

# Architekturempfehlung für Breakpilot – Offene, modulare Bildungsplattform im DACH-Raum

## Einführung und Kontext

**Breakpilot** ist konzipiert als digitale Plattform, die Lehrkräfte, Content-Creator und Eltern in der DACH-Region (Deutschland, Österreich, Schweiz) verbindet, um hochwertige Bildungs- und Pauseninhalte bereitzustellen. Ziel ist es, Lehrkräften einen **Arbeitsraum** für Vorbereitung und Durchführung von Lerneinheiten (insbesondere Bewegungspausen und Bildungsangebote), Creatoren ein **Studio** zur Erstellung und Verwaltung didaktischer Inhalte, und Eltern eine **Community** zum Austausch und zur Unterstützung zu bieten. Die nachfolgende Empfehlung beschreibt eine **umfassende Systemarchitektur** für Breakpilot, die sowohl strategische Ziele als auch technische Anforderungen berücksichtigt. Zentrale Leitprinzipien sind der Einsatz **offener Technologien** (Open Source) über alle Komponenten hinweg, strikte **Trennung der Nutzerbereiche** (Lehrer, Creator, Eltern) bei gleichzeitiger Vernetzung, ein innovatives **Monetarisierungskonzept** auf Basis von Impact-Punkten statt Kryptowährung, klare **Lizenzierungsmodelle** für Inhalte, eine robuste **Rollen- und Rechtearchitektur**, vollständige **DSGVO-Konformität** sowie Mechanismen für **Community-Moderation und Qualitätskontrolle**.

Diese Architekturempfehlung gliedert sich in einen **strategischen Teil** – der die Ziele, den Mehrwert und eine mögliche Roadmap skizziert – und einen **technischen Teil**, der die vorgeschlagenen Komponenten, Module, Schnittstellen, Datenflüsse und Governance-Mechanismen im Detail erläutert.

## Strategische Ausrichtung und Ziele

**Mission und Ziele:** Breakpilot soll Lehrkräften die Vorbereitung und Durchführung von (Pausen-)Lernaktivitäten erleichtern, qualitativ hochwertige Inhalte verfügbar machen und eine aktive Beteiligung von Content-Erstellern und Eltern fördern. Im Fokus steht der **DACH-Bildungsraum**, was bedeutet, dass sprachliche, kulturelle und curriculare Besonderheiten (z.B. Lehrpläne in DE/AT/CH) berücksichtigt werden. Durch eine *offene Plattform* mit Community-Charakter sollen Austausch und Kooperation zwischen Bildungspartnern gestärkt werden. Das Projekt verfolgt einen **Open-Education-Ansatz**, d.h. die kostengünstige Bereitstellung von Bildungsressourcen (idealerweise als Open Educational Resources) und die aktive Einbindung der Benutzer bei gleichzeitiger Wahrung höchster Datenschutz- und Qualitätsstandards.

**Differenzierung:** Breakpilot unterscheidet sich von herkömmlichen Lernmanagementsystemen und Content-Marktplätzen durch mehrere Alleinstellungsmerkmale:

- **Open-Source-First:** Nahezu alle Kernkomponenten – von Frontend-Frameworks über Backend-Services bis hin zur Hosting-Infrastruktur – sollen auf **Open-Source-**

**Software** basieren. Dies fördert Transparenz, Sicherheit und vermeidet Vendor-Lock-in. Viele Bildungseinrichtungen in der DACH-Region setzen bereits auf Open-Source-Lösungen; z.B. ist Moodle als offene Lernplattform weltweit im Einsatz[1] und mit **OLAT** gibt es ein Open-Source-LMS, das speziell im deutschsprachigen Raum verbreitet ist[2]. Breakpilot knüpft an diese Kultur an und integriert bewährte Open-Source-Tools (siehe unten) nahtlos in einer neuen Plattform.

- **Getrennter, aber vernetzter Nutzungsbereich:** Die Plattform ist *modular* aufgebaut, mit klar getrennten Bereichen für die Hauptnutzergruppen: **Lehrkräfte (Breakpilot-Arbeitsraum), Content-Creators (Creator-Studio) und Eltern (Eltern-Community)**. Diese Trennung stellt sicher, dass jede Gruppe eine auf ihre Bedürfnisse zugeschnittene Umgebung vorfindet[3]. So können Lehrkräfte z.B. in einem geschützten Bereich Unterrichtsmaterial suchen, planen und mit Kollegen teilen, während Creators im Studio neue Inhalte produzieren und verwalten. Eltern wiederum erhalten über die Community Zugang zu Diskussionen, Feedback-Möglichkeiten und begleitenden Inhalten zur schulischen Bildung ihrer Kinder – z.B. Informationen, wie sie bestimmte Aktivitäten zu Hause unterstützen können. Trotz dieser Trennung sind die Bereiche **technisch und organisatorisch vernetzt**: Über **Single Sign-On (SSO)** und abgestimmte Schnittstellen können z.B. Lehrkräfte auf Community-Diskussionen (soweit zulässig) zugreifen oder Creator mit Lehrern kollaborieren, ohne die sichere Trennung der Rollen aufzugeben. Diese modularen Teilbereiche erhöhen sowohl die **Usability** (jedem Nutzer nur relevante Funktionen anzeigen) als auch die **Sicherheit** (Rollenrechte strikt isolieren)[4].
- **Impact-basiertes Monetarisierungssystem:** Anstatt auf traditionelle Pay-per-Content-Modelle oder kryptowährungsbasierte Tokens setzt Breakpilot auf ein *Impact-Scoring-System* zur Monetarisierung. Dieses **nicht-tokenisierte** Belohnungssystem vergütet Content-Creators basierend auf der Wirkung und Nutzung ihrer Inhalte. Ein solches System vermeidet die Komplexität und Compliance-Risiken von Krypto-Tokens und ist erheblich **einfacher und rechtssicherer** zu implementieren[5]. Anders als z.B. die Social-Media-Plattform Minds, die Nutzer-Engagement mit Kryptowährung belohnt[6], bleiben bei Breakpilot alle Bewertungen und Punkte **intern** – sie dienen ausschließlich der fairen Verteilung von Einnahmen und der Reputation innerhalb der Community, ohne spekulativen Handelswert. Die *Impact-Punkte* werden z.B. durch Kennzahlen wie Nutzungsanzahl eines Inhalts (wie oft setzen Lehrer es ein), Qualität (Bewertungen durch Lehrkräfte) und evtl. Rückmeldungen der Eltern oder pädagogische Wirksamkeit bestimmt. Dadurch werden Creator **leistungsorientiert entlohnt**, und es entsteht ein Anreiz, didaktisch wertvolle, oft genutzte Inhalte bereitzustellen. Monetäre Mittel könnten etwa durch Schulabonnements, Fördergelder oder Premium-Services generiert und anschließend gemäß der Impact-Scores an die Creator verteilt werden. Dieses Verfahren sichert Nachhaltigkeit, ohne Lehrkräfte für einzelne Inhalte zur Kasse zu bitten. (Zum Governance-Aspekt siehe unten: die Regeln der Score-Berechnung und Verteilung müssen transparent und community-abgesegnet sein, um Akzeptanz zu finden.)

