood": 1,
  "impact": 1,
  "evidence": "excerpt or rule rationale",
  "rule_id": "SEC-R-012",
  "measures": [{"code": "MAT-ANTI-SLIP", "title": "Rutschschutz-Matten"},
  "responsible": "Set Safety", "due": "shooting-2d"}]
}
```

## **RiskReport (transient → write-through)**

```
{
  "report_id": "uuid",
  "script_id": "uuid",
  "summary": {"total_scenes": 120, "findings": {"LOW": 10, "MEDIUM": 8, "HIGH": 3,
  "CRITICAL": 1}},
  "findings": [SecurityFinding],
  "generated_at": "datetime",
  "version": {"rules": "1.0.0", "engine": "1.0.0"}
}
```

## **AuditMetadata (persistiert, inhaltsarm)**



```
{
  "job_id": "uuid",
  "script_id": "uuid",
  "report_id": "uuid",
  "status": "queued|running|delivering|succeeded|failed",
  "metrics": {"duration_ms": 12345, "pages": 120, "size_bytes": 1048576},
  "delivery": {"last_code": 201, "attempts": 1, "idempotency_key": "uuid"},
  "timestamps": {"created": "datetime", "delivered": "datetime"}
}
```

### **KnowledgeDocument (KB, optional/persistiert)**

```
{
  "doc_id": "uuid",
  "title": "string",
  "source": "UPLOAD|SHARE|URL",
  "tenant_id": "uuid",
  "ttl_hours": 720,
  "tags": ["SOP", "Checkliste"],
  "created_at": "datetime",
  "expires_at": "datetime"
}
```

### **Embedding (persistiert)**

```
{
  "embedding_id": "uuid",
  "doc_id": "uuid",
  "chunk_id": "string",
  "vector": [0.01, ...],
  "dim": 1024,
  "offset": 0,
  "length": 1200,
  "hash": "sha256"
}
```

Persistenz-Prinzip: Nur AuditMetadata wird in der DB gespeichert. Script, Findings und RiskReport existieren nur transient, werden nach erfolgreicher Übertragung an das ePro sofort gelöscht und verbleiben höchstens bis zum Ende des Retry-Fensters ( $\leq 6$  h TTL) in einem verschlüsselten Buffer.



## 7. Benutzerschnittstellen (DX)

Die Schnittstelle ist vollständig nach OpenAPI 3.1.1 beschrieben und wird mit einer interaktiven Oberfläche (Swagger/Redoc) ausgeliefert, über die sich alle Endpunkte einfach testen lassen. Für Betrieb und Administration gibt es Dashboards mit Kennzahlen (z. B. Zustellraten, Latenzen), eine Job-Übersicht ohne Inhalte sowie strukturierte Logs mit Trace-IDs zur schnellen Fehlersuche. Für die Integration stellen wir zusätzlich eine Postman-Collection und fertige Beispielanfragen bereit.

## 8. Abnahme des Projektes

### Abnahme – Grundprinzip

Wir nehmen pro Meilenstein auf Basis prüfbarer Nachweise ab: eine gültige OpenAPI-Beschreibung, lauffähige Builds, Testprotokolle und ein kurzes Demo-Video. Die Sichtung passiert gemeinsam im Review.

### Testumfang (was wir prüfen)

Wir prüfen in vier Bereichen:

1. Techniktests: Unit-Tests für Parser/Regeln/Scoring, Contract-Tests gegen OpenAPI, Integrationstests für Push/Pull und Idempotenz, Security-Tests (Zugriff, keine PII-Leaks) sowie Lasttests mit realistischen Skripten.