- **Klare Lizenzierungsstrategie:** Um sowohl offenen Bildungsansätzen gerecht zu werden als auch die Interessen der Plattform und Creator zu wahren, wird ein flexibles Lizenzmodell vorgeschlagen. Dabei können Inhalte entweder unter **Creative-Commons-Lizenzen** oder unter einer speziellen **Plattformlizenz** veröffentlicht werden. Creative-Commons (CC) sind im Bildungsbereich quasi Standard für offene Inhalte[7] – sie ermöglichen es Urhebern, anderen im Voraus bestimmte Nutzungsrechte einzuräumen. Dies fördert das einfache Teilen und Remixen von Materialien, was im Sinne der OER-Bewegung die Verbreitung guter Inhalte maximiert[8]. Alternativ kann eine *Plattformlizenz* zum Einsatz kommen, die z.B. besagt, dass Inhalte frei auf Breakpilot verwendet werden dürfen, aber außerhalb der Plattform nur mit separater Genehmigung. Diese Option könnte für solche Fälle dienen, in denen Creator ihre Werke nicht komplett freigeben möchten, oder für durch Breakpilot beauftragte exklusive Inhalte. Strategisch empfehlen wir jedoch, **weitgehend auf CC-Lizenzen zu setzen**, etwa CC BY-SA oder CC BY-NC (zur Sicherung nicht-kommerzieller Nutzung). Dies senkt rechtliche Hürden für Schulen und Lehrkräfte, erhöht die Reichweite der Materialien und unterstreicht den Open-Education-Grundgedanken der Plattform. Wichtig ist, dass die Lizenzierung **einheitlich und gut kommuniziert** wird – z.B. durch Einbindung eines Lizenzwählers bei Uploads und entsprechende Kennzeichnung jedes Inhalts. Creative Commons selbst stellt Leitfäden bereit, wie Plattformen CC in UI und AGB integrieren können[9].
- **Hohe Datenschutz- und Sicherheitsstandards:** Als Bildungsplattform im deutschsprachigen Raum muss Breakpilot von Anfang an **DSGVO-konform** und datenschutzfreundlich gestaltet sein. Strategisch bedeutet dies, dass *Privacy by Design* ein Kernprinzip ist – Datenschutz nicht nur als Compliance-Aufgabe, sondern als Qualitätsmerkmal der Architektur[10]. Nutzer (insbesondere Lehrkräfte und Eltern) sollen Vertrauen haben, dass ihre persönlichen Daten und die Daten der Schüler, mit denen sie arbeiten, geschützt sind. Alle Funktionen werden daraufhin geprüft, nur die **notwendigsten personenbezogenen Daten** zu erheben und zu verarbeiten (Datenminimierung)[11]. Die Speicherung erfolgt *ausschließlich auf Servern in der EU* (vorzugsweise in Deutschland oder Nachbarländern), um Rechtskonformität und Datensouveränität sicherzustellen[12]. Durch den Einsatz offener Technologien kann Breakpilot zudem Transparenz über die Datenflüsse schaffen – idealerweise ist der Code quelloffen, sodass die Community Sicherheitsprüfungen durchführen kann. Strategisch kann Breakpilot somit mit **Datenschutz als Wettbewerbsvorteil** auftreten, was im Schulumfeld besonders bedeutsam ist (Stichwort: viele Schulen meiden z.B. US-Cloud-Dienste wegen Privacy Shield/SCC-Problematik[13] – eine Open-Source-Lösung in EU-Cloud erfüllt diese Anforderungen besser).
- **Community-Moderation und Qualitätssicherung:** Ein wesentliches Unterscheidungsmerkmal ist die geplante *Community-Governance* von Breakpilot. Anstatt nur zentral moderiert zu werden, soll die Community – vor allem erfahrene Lehrkräfte und ggf. Elternvertreter – in die **Moderation und Qualitätssicherung**

einbezogen werden. Breakpilot etabliert dazu klare **Community-Richtlinien** und Moderationsrollen. So werden etwa engagierte Nutzer zu Moderatoren ernannt, die Beiträge in der Eltern-Community begleiten und bei Bedarf eingreifen (siehe Moderationskonzept unten). Inhalte von Creators können durch **Peer-Review** und Nutzerbewertungen evaluiert werden, bevor sie breite Sichtbarkeit erlangen. Ein solches partizipatives Modell steigert die **Qualität** der angebotenen Materialien und Diskussionen, fördert eine *professionelle Lerngemeinschaft* unter den Lehrkräften und schafft Vertrauen bei Eltern. Gleichzeitig sind technische Filter und automatische Prüfungen vorgesehen, um Verstöße gegen Richtlinien (z.B. unangemessene Sprache, Datenschutzverletzungen in Inhalten) früh zu erkennen. Die Kombination aus *technischer* und *menschlicher Moderation* stellt sicher, dass die Plattform ein **sicherer, konstruktiver Raum** bleibt[14]. Qualitätssicherung wird integraler Bestandteil: z.B. mithilfe eines **Qualitätskriterien-Katalogs** (angelehnt an OER Commons Standards), der bei der Inhaltserstellung und -prüfung zur Anwendung kommt[15].

Zusammenfassend positioniert sich Breakpilot strategisch als **moderne, offene Bildungsplattform**, die die Stärken von Community-Ansätzen (Crowd-Wisdom, Offenheit) mit professionellen Anforderungen (Datenschutz, Qualität, Zuverlässigkeit) vereint. Diese Differenzierung soll Breakpilot gegenüber bestehenden Lösungen (wie proprietären Schul-Plattformen oder reinen Content-Marktplätzen) hervorheben und langfristig eine breite Nutzerbasis in der DACH-Region sichern.

## Entwicklungs-Roadmap (Strategischer Fahrplan)

Um die genannten Ziele umzusetzen, wird eine **phasenweise Einführung** vorgeschlagen. Die modulare Architektur erlaubt es, einzelne Komponenten schrittweise aufzubauen und zu integrieren. Ein möglicher Roadmap-Entwurf:

1. **Phase 1 – Kernplattform & MVP:** Aufbau der grundlegenden Infrastruktur und Launch eines **Minimum Viable Product**. In dieser Phase steht der **Lehrer-Arbeitsraum** mit Content-Zugriff im Vordergrund. Wichtige Deliverables:
2. **User-Management & Rollen:** Einrichtung des Identity-Management-Systems mit den Grundrollen (Lehrer, Creator, Eltern, Admin). Basis-Registrierung und Login (möglichst direkt mit SSO-Vorbereitung).
3. **Content-Bibliothek (MVP):** Implementierung einer ersten Version des Content-Management-Modules, ggf. mit Integration einer bestehenden OER-Bibliothek. Lehrkräfte können bereits Inhalte durchsuchen, nach Fächern/Themen filtern und Material herunterladen oder online verwenden.
4. **Creator-Studio (MVP):** Auswahl und Integration eines Open-Source-Authoring-Tools (z.B. **H5P** für interaktive Inhalte) im Creator-Bereich, sodass erste Creator oder pilotierende Lehrkräfte eigene Materialien erstellen und hochladen können.
5. **Frontend Grundgerüst:** Entwicklung einer Weboberfläche für Lehrkräfte und Creator mit Dashboard, Content-Suche und einfachen Community-Features (z.B. Kommentar- oder Bewertungsmöglichkeit zu Inhalten). UI-Design fokussiert auf Einfachheit und klare Trennung der Bereiche.

6. **Hosting & Datenschutz:** Einrichtung der Server in einer EU-Cloud, Implementierung von TLS-Verschlüsselung, Datenschutzerklärung und Einwilligungsabfragen (z.B. Cookie-Opt-In) gem. DSGVO.
7. **Interne Qualitätssicherung:** Vor dem Launch werden einige Pilot-Inhalte durch Fachexperten überprüft, Moderationsrichtlinien aufgestellt und das System auf Sicherheit getestet.
8. *Erfolgskriterium Phase 1:* Ein kleiner Nutzerkreis (Pilotlehrkräfte und Creator) kann sich anmelden, Inhalte bereitstellen und nutzen. Feedback wird gesammelt.
9. **Phase 2 – Community & Erweiterung:** In dieser Phase wird die **Eltern-Community** sowie erweiterte Funktionen eingeführt.
10. **Eltern-Community-Plattform:** Integration einer Community-Lösung, z.B. eine **weiße-Label-Instanz von Minds Network** oder alternativ ein Forum (Discourse). Eltern können sich nun registrieren, austauschen und Beiträge posten. SSO zwischen Hauptplattform und Community wird aktiviert, sodass z.B. Lehrkräfte auch an Eltern-Diskussionen teilnehmen können (falls vorgesehen, etwa in moderierter Form).
11. **SSO & Integration:** Fertigstellung der Single-Sign-On-Lösung zwischen Lehrer/Creator-System und der Community-App. Einheitliches Nutzerprofil, aber differenzierte Datenfreigaben (Eltern sehen z.B. keine internen Lehrer-Daten und umgekehrt).
12. **Moderations-Team aufbauen:** Rekrutierung erster Moderatoren aus der Community oder Projektteam, Schulung in Moderationstools und Richtlinien. Einführung von Mechanismen wie Content-Flagging (Nutzer können unangemessene Inhalte melden).
13. **Weitere Creator-Tools:** Ausbau des Creator-Studios, evtl. Integration von Tools wie **OnlyOffice** oder Etherpad für kollaborative Materialerstellung, **Nextcloud** für Datenaustausch (Selbst-Hosting von Dateien)[\[16\]](#). Creators erhalten Analytics-Dashboards, um zu sehen, wie oft ihre Inhalte abgerufen werden.
14. **Content-Kuration:** Aufbau von thematischen **Kuratoren-Teams** (erfahrene Lehrkräfte), die neu hochgeladene Inhalte sichten, verschlagworten und bei Bedarf in kuratierte Sammlungen aufnehmen. Einführung eines *Bewertungs- und Kommentarsystems* für Inhalte (Lehrkräfte können 1-5 Sterne vergeben und Feedback hinterlassen).
15. **Lizenzmanagement:** Vollständige Implementierung der Lizenzwahl beim Upload (CC-Lizenzen) und Darstellung der Lizenzinfos auf Inhaltsseiten. Anpassung der AGB, sodass klargestellt ist, unter welchen Bedingungen Inhalte genutzt werden dürfen.
16. *Erfolgskriterium Phase 2:* Deutlich wachsende Beteiligung: Mehr Creator laden Inhalte hoch, erste engagierte Eltern in der Community. Das System bewältigt moderat steigende Last und bleibt stabil. Die meisten Inhalte sind qualitativ und moderiert.

17. **Phase 3 – Monetarisierung & Skalierung:** Sobald ein ausreichender Katalog an Inhalten und aktiven Nutzern vorhanden ist, wird das Monetarisierungssystem aktiviert.
18. **Impact-Scoring-Modul:** Deployment des Impact-Auswertungssystems. Es aggregiert nun live die Nutzungsdaten: z.B. Aufrufe/Downloads pro Inhalt, Durchschnittsbewertungen, Feedback-Qualität. Aus diesen Daten berechnet es regelmäßig (z.B. monatlich) Impact-Scores je Content und je Creator.
19. **Monetarisierung:** Integration eines **Zahlungsmoduls** (z.B. Stripe API) für Ein- und Auszahlungen. Geschäftsmodell konkretisieren: entweder Schulabonnements oder Freemium-Modell (z.B. Basisnutzung frei, Premium-Funktionen gegen Gebühr) – die Einnahmen fließen in einen Creator-Pool. Das System verteilt nach definierten Regeln die Erlöse an Creator entsprechend ihrer Impact-Scores. *Wichtig:* Dieses System bleibt **ohne Krypto-Token** – alle Abrechnungen erfolgen in € über klassische Zahlungskonäle, was im Vergleich zu tokenbasierten Ansätzen deutlich compliance-freundlicher ist[5].
20. **Skalierung & Performance:** Wenn die Nutzerzahlen steigen, Hochskalieren der Infrastruktur – z.B. Umstellung auf Kubernetes-Clustering, Einrichtung von Load Balancing und Caching (z.B. mit Varnish oder Redis). Metriken-Monitoring (Prometheus/Grafana) zur Überwachung. Sicherstellen, dass auch bei höherem Aufkommen an Video/Medien die Auslieferung performant bleibt (ggf. Einsatz eines CDN).
21. **Mobile Apps:** Entwicklung einfacher **Mobile-Apps** oder PWA (Progressive Web App) für den Lehrerbereich und die Eltern-Community, um Nutzung via Smartphone zu verbessern (wichtig, da Eltern v.a. mobil online sind). Minds Network liefert ggf. schon White-Label-Apps, die gebrandet werden können[17][18].
22. **Weiterentwicklung Inhalte:** Einführung weiterer Inhaltstypen und Tools, z.B. Quiz-Generatoren, Gamification-Elemente für Schüler (falls später Schülerzugänge geplant sind) oder Schnittstellen zu Schulsoftware.
23. **Marketing & Community Growth:** Parallel Aufstockung des Supports, Community-Events (z.B. Webinare für Lehrer, „Creator-Wettbewerbe“), um Plattformwachstum zu fördern.
24. *Erfolgskriterium Phase 3:* Breakpilot generiert erste Einnahmen, Creator werden erfolgreich ausbezahlt, was deren Motivation steigert. Die Plattform hat eine stabile, skalierbare technische Basis erreicht, um weiter zu wachsen.
25. **Phase 4 – Optimierung & Langfristige Governance** (fortlaufend nach dem großen Launch):
26. **Feedback-Schleifen:** Etablieren fester Kanäle, um Feedback von Lehrern, Creatorn und Eltern einzusammeln (Community-Umfragen, Beirat mit Lehrern etc.), damit die Plattform iterativ verbessert wird.
27. **Curriculare Integration:** Zusammenarbeit mit Schulen/Behörden, um Inhalte gezielt an Lehrpläne anzupassen, evtl. Siegel oder Zertifizierungen für bestimmte Inhalte (Qualitätssiegel).

28. **Internationalisierung:** Gegebenenfalls Übertrag des Konzepts auf andere Sprachen/Regionen, sofern strategisch gewünscht, beginnend mit minor Adaptationen (z.B. Südtirol oder EU-Englisch).
29. **Fortlaufende Sicherheits- und Datenschutz-Audits:** Regelmäßige Prüfungen und Updates, um neue DSGVO-Anforderungen oder Sicherheitsbedrohungen zu adressieren (z.B. Penetrationstests, Notfallpläne).
30. **Nachhaltigkeit:** Sicherstellung der finanziellen Nachhaltigkeit (ggf. zusätzliche Förderer akquirieren, Premium-Features definieren) und der Community-Pflege (Moderatorenprogramm, Creator-Schulungen etc.).

Diese Roadmap ist flexibel zu handhaben – einzelne Schritte können parallel verlaufen (z.B. schon früh mit Moderatoren arbeiten) oder je nach Ressourcen angepasst werden. Wichtig ist aber die **Priorisierung**: erst eine solide Kernplattform schaffen, dann Community und Funktionen ausbauen, schließlich Monetarisierung hinzufügen. So kann Breakpilot organisch wachsen und Risiken minimieren.

## Technische Architekturübersicht

### Gesamtarchitektur und Module

Die vorgeschlagene Systemarchitektur von Breakpilot folgt einem **modularen, service-orientierten Ansatz** (angelehnt an Microservices-Prinzipien), um **Skalierbarkeit** und **Entkopplung** zu erreichen. Statt eines Monolithen wird das System in mehrere **Hauptmodule** bzw. Subsysteme unterteilt, die über klar definierte Schnittstellen (APIs) miteinander kommunizieren. Diese Module entsprechen weitgehend den funktionalen Bereichen der Plattform:

- **1. Benutzer- und Zugriffsmanagement (Identity & Access Service):** Zentrale Verwaltung aller Benutzerkonten, Authentifizierung (Login/SSO) und Autorisierung (Rollenzuweisung, Berechtigungen). Dieser Service stellt sicher, dass z.B. nur Lehrkräfte auf den Lehrerbereich zugreifen können, Creator auf ihre Tools etc. und implementiert das **Role-Based Access Control (RBAC)** Modell[4]. Ein Open-Source-Tool wie **Keycloak** kann hier eingesetzt werden, da es OIDC/OAuth2-fähig ist und SSO, Rollen sowie Benutzergruppen out-of-the-box unterstützt.
- **2. Content-Management-System (CMS) & Repository:** Kernstück für die Verwaltung aller Bildungsinhalte (Videos, Arbeitsblätter, Interaktive Übungen etc.). Beinhaltet eine Inhaltsdatenbank (inkl. Metadaten wie Titel, Beschreibung, Schlagworte, Altersstufe, Lizenztyp, Creator-Zuordnung) sowie einen **Dateispeicher** für die Inhalte selbst. Das CMS bietet APIs für CRUD-Operationen auf Inhalte, Suchen/Filtern und liefert Inhalte an das Frontend aus. Hier könnte z.B. ein **Headless CMS** (wie **Strapi** oder **Directus**) im Hintergrund laufen, um Inhalte zu organisieren, oder es wird custom entwickelt mit z.B. Node.js/Python und einer PostgreSQL-Datenbank. Wichtig: Die **Lizenzinformationen** jedes Inhalts werden im CMS hinterlegt und bei Anfragen mitgeliefert, sodass Frontends die CC-Lizenz

deutlich anzeigen (inkl. Icons/Links). Ebenso werden **Versionierungen** von Inhalten unterstützt, falls Creator Updates vornehmen.