2. Stage-Betrieb: Ende-zu-Ende mit der Mistral-API gegen das Stage-ePro (mit Testfilmprojekten).
3. Produktionsbetrieb: Ende-zu-Ende mit lokalem Mistral (keine externen LLM-Aufrufe).
4. Wissensbasis (KB): Ablaufzeiten (TTL) und Zugriff nur auf freigegebene Quellen.

Verbindliche Abnahmetests (müssen bestehen):

1. Automatischer Start: Ein Upload im eProjekt startet den eKI-Job selbständig (Service-Account).
2. Push-Fluss: Nach erfolgreicher Übergabe (2xx) sind keine Inhalte mehr in der eKI (nur Metadaten).



3. Pull-Fluss: Report ist einmalig abrufbar; nach dem ersten 2xx wird er sofort gelöscht.
4. Retries & TTL: Bei Fehlern wird bis zu 6 Stunden automatisch erneut zugestellt; danach automatische Löschung und Metadaten-Webhook.
5. Idempotenz: Gleicher Idempotency-Key erzeugt keine Duplikate im eProjekt.
6. Großdokumente: 300–350 Seiten werden in  $\leq 40$  Min verarbeitet; Ressourcenlimits werden eingehalten.
7. Log-Hygiene: In Logs stehen keine Drehbuchtexte oder Findings.
8. Stage-Sign-off: Erfolgreiche Ende-zu-Ende-Tests auf Stage mit der Mistral-API.
9. Prod-Cutover-Sign-off: Erfolgreiche Paritäts-/Smoke-Tests mit lokalem Mistral (ohne externe Calls).

### **Abnahmekriterien (gesamt)**

Die OpenAPI ist gültig, Authentifizierung und Monitoring laufen, alle 9 Tests sind bestanden, und Betriebsdoku plus Postman-Collection liegen vor.

### **Ablauf mit Kanban/Trello im Projektbetrieb**

Jeder Monats-Meilenstein hat eine Trello-Karte mit Beschreibung, Akzeptanzkriterien und Links zu den Nachweisen. Nach Fertigstellung verschiebt der Auftragnehmer die Karte auf „Zur Abnahme“. Der Auftraggeber prüft und verschiebt bei Zustimmung auf „Abgenommen“ (kurzer Kommentar genügt). Die Rechnung für den Meilenstein wird nach dieser Trello-Freigabe gestellt.

## **9. Risiken, Annahmen, Abhängigkeiten**

Risiken:

- Stage-Bereitstellung & Endpunkte durch eProjekt-Team verzögern Integration; Abhängigkeit vom Teamkalender.
- Provider-Wechsel (Cloud→Lokal): Modell-/Prompt-Parität; Performance-Divergenzen möglich (Mitigation: Paritäts-Tests, Tuning).
- Datenfreigaben: Für Stage/Einsatz der Mistral API sind ggf. Testdaten/Anonymisierung erforderlich.
- Qualität von PDF-Skripten (OCR-Fehler); Domänenvarianten der Szenen-Notation; False Positives/Negatives.



Annahmen:

- eProjekt-Team stellt Stage mit Testfilmprojekt(en) + Endpunkten zur Verfügung.
- Freigabe/Anonymisierung für etwaige echte Testskripte erfolgt durch eProjekt.
- In Prod steht eine geeignete Hardware (bevorzugt GPU  $\geq 24$  GB) für lokales Mistral bereit.
- Fachinhalte (KB) werden vom Sicherheitsbeauftragten der Filmakademie fortlaufend geliefert und gepflegt.

## 10. 12-Monats-Roadmap (Planungsstand)

Hinweis: Die folgende Roadmap strukturiert die Ergebnisse in monatliche Arbeitspakete mit klaren Artefakten. Regelmäßige Reviews mit dem Auftraggeber sind vorgesehen.

### **M01 – Projektgerüst & OpenAPI v0.1**

Artefakte: Repository, CI/CD, OpenAPI v0.1, Schnittstellen-Testadapter, Postman-Collection, kompaktes Demo-Material.

### **M02 – Parser Basis (FDX) & Testdataset**

Artefakte: FDX-Parser, Szenenmodell, kuratiertes Testdataset, Unit-Tests.

### **M03 – PDF & Streaming-Parsing**

Artefakte: PDF-(Text/OCR)-Parsing, Streaming-Pipeline, Benchmark für 120 Seiten.