- **3. Creator-Studio Service:** Ein Modul bzw. Frontend-Komponente, die speziell für Content-Ersteller entworfen ist. Es bietet eine **Web-Oberfläche zum Hochladen und Bearbeiten** von Inhalten. Technisch greift es auf das CMS zu (über dessen API), verfügt aber zusätzlich über Integrationen zu speziellen Tools: z.B. Einbettung eines **H5P-Editors** für interaktive Übungen, Integration von **OnlyOffice/Collabora** für gemeinsames Bearbeiten von Dokumenten oder Präsentationen innerhalb des Browsers[19], und gegebenenfalls Schnittstellen zu Bild-/Videobearbeitung (für Videos z.B. Link zu einem Cloud-Videoeditor oder zumindest Upload/Transcoding via FFmpeg im Backend). Das Creator-Studio stellt auch **Analytics für Creator** bereit: es kann vom Analytics-Modul (siehe 6) Daten abrufen und dem Creator anzeigen, wie oft seine Inhalte genutzt wurden, welches Feedback/Rating sie erhielten usw. So haben Creator ein eigenes Dashboard zur Erfolgskontrolle ihrer Werke.
- **4. Lehrer-Arbeitsraum Service:** Dies ist das Haupt-Frontend für Lehrkräfte. Hier können Lehrende Inhalte **recherchieren, planen** und für ihren Unterricht zusammenstellen. Hauptfunktionen:
  - **Content-Suche und -Katalog:** Ein komfortables UI, um die Inhalte aus dem CMS zu durchsuchen (Filter nach Fach, Klassenstufe, Dauer, etc.). Mögliche Implementierung: eine Such-Engine wie **Elasticsearch** indexiert die Inhaltsmetadaten, das Frontend stellt Echtzeit-Filtersuchen bereit.
  - **Content Player/Viewer:** Möglichkeit, Inhalte direkt in der Plattform anzusehen (Video-Player für Videos, PDF-Viewer für Dokumente, H5P-Player für interaktive Übungen). Alternativ Download-Optionen, falls offline genutzt.
  - **Kollegiums-Funktionen:** Evtl. ein Bereich, wo Lehrer sich untereinander austauschen – z.B. Foren oder Gruppen (z.B. „Meine Schule“-Gruppe oder thematische Gruppen). Dies könnte perspektivisch über die Integration der Community-Funktion erfolgen, ist aber im Lehrerbereich vielleicht separiert (ein Forum nur für Lehrkräfte). Hier könnte **Rocket.Chat** oder **Discourse** als internes Lehrerforum dienen, falls benötigt.
  - **Playlist/Session-Planer:** Eine Funktion, um mehrere Inhalte zu einer Session zusammenzustellen (z.B. 5 Inhalte für 5 Pausen der Woche) – mit Timeline oder Kalenderintegration. Dies würde auf dem Frontend die Inhalts-APIs nutzen und z.B. dem Nutzer ermöglichen, eine Liste zu speichern.
  - **Feedback & Bewertung:** Lehrer können Inhalte bewerten (Sterne, Kommentar). Diese Interaktion wird vom Lehrer-Frontend bereitgestellt, aber die Daten fließen ins CMS/Analytics zurück.
  - **Zugriff auf Community (lesend):** Wenn relevant, könnte Lehrern z.B. ein Feed aus der Eltern-Community angezeigt werden, etwa Fragen von Eltern oder Erfolge, die geteilt werden – allerdings muss man hier datenschutztechnisch vorsichtig sein. Alternativ bleibt die Eltern-Community völlig getrennt – das wird weiter unten

erläutert. Im Minimalfall kann im Lehrerbereich ein Link zur Eltern-Community existieren, der in die separate App führt, um Kontextwechsel bewusst zu machen.

- **5. Eltern-Community Plattform:** Technisch und aus Nutzersicht ein eigenständiges Modul. Hierfür wird eine **Open-Source-Community-Software** eingesetzt, um Rad neu zu erfinden zu vermeiden. Die Anfrage nennt explizit *Minds Network* als Option – Minds ist ein verteiltes soziales Netzwerk, das als White-Label-Lösung Communities hosten kann[20][21]. Alternativ käme ein klassisches Foren-/Gruppensystem in Frage (z.B. **Discourse** oder **Flarum** für Foren; oder **Mastodon**/ActivityPub für Social Feed). Minds bietet allerdings bereits Features wie Gruppen, Chats, Push Notifications, Rollen und sogar Mitgliedsbeiträge[22], was gut zum Monetarisierungskonzept passen könnte (z.B. exklusive Gruppen für bestimmte Mitglieder). In unserem Entwurf nehmen wir an, Breakpilot nutzt eine **eigene Minds-Instanz**: Diese läuft entweder separat gehostet (Open-Source-Code von Minds) oder als Cloud-Dienst von Minds Networks. Die Eltern-Community ist über **APIs** lose mit dem Kernsystem verbunden. Konkret bedeutet das:
  - Einheitliches SSO: Eltern, die sich auf der Community registrieren, werden auch in der Hauptbenutzerverwaltung bekannt gemacht (und umgekehrt), damit man nicht zwei Logins hat. Keycloak könnte als zentraler IdP fungieren, während Minds über OAuth/OpenID Connect angebunden wird[23].
  - Datenaustausch: Im Grunde operiert die Eltern-Community autonom – Eltern posten Beiträge, kommentieren, erstellen Gruppen. Lehrkräfte und Creator können sich (so sie möchten) dort ebenfalls anmelden und ihre Perspektive einbringen. Eine mögliche Integration: Bei neuen veröffentlichten Inhalten könnte automatisch in der Eltern-Community ein Post generiert werden („Neuer Inhalt XY jetzt verfügbar – diskutiert hier!“), um Eltern auf dem Laufenden zu halten. Dies ließe sich über die Minds-API und Webhooks vom CMS realisieren.
  - Moderation: Die Moderation der Community erfolgt teils durch Community Manager von Breakpilot, teils durch Moderationsrollen in Minds selbst. Minds ist Open Source, was wichtig ist, damit wir unsere strengeren Moderationsrichtlinien umsetzen können – z.B. im Gegensatz zur Standard-Minds-Philosophie (die sehr freie Redefähigkeit betont[24]) werden wir hier *konsequent alle extremen oder bildungsfremden Inhalte unterbinden*. Das heißt, Einstellungen und ggf. Code-Anpassungen sorgen dafür, dass Content-Filter aktiv sind und Moderatoren Beiträge löschen oder User sperren können, wenn diese gegen die Richtlinien verstoßen.
  - Inhalte in Community: Eltern könnten z.B. Fotos/Videos aus dem Alltag ihrer Kinder teilen oder Fragen stellen („Hat jemand Tipp für Bewegungsspiel zu Thema X?“). Diese UGC (User-generated Content) werden in der Community-DB gespeichert, nicht im Breakpilot-CMS – außer es besteht Anlass, sie als Lerninhalt zu übernehmen. Man könnte perspektivisch besonders gute Eltern-Beiträge (z.B. ein Spiel aus dem Familienleben) an Creator zur Aufbereitung weitergeben – dies aber nur mit Einverständnis und als Option.

- **6. Analytics- und Impact-Scoring Service:** Ein separater Backend-Service, der sämtliche **Nutzungsdaten und Metriken** sammelt und auswertet. Hier laufen Events zusammen: jede Inhaltsnutzung (Play, Download), jede Bewertung/Kommentar, Teilnahme der Eltern etc. Diese Daten werden in einer Analytics-Datenbank gespeichert (z.B. einer NoSQL-DB oder einer relationalen DB, je nach Bedarf). Darauf aufbauend berechnet der Service in regelmäßigen Intervallen den **Impact-Score** für Inhalte und Creator. Algorithmen (anfangs fest kodiert, später evtl. Machine-Learning-unterstützt) gewichten die verschiedenen Faktoren. Beispielsweise:  $Score = Anzahl\ Lehrernutzungen \times 2 + Anzahl\ positiver\ Bewertungen \times 1 + Eltern-Feedbacks \times 1$  etc.. Die Ergebnisse speist der Service zurück ins System:
  - An das Creator-Studio (zur Anzeige im Creator-Dashboard).
  - An das Monetarisierungsmodul für die Auszahlung.
  - Optional an das Frontend für Lehrer als Sortierkriterium („zeige beliebteste Inhalte zuerst“).
  - Der Service kann auch **Berichte** erzeugen, z.B. Monatsberichte über Plattformnutzung, die für interne Zwecke oder Community-Kommunikation genutzt werden („State of Breakpilot“).
  - Datenschutz: Wichtig ist, dass Analytics aggregiert und keine personenbezogenen Profile erstellt (also keine heimliche Überwachung einzelner Lehrer). Nur globale und contentbezogene Auswertung. Persönliche Daten werden entweder gar nicht erfasst oder anonymisiert/pseudonymisiert, um DSGVO zu genügen (z.B. interne Nutzer-IDs statt Klarnamen in Logs).
- **7. Monetarisierungs- und Zahlungsmodul:** Dieser Baustein kümmert sich um die **finanziellen Transaktionen**. Er besteht einerseits aus einem **Backend-Service**, der die Verknüpfung zu Zahlungsanbietern (Stripe, PayPal o.ä.) herstellt, andererseits aus Frontend-Komponenten im Creator- und Admin-Bereich:
  - **Einnahmen verwalten:** Falls Schulen oder Nutzer zahlen (z.B. Abo), verarbeitet das Modul diese Zahlungen automatisiert, erstellt Rechnungen und verbucht Beträge in einen Einnahmentopf.
  - **Auszahlungen an Creator:** Basierend auf den Impact-Scores aus dem Analytics-Service berechnet es z.B. monatlich, welcher Creator welchen Anteil am Topf erhält. Die Auszahlung kann automatisiert via Zahlungsanbieter erfolgen (dazu müssen Creator eine Auszahlungsmethode hinterlegt haben). Alternativ stellt das Modul Berichte bereit, und ein Administrator stößt manuell Zahlungen an (je nach Skalierung).
  - **Kein Token, kein Wallet:** Es wird bewusst *keine* eigene Kryptowährung oder Token-Wallet eingeführt, sondern auf klassische Währungen gesetzt. Damit bleibt das System **übersichtlich und rechtlich unproblematisch** – tokenlose Modelle gelten als einfacher konform zu Regulierungen[5].

- **Mitgliedschaften (Community):** Sollte Breakpilot neben Impact-Payouts weitere Monetarisierung wollen, z.B. Premium-Mitgliedschaften für erweiterten Zugang, kann das Modul dies ebenfalls abwickeln. Interessanterweise bietet Minds Networks selbst z.B. die Möglichkeit, **Mitgliedschaftsstufen** und bezahlte Abonnements für Communitys einzurichten[21][25]. Diese Funktion könnte man für die Eltern-Community nutzen: z.B. ein freiwilliger Förderbeitrag von Eltern für exklusive Inhalte/Gruppen, der dann auch in den Content-Fördertopf fließt.
- **8. Administration & Governance-Tools:** Schließlich ein interner Bereich für die Plattformbetreiber (Admins, Moderatoren, Qualitätssicherer). Dieser besteht aus:
  - **Admin-Backend:** z.B. direkt via Keycloak-Admin-GUI fürs Benutzermanagement (Anlegen/Bearbeiten von Benutzerrollen), evtl. einem Admin-Frontend im CMS für Inhaltsübersicht (wo man notfalls Inhalte löschen oder sperren kann).
  - **Moderationskonsole:** Ein Dashboard, das alle gemeldeten Inhalte/Beiträge anzeigt, Werkzeuge zum Sperren/Entfernen bietet und Logging für Moderationsaktionen. Für die Eltern-Community wird entweder das Minds-eigene Moderationsinterface verwendet (Admins können dort Inhalte entfernen, Nutzer verwarnen etc.), oder eine konsolidierte Ansicht, die Infos von Minds und vom CMS (für Lehrinhalte) zusammenführt.
  - **Analytics für Admins:** Ein Super-Dashboard mit Kennzahlen zur Nutzung, zur Systemleistung (vielleicht integriert via Grafana) und zu Community Health (Anzahl aktiver User, Moderationsfälle etc.).
  - **Konfiguration:** Möglichkeiten, systemweite Einstellungen vorzunehmen (Lizenzoptionen, Kategorien für Inhalte, etc.). Diese könnten über einfache Config-Files oder ein UI angepasst werden.

Die obige Auflistung zeigt, dass Breakpilot aus **mehreren gekoppelten Systemen** besteht. Die Kommunikation zwischen den Komponenten erfolgt hauptsächlich über **HTTP-basierte APIs** (REST/JSON oder GraphQL, wo passend) sowie teils asynchron über ein **Nachrichtensystem** (z.B. RabbitMQ oder Apache Kafka), insbesondere um Events (wie „Content X wurde heruntergeladen“) zuverlässig an den Analytics-Service weiterzuleiten, ohne den Benutzerfluss zu blockieren.

Ein **API-Gateway** oder *Reverse Proxy* (wie **NGINX** oder **Traefik**) steht typischerweise vor den Services, um einen einheitlichen Endpunkt zu bieten, Load Balancing zu unterstützen und SSL/TLS zu terminieren. Darüber hinaus kann ein Gateway auch benutzt werden, um zentrale Authentifizierungslogik (JWT-Token-Überprüfung via Keycloak) voranzustellen, sodass die Microservices entlastet werden.

Zusammengefasst entsteht folgender **Architektur-Stack**: - **Frontend:** Moderne Webtechnologie (z.B. React oder Angular) für Lehrer- und Creator-UI; separate Community-App (Minds, evtl. React Native für mobil). Responsive Design für Tablet/Smartphone-Nutzung. - **Backend:** Mischung aus **Microservices** (z.B. Node.js/NestJS oder Spring Boot für verschiedene Services). Nutzung verschiedener Technologien je nach Eignung (z.B. Python für Analytics, Node.js für Realtime-Features). Daten werden in **Open-Source-**

**Datenbanken** gespeichert (PostgreSQL für relationale Daten; MongoDB/CouchDB für ggf. Dokumentenablage, Redis für Caching Sessions wie z.B. JWT-Blacklist oder Rate Limiting). - **Infrastructure/Hosting:** Containerbasierte Deployment (Docker) orchestriert via **Kubernetes** (Open-Source-Orchestrierung) für Skalierbarkeit. Cloud-agnostisch aber ideal auf EU-Cloud (z.B. IONOS, Scalingo, oder AWS/Azure mit Region Frankfurt/Dublin – wobei US-Provider wegen Cloud Act Problematik abzuwägen sind[26]). Files werden in einem **objektbasierten Storage** abgelegt (z.B. ein S3-kompatibler Dienst in EU, oder selbstgehostetes **MinIO**). - **Security & Privacy:** TLS-Verschlüsselung überall, feingranulares RBAC, Logging und Monitoring in Übereinstimmung mit DSGVO (Pseudonymisierung wo möglich, keine Speicherung von IPs oder persönlichen Identifikatoren länger als nötig). Automatisierte Backups der Datenbanken (verschlüsselt) und Notfallwiederherstellungskonzepte, um Ausfälle zu überstehen.

Um das Konzept zu verdeutlichen, hier eine schematische Darstellung (textuell) der **Kommunikation:** Wenn ein Lehrer im Frontend nach einem Inhalt sucht, ruft das Frontend die **Content-API** des CMS auf. Der Request geht durchs Gateway, Auth-Token wird geprüft, dann liefert das CMS die Treffer zurück. Der Lehrer startet nun ein Video – das Video wird vom File Storage (über CDN oder direkt) gestreamt, und parallel schickt das Frontend ein „Content genutzt“-Ereignis an den Analytics-Service (entweder direkt via API oder eventbasiert). Der Analytics-Service loggt dies und erhöht temporär Zähler. Am Monatsende wertet er alle Logs aus und errechnet Score X für den Creator. Das Monetarisierungsmodul holt sich diesen Wert und erstellt eine Auszahlung via Stripe an den Creator. In der Eltern-Community bekommt eventuell der entsprechende Content-Thread eine automatische Nachricht „Lehrer Y hat dieses Material heute verwendet – wie fanden es die Schüler?“. Ein Elternteil könnte darauf antworten; sollte diese Antwort gegen Richtlinien verstoßen, sieht ein Moderator es im Moderationsdashboard und kann eingreifen (z.B. entfernen). Gleichzeitig greifen DSGVO-Mechanismen: Der Lehrer könnte in seinem Profil z.B. einstellen, dass seine Nutzungsaktivitäten nicht öffentlich geteilt werden – dann würde obige automatische Nachricht unterbleiben. Dieses Beispiel zeigt, wie die Module zusammenspielen, aber auch wie **Governance-Regeln** (Datenschutzpräferenzen, Moderationsrichtlinien) als Schicht darüberliegen.

Im Folgenden werden einige dieser wichtigen Aspekte – Open-Source-Technologieeinsatz, Trennung der Teilbereiche, Monetarisierung, Lizenzierung, Rollen/Rechte, Datenschutz, Moderation/QS – noch einmal gezielt beleuchtet.

## Open-Source-Komponenten für Frontend, Backend und Hosting

Der Einsatz von Open-Source-Komponenten ist ein Kernanliegen von Breakpilot und durchzieht die gesamte Architektur. Dies bietet nicht nur Kostenvorteile, sondern fördert auch Anpassbarkeit und Sicherheit durch Einsicht in den Quellcode. Nach dem Prinzip „Invented Here Syndrome vermeiden“ soll wo immer möglich auf **bewährte OSS-Bausteine** zurückgegriffen werden, anstatt alles von Grund auf neu zu entwickeln.

**Frontend:** Für die Web-Frontends (Lehrer und Creator) empfiehlt sich ein Framework wie **React** (MIT-Lizenz) oder **Angular** (GPL), ergänzt um Open-Source-UI-Bibliotheken (z.B. **Material-UI** für React, um ein modernes, barrierearmes Design zu gewährleisten). Auch

**Vue.js** wäre eine Option. Wichtig ist eine komponentenbasierte Architektur, die modulare Entwicklung ermöglicht. Da Echtzeit-Interaktionen minimal sind (hauptsächlich Lesen von Inhalten, Abgeben von Feedback), reicht klassische REST-Kommunikation; bei Bedarf kann **WebSocket** (z.B. über Socket.IO) für Live-Features nachgerüstet werden (etwa wenn Chat unter Lehrern implementiert wird). Für die Eltern-Community, falls Minds gewählt wird: diese bringt eigene Frontends mit (Web und Mobile). Minds ist in **Angular/TypeScript** entwickelt[27] und stellt Android/iOS-Apps zur Verfügung. Diese können mit Breakpilot-Branding versehen werden[18]. Sollte eine Alternative genutzt werden (z.B. Discourse für Foren), gibt es ebenfalls Web- und mobile Apps (Discourse hat z.B. React Native basierte Apps). Somit kann Breakpilot für die Community auch eine **eigene App** herausbringen, ohne von Null zu beginnen.