### **M04 – Risiko-Taxonomie v1 & Scoring**

Artefakte: Regelwerk (physisch/psychisch/umgebung), Scoring-Engine, Maßnahmenkatalog-Seed.

### **M05 – Reports (JSON/PDF) & One-Shot-GET**

Artefakte: JSON/PDF-Generator, Idempotenz-Design, One-Shot-GET-Endpoint, Mapping zur ePro-API.





## **M06 – LLM-Adapter (Mistral API) & KB-Grundlage**

Artefakte: Cloud-Adapter, Prompt-Templates, KB-Ingest (pgvector) mit freigegebenen Beispielunterlagen.

## **M07 – Großdokument-Optimierung (>300 Seiten)**

Artefakte: Parallelisierung, Ressourcen-Tuning, Benchmark 300–350 Seiten (Text-PDF).

## **M08 – Security/Privacy & Delete-on-Delivery**

Artefakte: Lösch-/TTL-Mechanismen, Log-Hygiene, Secrets-Handling, technische Nachweise.

## **M09 – Observability & SLOs**

Artefakte: Prometheus-Metriken, Grafana-Dashboards, Alerts, SLO-Definitionen.

## **M10 – Outbound-Adapter Hardening (Push/Pull)**

Artefakte: Retries/Backoff, Dead-Letter-Queues, Idempotenz-Nachweise, Failover-Szenarien.

## **M11 – Lokaler LLM-Adapter & Paritätstests**

Artefakte: LocalMistralAdapter, Paritäts-Reports Cloud↔Lokal, Container-Images, Betriebsleitfaden.

## **M12 – UAT-Paket & Übergabe**

Artefakte: UAT-Paket (Testskripte, Protokolle, OpenAPI v1.0, Betriebsdoku), Go-Live-Checkliste, Schulungsunterlagen.



## 11. Aufwände nach Meilensteinen

Meilenstein	Beschreibung	PT
M01	Projektgerüst & OpenAPI v0.1	4
M02	Parser Basis (FDX) & Testdataset	4
M03	PDF & Streaming-Parsing	4
M04	Risiko-Taxonomie v1 & Scoring	3
M05	Reports (JSON/PDF) & One-Shot-GET	3
M06	LLM-Adapter (Mistral API) & KB-Grundlage	4
M07	Großdokument-Optimierung (>300 Seiten)	4
M08	Security/Privacy & Delete-on-Delivery	3
M09	Observability & SLOs	3
M10	Outbound-Adapter Hardening (Push/Pull)	4
M11	Lokaler LLM-Adapter & Paritätstests	4
M12	UAT-Paket & Übergabe	4

Summe (max): 44 PT.

Hinweis: Jeder Meilenstein wird als Trello-Karte im gemeinsamen Kanban-Board geführt; Abnahme und Freigabe erfolgen dort.

## 17. Kommerzielle Rahmenbedingungen

Vergütungsmodell: Festpreis auf Basis der maximalen Aufwandsschätzung.

Festpreis gesamt: 72.600 € zzgl. USt. (44 PT × 1.650 €).

Abrechnung je Meilenstein gemäß PT-Zuordnung und nach Freigabe der Trello-Karte durch den Auftraggeber.

Beispiel:

- 4-PT-Meilenstein: 6.600 € zzgl. USt.
- 3-PT-Meilenstein: 4.950 € zzgl. USt.



## 18. Rechte & Eigentum

Die Eigentums- und Nutzungsrechte liegen nach Vollendung des Projekts und vollständiger Vergütung beim Auftraggeber.

Ausgenommen davon sind Drittkomponenten wie Open-Source- und Drittsoftware. Diese verbleiben unter ihren jeweiligen Lizenzen und es werden keine exklusiven Rechte daran übertragen.