**Backend & Frameworks:** Die einzelnen Services können in verschiedenen Sprachen realisiert sein, was dank klarer API-Abgrenzung möglich ist (**Polyglot-Architektur**). Einige Vorschläge: - **Identity Service:** Hier könnte **Keycloak** als komplette Lösung dienen. Keycloak (Red Hat) ist Open Source und implementiert alle Standardflows (Login, SSO, User Federation, 2FA, LDAP-Anbindung etc.). Dadurch spart man Entwicklungszeit. Keycloak kann als Docker-Container deployt und via Admin-UI konfiguriert werden. Die Benutzerverwaltung zentralisiert sich damit. - **Content/CMS Service:** Option A: Einsatz eines Headless CMS wie **Strapi** (Node.js, MIT-Lizenz) oder **Directus** (GNU). Diese liefern sofort APIs für Collections, Auth etc., müssten aber ggf. um Features wie den Impact-Score oder spezifische Lizenz-Handling ergänzt werden. Option B: Maßgeschneiderte Implementierung, z.B. mit **NestJS** (ein aufgeräumtes Node.js-Framework) oder mit Python (**Django REST Framework**). Da Relationen (Creator-zu-Inhalt, Inhalte zu Kategorien) wichtig sind, bietet sich relationales DB-Schema an; **PostgreSQL** als Open-Source-Datenbank mit starkem Ökosystem ist ideal. - **Creator & Teacher Services:** Diese sind eher dünne Schichten, oft vor allem Frontend, die Backend-Funktionalität nutzen. Dennoch könnten z.B. spezielle Funktionen wie „exportiere meine Session als PDF“ als Microservice ausgelagert werden (z.B. in Python with reportlab, falls nötig). Grundsätzlich ist aber viel Logik clientseitig möglich, sodass hier wenig separate Backendservices nötig sind. - **Community:** Falls Minds, wird der Minds-Server (PHP + MySQL) betrieben. Alternative Forensoftware (Discourse) basiert auf Ruby. Unabhängig davon sollte die Community-Software *containerisiert* und in die DevOps-Pipeline integriert sein, um konsistente Deployments zu gewährleisten. - **Analytics Service:** Für Metrikenverarbeitung bietet sich z.B. **Python** (pandas/numpy) an, falls komplexere Berechnungen, oder **Golang** für performante Aggregation. Ein event-getriebener Ansatz mit Kafka-Consumer wäre robust. Die resultierenden Score-Daten können in einer kleinen relationalen Tabelle gespeichert werden (Creator-ID, Monat, Score, Auszahlungsbetrag). - **Payment Integration:** Hier gibt es Bibliotheken in allen Sprachen für Stripe/PayPal. Node.js mit Stripe SDK wäre unkompliziert. Wichtig ist keine sensiblen Zahlungsdaten selbst zu speichern, sondern Tokens der Anbieter zu nutzen (DSGVO/Bafin-Compliance). - **Search:** Sollte die Suche über Volltext gehen (z.B. in Beschreibungen), ist **Elasticsearch** (Apache 2.0 Lizenz) geeignet. Dieses System wäre ein separater Dienst, der regelmäßig vom CMS mit Daten versorgt wird (entweder per Event bei Content-Änderung oder Batch-Index). - **File Storage:** Evtl. Betrieb eines **Nextcloud** für Files? Nextcloud liesse sich als Lehrer-Dateiablage nutzen, aber für

die Inhalte-Auslieferung reicht vermutlich ein einfacher Objektspeicher. Man könnte Nextcloud für intern (z.B. Kollaboration in Lehrerteams) in Phase 4 noch ergänzen. In Phase 1-3 ist es vermutlich Overkill. Stattdessen: **MinIO** als Open-Source-S3-Storage im eigenen Cluster – darüber könnten Videos, PDFs etc. ausgeliefert werden. Alternativ nutzt man Cloud-Speicher (was dann nicht rein OSS ist, aber pragmatisch sein kann). - **DevOps & Infrastruktur: Kubernetes** (Open Source, CNCF) als Orchestrator, mit **Helm** für Deployments. CI/CD z.B. mit **Jenkins** oder **GitHub Actions** (wobei letzteres SaaS ist). Monitoring via **Prometheus** (OSS) + **Grafana** (OSS) zur Visualisierung. Logging via **ELK Stack** (Elasticsearch, Logstash, Kibana) oder moderner **EFK** (Fluentd statt Logstash). Für Benachrichtigungen: Integration eines Open-Source-Mailing-Servers (z.B. **Postal** oder einfach per SMTP-Relay). - **AI-Features** (optional, in Zukunft): Da viel von KI in Bildung die Rede ist, könnte Breakpilot mit Open-Source-LLMs (z.B. lokal gehosteter Language Model für automatische Tagging von Inhalten oder KI-gestützte Feedbackanalyse) experimentieren. Das wäre allerdings zusätzlich und nicht Kern der initialen Architektur.

All diese Komponenten sind **untereinander austauschbar** dank standardisierter Schnittstellen. Sollte z.B. Minds doch nicht passen (etwa wegen moderations-technischer Bedenken), könnte man auf eine Alternative wie **Discourse** (ebenfalls open source) wechseln, die Integration via SSO und API aber ähnlich gestalten.

Insgesamt resultiert eine **Best-of-Breed Architektur aus OSS**, die hohe Flexibilität bietet. Wichtig ist, dass alle verwendeten Open-Source-Komponenten auch auf **aktive Communities** und regelmäßige Updates zurückgreifen können, damit Sicherheitspatches schnell eingespielt werden können. Durch die modulare Auswahl kann Breakpilot vom Know-how der Open-Source-Community profitieren – z.B. bewährt sich Rocket.Chat in Hochschulen für Chat[28], BigBlueButton für Webkonferenzen in virtuellen Klassenzimmern[29] (falls Breakpilot sowas je benötigt), Nextcloud für DSGVO-konforme Dateiablage[16] etc. Diese Tools lassen sich an vielen Stellen andocken.

## Integration der Teilbereiche: Lehrer-Arbeitsraum, Creator-Studio und Eltern-Community

Die **klare Trennung** der drei Nutzerbereiche ist nicht nur in der UI sichtbar, sondern auch technisch umgesetzt. Jeder Bereich hat eigene Komponenten (siehe Module oben) und teils separate Subdomains oder Apps: - Lehrkräfte-Bereich: z.B. erreichbar unter `teachers.breakpilot.de` (Web) und evtl. in Zukunft als „Breakpilot Lehrer“ App. - Creator-Studio: z.B. `studio.breakpilot.de` oder integriert in dieselbe Webanwendung wie Lehrer, aber mit Feature-Toggles nach Rolle (je nachdem, ob viele Lehrer auch Creator sind – was möglich ist. Möglicherweise hat man eher einen kombinierten „Professional“-Bereich und unterscheidet innerhalb der Funktionen). - Eltern-Community: z.B. `community.breakpilot.de` oder eigenständige Domain/Apps.

Diese logische Trennung ermöglicht es, **unabhängig zu skalieren**: Sollte z.B. die Eltern-Community plötzlich riesigen Zulauf bekommen, kann man den Community-Server getrennt hochfahren, ohne die Kernplattform zu gefährden. Ebenso können

sicherheitskritische Teile (Lehrer/CMS) isoliert werden – sie stehen z.B. hinter einem VPN, während die Community öffentlicher ist.

Trotzdem braucht es **Kohärenz**: - **Single Sign-On (SSO)**: Wie erwähnt, dient ein zentrales Identity-Management der nahtlosen Nutzung. Ein Lehrer, der z.B. auch selbst Elternteil ist und die Community nutzen will, soll nicht zwei Accounts benötigen. Daher wird im Identity-Service Multi-Roles unterstützt (eine Person kann mehrere Rollen haben). Beim Login wählt der Nutzer ggf., in welcher Rolle er gerade agiert (oder die UI unterscheidet automatisch per Kontext). SSO basiert auf offenen Standards (OpenID Connect/OAuth2). Minds z.B. unterstützt SSO-Integration[23], und andere Tools wie Discourse ebenfalls. Somit wird das **Benutzerprofil** und die **Authentifizierung** zentral gehalten. - **APIs und Datenflüsse**: Bestimmte Funktionen erfordern Datenfluss zwischen den Bereichen: - Wenn ein Creator einen Inhalt veröffentlicht (im Studio), soll dieser sofort im Lehrer-Katalog erscheinen. Lösung: Der CMS-Service indexiert den neuen Inhalt und er ist via API für Lehrer-Frontend abrufbar. Gegebenenfalls wird auch eine *Benachrichtigung* an interessierte Lehrer generiert (z.B. „neuer Inhalt in deiner abonnierten Kategorie“ – per E-Mail oder in-App-Notification). Solche Nachrichten könnte man auch in die Community posten, wie zuvor erwähnt. - Wenn ein Lehrer einen Inhalt als „besonders hilfreich“ markiert, könnte diese Info dem Creator zurückgemeldet werden (Analytics) und auch Eltern gezeigt werden („Inhalt X wurde von vielen Lehrern in KW 10 genutzt“ – generiert Interesse bei Eltern). Allerdings vorsichtig: Nicht zu viele interne Daten nach außen geben. - **Datenschutz-Trennung**: Per Design sollten *personenbezogene Daten* nur im Identity-Service liegen. Weder Community noch CMS speichert Klarnamen, Adressen o.ä., sondern nur anonyme IDs und evtl. Pseudonyme. So können die Systeme interagieren, ohne volle Daten auszutauschen. Beispiel: Die Community weiß nichts über den Schulbezug eines Elternteils, und das Lehrersystem weiß keinen Klarnamen von Eltern – sie interagieren nur über Foren-Posts, die eh öffentlich innerhalb der Community sind. - **Entkoppelte Entwicklung**: Die Teilbereiche können auch getrennt weiterentwickelt werden. Beispielsweise könnte man das Creator-Studio erweitern oder austauschen, ohne die Lehrerfunktionen zu beeinträchtigen, solange die Content-API stabil bleibt.

Gerade weil Lehrer-Arbeitsraum und Eltern-Community unterschiedliche Zielgruppen haben, ist diese **saubere Trennung** wichtig. Didaktisch und organisatorisch soll es kaum Überschneidungen geben – z.B. Eltern sehen nicht das Lehrer-Interface mit all den Material-Details, und Lehrer bekommen im Arbeitsraum keine störenden Eltern-Diskussionen angezeigt. Allerdings gibt es Berührungspunkte über **moderierte Schnittstellen**: z.B. könnte Breakpilot einen Bereich „Elternfragen an den Experten“ anbieten, wo Eltern pädagogische Fragen stellen und ausgewählte Lehrkräfte oder Creator antworten. So etwas ließe sich realisieren, indem z.B. im Eltern-Community-Frontend ein spezielles Frageformular implementiert wird, dessen Eingaben ins Lehrer-System gelangen (z.B. als Ticket oder in ein Lehrerforum). Ein Moderator könnte dann anonymisiert die Antworten zurückspielen. Solche Cross-Over-Features sind denkbar, aber nur mit sorgfältigem Design. In der Architektur lassen sie sich durch definierte APIs (z.B. Community-Service ruft eine „askTeacher“ API im Lehrer-Service auf) umsetzen, was wir aber erst nach Etablierung der Grundfunktionen angehen würden.

Zusammenfassend: Die Integration der Teilbereiche erfolgt **lose gekoppelt**, mit Single Sign-On als verbindendem Element und wohldefinierten Datenflüssen für bestimmte Funktionen. Dadurch bleiben die Bereiche autonom und anpassbar, erfüllen aber gemeinsam das Plattformziel.

## Monetarisierungsmodul und Impact-Scoring-System

Das Monetarisierungsmodell von Breakpilot basiert – wie beschrieben – auf einem **nicht-tokenisierten Impact-Scoring-System**. Hier wird nun technisch erläutert, wie dieses System implementiert wird und funktioniert, sowie wie es sich vom (tokenisierten) Standard einiger Communities unterscheidet.

**Impact-Score Berechnung:** Der entsprechende Analytics-Service sammelt kontinuierlich Daten. Konkret werden folgende Event-Typen berücksichtigt (anpassbar): - *Nutzungsereignisse:* Jedes Mal, wenn ein registrierter Lehrer einen Inhalt in seinem Arbeitsraum öffnet, abspielt oder herunterlädt, wird ein Event „Content XYZ genutzt von User U“ erzeugt. Idealerweise enthält dieses Event auch Kontext, z.B. „im Unterricht verwendet“ (wenn Lehrer dies kenntlich machen) oder nur „durchgelesen“. - *Feedbackereignisse:* Wenn ein Lehrer einen Inhalt bewertet (Sterne gibt) oder einen Kommentar abgibt, wird dies erfasst („Content XYZ Bewertung 5 von User U“). - *Weiterempfehlung:* Falls es die Möglichkeit gibt, Inhalte zu „favorisieren“ oder Kollegen zu empfehlen, zählen solche Aktionen positiv. - *Community-Resonanz:* Falls Elternbeiträge mit bestimmten Inhalten verknüpft sind (z.B. „Hat jemand die Übung XY auch daheim probiert?“), könnte auch gemessen werden, wie viel Positives darüber berichtet wird. Das ist schwer quantitativ, aber man könnte einfach die Anzahl an Threads/Posts zu einem Inhalt zählen. - *Administratives Qualitätsurteil:* Sollten Content-Moderatoren einen Inhalt zertifizieren (z.B. „Lehrplan-kompatibel“ Label), könnte das dem Score einen Bonus geben.

All diese Daten laufen als Logs in den Analytics-Service. Er kann die Rohdaten in einer Datenbank vorhalten, ggf. verdichtet nach Tag/Woche. Ein geplanter Prozess (z.B. Cronjob monatlich) iteriert dann über alle Inhalte und berechnet pro Inhalt einen numerischen Score. Beispielformel (fiktiv):

$$\begin{aligned} \text{Score} = & 10 * (\text{Anzahl einzigartige Lehrer, die Inhalt genutzt haben}) \\ & + 5 * (\text{Anzahl Gesamt-Nutzungen}) \\ & + 20 * (\text{durchschnittliche Bewertungssterne}) \\ & + 5 * (\text{Anzahl Bewertungen}) \\ & + 2 * (\text{Anzahl Kommentare}) \\ & + 5 * (\text{Moderator-Qualitätspunkt, 1 oder } 0) \end{aligned}$$

Dieses Schema kann mit der Community abgestimmt sein. Wichtig ist, es **transparent** zu machen (Creators wissen, was sich wie auswirkt). Man könnte sogar den Code oder die Formel offenlegen oder gemeinsam beschließen, um Fairness sicherzustellen – ein Governance-Aspekt.

**Nicht-tokenisierte Belohnung:** Anders als z.B. bei Minds, wo Engagement in Form von Kryptowährungs-Tokens entlohnt wird<sup>[30][6]</sup>, bleibt Breakpilot beim traditionellen Geld. Das heißt, der Impact-Score dient als **Verteilungsschlüssel** für reale Währung, nicht als

handelbarer Vermögenswert selbst. Dieses Vorgehen hat mehrere Vorteile: - **Rechtliche Einfachheit:** Keine Wertpapier- oder Finanzmarktfragen, kein KYC für Token-Wallets etc., stattdessen normale Buchhaltung. Das wurde bereits beim Launchpad-Vergleich hervorgehoben: *Nicht-Token-Plattformen sind oft einfacher und compliance-freundlicher*[5]. - **Weniger Missbrauchsgefahr:** Kein Spekulationsanreiz, kein Pump-and-dump von Token, keine Bot-Armeen, die Tokens farmen. - **Klarheit für Nutzer:** Lehrer und Creator müssen keine Crypto-Wallets verstehen, sondern sehen einfach in Euro, was Sache ist. - **Steuerbarkeit:** Das Plattformteam kann Kriterien anpassen, ohne Blockchain-Abhängigkeiten. Bei Bedarf kann man extreme Fälle (z.B. Score-Manipulation durch Eigenklicks) manuell korrigieren.

**Auszahlung und Verwaltung:** Angenommen, im Monat Dezember stehen 1000 € zur Ausschüttung an Creator bereit (aus Einnahmen). Der Impact-Score jedes Creators wird ermittelt, indem man die Scores seiner einzelnen Inhalte summiert (oder falls Creator auch nach anderen Parametern bewertet werden, kann man auch direkt Score pro Creator über alle Beiträge messen). Dann bekommt Creator A evtl. 200 Punkte, B 150, C 50 usw., was in Anteile umgerechnet wird. Creator A hätte in dem Beispiel 20% der Gesamtpunkte, also 200 € Auszahlung. Das Monetarisierungsmodul generiert daraus automatisiert eine Transaktion: Über Stripe Connect könnte man jedem Creator Guthaben gutschreiben oder direkt auszahlen (mit vorheriger Verifikation der Bankdaten). Rechnungsbelege werden erstellt.