Schutzrechte Dritter: Der Auftragnehmer sichert zu, nur rechtmäßig verwendbare Inhalte/Komponenten einzusetzen.

## 19. Abstimmung (“Brückenmeetings”)

eKI und ePro stimmen ihre Schnittstellen gemeinsam ab – damit es keine Überraschungen oder inkompatiblen Änderungen gibt.

### **Rhythmus & Format**

Alle 2 Wochen, ca. 45 Min. Bei akuten API-Änderungen oder Blockern gibt's kurzfristig einen Zusatztermin. Remote/hybrid, fester Slot, kurze Agenda und Demo/Beispiele.

### **Teilnehmende**

eKI-Entwicklung und ePro-Tech-Lead/Backend. Optional: Sicherheitsbeauftragter (Fachinhalte) und IT-Betrieb (Infrastruktur).

### **Was bringen wir mit?**

Kurzbeschreibung der Änderung (1-Pager/RFC), OpenAPI-Diff, Beispiel-Requests/Responses, Update des Testadapters sowie offene Risiken/Entscheidungen.

### **Was kommt heraus?**

Klare Entscheidung, To-do-Liste mit Verantwortlichen und Termin, aktualisierte API-Version, geplanter Release-/Abkündigungszeitpunkt und erfolgreiche Handshake-Tests auf Stage.



## **Dokumentation**

Kurzer Eintrag im gemeinsamen Bridge-Log (wer/was/bis wann), Artefakte versioniert.

## **Eskalation**

Bleibt etwas >48 h blockiert, geht es an die Projektleitung; Entscheidung im nächsten Steering oder ad-hoc.

# **20. Betrieb & Support nach Projektlaufzeit**

Supportvertrag: Vor Ablauf der 12-monatigen Projektlaufzeit wird ein Supportvertrag geschlossen, der Support und Wartung (Fehlerkorrekturen, kleinere Anpassungen, Sicherheitsupdates) regelt.

Fehlerbehandlung im Betrieb: Meldung → Reproduktion → Fehlerklasse → Fix-Release → Nachtest; Reaktions-/Behebungszeiten werden im Supportvertrag festgelegt.

Der Supportvertrag wird spätestens mit Passieren des 10. Meilensteins vereinbart.

# **21. Ausblick & Plattformcharakter (weitere Module)**

Die eKI wird als skalierbare Plattform betrieben. Auf Basis der definierten Architektur können in späteren Ausbaustufen u. a. folgende Module integriert werden (Beispiele):

- Abspann-Check (Credits Validation): Plausibilisierung/Normierung von Credits gegen Stammdaten, Exportformate für Festivals/Sender.
- Genehmigungsvorcheck: Hinweise zu jugendschutz-/arbeitszeit-/ortsbezogenen Auflagen, Checklisten-Generierung (prä-juristisch).
- Gender- & Diversitäts-Check: Indikatoren, Sprache/Representation-Hinweise, freiwillige Leitfäden.
- Datenschutz-/PII-Scan & Einwilligungs-Matching: Erkennung personenbezogener Daten in Artefakten und Abgleich mit verfügbaren Einwilligungen/Verträgen.
- Intimacy-Coordination-Support: Checklisten, Closed-Set-Empfehlungen, Rollen/Verantwortlichkeiten.

StefanAI - Research & Development  
Graeffstr. 22, 50823 Köln  
Mobil: +491775228242  
E-Mail: [info@stefanai.de](mailto:info@stefanai.de)



- Produktionsplanung/Disposition-Assist: Tagging von Szenen/Orten/Assets zur Unterstützung der Disposition.
- Festival-/Wettbewerb-Export-Check: Formale/inhaltliche Mindestanforderungen, Format-Exports.
- Qualitäts-/Kontinuitäts-Checks: Konsistenzprüfungen (Figuren, Requisiten, Orte, Zeitachsen).

Die eKI-API und ihre Infrastruktur (OpenAPI-Verträge, Auth, Observability, Outbound-Adapter, KB-Mechanismen) bilden damit den technischen Grundstein für diese und weitere Module – ohne erneuten Architekturaufbau.



## Anhang: OpenAPI 3.1 – Skizze

```
openapi: 3.1.0
info:
  title: eKI Security Check API
  version: 1.0.0
servers:
  - url: https://eki.example/api
security:
  - bearerAuth: []
components:
  securitySchemes:
    bearerAuth:
      type: http
      scheme: bearer
  schemas:
    CheckRequest:
      type: object
      properties:
        ruleset: { type: string, default: default }
        delivery: { type: string, enum: [push, pull], default: push }
        locale: { type: string, default: de-DE }
      required: [delivery]
    JobStatus:
      type: object
      properties:
        job_id: { type: string, format: uuid }
        status: { type: string, enum: [queued, running, delivering, succeeded, failed] }
        report_id: { type: string, nullable: true }
        created_at: { type: string, format: date-time }
    RiskReport:
      type: object
      properties:
        report_id: { type: string, format: uuid }
        script_id: { type: string, format: uuid }
        summary:
          type: object
          properties:
            total_scenes: { type: integer }
            findings:
              type: object
```



```
    additionalProperties: { type: integer }
    generated_at: { type: string, format: date-time }
    required: [report_id, script_id, summary]
paths:
  /v1/security/check:
    post:
      summary: Synchroner Sicherheitscheck (kleine Payloads)
      security: [{ bearerAuth: [] }]
      requestBody:
        required: true
        content:
          multipart/form-data:
            schema:
              type: object
              properties:
                file: { type: string, format: binary }
                request: { $ref: '#/components/schemas/CheckRequest' }
                required: [file]
      responses:
        '200':
          description: Ergebnis zugestellt (push) oder bereit zur Abholung (pull)
          content:
            application/json:
              schema:
                type: object
                properties:
                  delivery: { type: string, enum: [push, pull] }
                  status: { type: string, enum: [delivered, pending] }
                  report_id: { type: string, nullable: true }
        '400': { description: Bad request }
        '401': { description: Unauthorized }
  /v1/security/check:async:
    post:
      summary: Asynchroner Sicherheitscheck (große Dateien)
      requestBody:
        required: true
        content:
          multipart/form-data:
            schema:
              type: object
              properties:
                file: { type: string, format: binary }
```

StefanAI - Research & Development  
Graeffstr. 22, 50823 Köln  
Mobil: +491775228242  
E-Mail: [info@stefanai.de](mailto:info@stefanai.de)



```
    request: { $ref: '#/components/schemas/CheckRequest' }
    required: [file]
responses:
  '202':
    description: Job angenommen
    content:
      application/json:
        schema: { $ref: '#/components/schemas/JobStatus' }
/v1/security/jobs/{job_id}:
get:
  summary: Jobstatus (ohne Reportinhalte)
  parameters:
    - in: path
      name: job_id
      required: true
      schema: { type: string }
  responses:
    '200':
      description: Status
      content:
        application/json:
          schema: { $ref: '#/components/schemas/JobStatus' }
    '404': { description: Not found }
/v1/security/reports/{report_id}:
get:
  summary: One-Shot-Abholung (nur bei delivery=pull)
  parameters:
    - in: path
      name: report_id
      required: true
      schema: { type: string }
  responses:
    '200':
      description: RiskReport
      headers:
        X-One-Shot:
          description: Wird nach erstem 2xx-Abruf gelöscht
          schema: { type: string, example: 'true' }
      content:
        application/json:
          schema: { $ref: '#/components/schemas/RiskReport' }
    '404': { description: Nicht (mehr) verfügbar }
```





x-webhooks:  
security.delivery.failed:  
post:  
summary: Zustellung fehlgeschlagen (Metadaten)  
requestBody:  
required: true  
content:  
application/json:  
schema:  
type: object  
properties:  
job\_id: { type: string }  
report\_id: { type: string }  
reason: { type: string }  
attempts: { type: integer }