**Missbrauchsschutz:** Um Missbrauch (z.B. ein Creator meldet sich als Lehrer an und spammt seine eigenen Inhalte, um Score zu erhöhen) zu verhindern, greifen mehrere Mechanismen: - Rollentrennung: Ein **Creator sollte keine zweite eigene Lehrkraft-Identität** haben dürfen, zumindest nicht, um eigene Inhalte zu nutzen. Evtl. kann das System erkennen, wenn der Ersteller eines Inhalts ihn selber aufruft – diese Events werden vom Score ausgeschlossen. - Begrenzung: Extrem häufige Nutzung eines Inhalts durch denselben Lehrer bringt abnehmenden Grenznutzen – Score-Formel kann z.B. pro Lehrer nur einmal pro Monat zählen. - Moderation: Die Community kann verdächtiges Verhalten melden. Zudem kann das Team Auszahlungen zurückhalten, wenn Manipulation vermutet wird. - Transparenz: Durch offengelegte Score-Faktoren kann die Community selbst ein Auge auf die Fairness haben – wie bei Open Source ist offen ein gewisser Schutz (viele Augen, Prinzip).

**Vergütung der Lehrer?** Breakpilot primär vergütet Creator. Lehrkräfte als Endnutzer zahlen in der Regel nicht (bzw. indirekt über Institutionen). Allerdings könnte man Anreize für besonders aktive Lehrer schaffen – etwa Badges oder Ranglisten in der Community (Gamification). Eine direkte Bezahlung von Lehrern für Content-Nutzung ist unüblich und würde falsche Anreize setzen (man will ja, dass sie frei auswählen, nicht fürs Nutzen bezahlt werden). Stattdessen soll das Impact-System ja Content-Qualität belohnen.

**Optionale Einnahmemodelle:** Neben dem Impact-Pool könnten weitere *Monetarisierungsfeatures* bestehen: - **Spenden/Tipps:** Eltern oder andere können freiwillig Creator für bestimmte Inhalte belohnen (ähnlich Patreon). Das könnte über die Community-App laufen (Minds hat z.B. Donations-Funktion[31]). Dies wäre zusätzlich zum

Impact-System und direkt vom Nutzer getrieben. - **Lizenzgebühren für kommerzielle Nutzung:** Wenn Schulen Inhalte außerhalb Breakpilot in Publikationen nutzen wollen, könnte man separate Lizenzierung (Plattformlizenz) anbieten gegen Gebühr, welche teils an Creator fließt. - **Affiliate-Programme:** Z.B. wenn ein Inhalt eine bestimmte externe Ressource empfiehlt (Buch, Sportgerät), könnte man Partnerships nutzen. Allerdings ist das bei öffentlicher Bildung sensibel (Werbefreiheit). - **Premium-Account:** Evtl. ein kostenpflichtiger Pro-Account für Lehrkräfte mit erweiterten Funktionen (z.B. personalisierter Beratung durch Experten, Zugang zu Premium-Inhalten, oder eine Klasse-Set-Funktion mit Schüler-Apps) – die Einnahmen fließen in den Betrieb und indirekt wieder in den Content-Pool.

Für den Kern ist aber das Impact-System zentral, da es die Vision einer **gemeinwohlorientierten Plattform** unterstützt: Inhalte sind offen zugänglich, aber Creator erhalten trotzdem einen monetären Anteil gemessen an ihrem Bildungsbeitrag (Impact). Dieses Modell ist innovativ und muss gut kommuniziert werden.

Technisch ist es in der Architektur verankert durch den dedizierten Analytics/Scoring-Service sowie das Payment-Modul. Solche **Governance-Elemente** wie die Score-Berechnung könnten vielleicht irgendwann der Community mitübergeben werden (z.B. ein Gremium aus Lehrern und Creatoren, das Score-Regeln jährlich prüft) – das gehört zur strategischen Governance, wird aber hier nur angemerkt.

## Lizenzmodelle: Creative Commons vs. Plattformlizenz

Die **Lizenzierung der Inhalte** auf Breakpilot ist sowohl aus rechtlicher als auch aus Community-Sicht ein kritischer Aspekt. Es gilt, Urheberrechte der Content-Ersteller zu respektieren, gleichzeitig aber das Teilen und Remixen der Materialien zu fördern, um pädagogischen Mehrwert zu schaffen. Hier werden die vorgesehenen Lizenzmodelle erläutert und in die Systemarchitektur eingebunden.

**Creative-Commons-Lizenzen (Standardlösung):** Wie bereits strategisch empfohlen, sollen Creator ihre Inhalte in der Regel unter einer Creative-Commons-Lizenz veröffentlichen. CC-Lizenzen sind speziell im Bildungsbereich weit verbreitet und anerkannt<sup>[7]</sup> – viele staatliche Bildungsmedien setzen auf CC BY oder CC BY-SA, um OER zu ermöglichen. Für Breakpilot kämen primär in Frage: - **CC BY 4.0:** erlaubt Nutzung, Veränderung, Weiterverbreitung, auch kommerziell, sofern der Urheber genannt wird. - **CC BY-SA 4.0:** wie CC BY, aber mit „Share Alike“, d.h. abgeleitete Werke müssen wieder unter gleicher Lizenz stehen. - **CC BY-NC 4.0:** erlaubt Nutzung, Veränderung, Verbreitung, aber *nicht kommerziell*. Viele OER nutzen NC, um kommerzielle Nachnutzung (z.B. durch Verlage) zu untersagen. - **CC BY-NC-SA 4.0:** Kombination von NC und SA. - (CC0 oder Public Domain ist unwahrscheinlich, weil Creator meist wenigstens Attribution wollen.)

Standardmäßig könnte Breakpilot z.B. **CC BY-SA** vorschlagen, da dies offen ist und gleichzeitig sicherstellt, dass eventuelle Weitergaben wieder der Community zugutekommen (Copyleft-Effekt). Alternativ, wenn Creator Bedenken haben, wäre CC BY-NC-SA eine Option, um kommerzielle Fremdnutzung zu unterbinden, während die

Plattform selbst (die ja evtl. kommerziell in gewissem Sinne ist) als Ausnahme in den AGB geregelt wird.

**Plattformlizenz (ergänzend):** Trotz CC könnte eine *zusätzliche Vereinbarung* nötig sein, um die Nutzung innerhalb von Breakpilot klar abzudecken. Oft regeln Plattform-AGBs, dass der Creator dem Plattformbetreiber ein Nutzungsrecht einräumt, um den Content anzuzeigen, zu verbreiten etc. Bei CC BY lizenzierten Inhalten ist das zwar automatisch gegeben (weil jeder – damit auch die Plattform – es nutzen darf), aber um rechtlich sauber zu sein, wird Breakpilot in den Nutzungsbedingungen festhalten: *„Der Creator räumt Breakpilot ein einfaches, räumlich unbeschränktes Nutzungsrecht ein, um den Content auf der Plattform zu hosten, anzuzeigen, zu vervielfältigen und zum Download anzubieten.“* So ist selbst bei etwas exotischeren CC-Bedingungen (z.B. NC) der Plattformbetrieb gedeckt, da Breakpilot strenggenommen möglicherweise als kommerzieller Akteur gesehen werden könnte (wenn es Geld einnimmt).

Eine **Plattform-exklusive Lizenz** könnte in Ausnahmefällen angewandt werden: Etwa wenn Breakpilot eigene Inhalte produzieren lässt (z.B. durch beauftragte Pädagogen) und diese zunächst nur auf Breakpilot verfügbar sein sollen. Dann würde man nicht CC nutzen, sondern in der Plattform anzeigen: "© Breakpilot, alle Rechte vorbehalten, nur zur Nutzung auf Breakpilot". Dies sollte aber Ausnahme bleiben, da es dem Open-Education-Grundsatz widerspricht. Dennoch kann es sinnvoll sein, diese Möglichkeit in der Architektur zuzulassen – d.h. das CMS muss Inhalte auch als „Proprietär“ kennzeichnen können und sicherstellen, dass solche Inhalte nicht via offene OER-Kataloge verteilt werden.

**Implementierung im System:** - Im **Upload-Workflow** im Creator-Studio wird der Creator aufgefordert, eine Lizenz auszuwählen (Dropdown mit CC Optionen, plus "Breakpilot-Standardlizenz" falls erwünscht). Hilfetexte erklären die Unterschiede. Creative Commons bietet sogar Tools an, um die Auswahl zu erleichtern[9]. - Die gewählte Lizenz wird in den **Metadaten** des Inhalts gespeichert (z.B. license: CC BY-SA 4.0). Es wäre gut, auch maschinenlesbar (Creative Commons bietet dafür Rights Expression Language) zu speichern, damit Suchmaschinen es indexieren können[32][33]. - Im **Frontend** beim Inhalt wird die Lizenz klar angezeigt, inkl. Symbol und Link zum Lizenztext. So wissen Nutzer, was erlaubt ist. Auch Filter nach Lizenz könnten angeboten werden (z.B. „zeige nur Inhalte, die ich bearbeiten darf“). - **Plattform-Lizenz-Fall:** Falls verwendet, würde man das als eigene Option anzeigen („Nur Nutzung auf Breakpilot gestattet“). Für Nutzer sollte das ein erkennbares Symbol haben, und evtl. ein Warnhinweis, dass Weitergabe nicht erlaubt ist, um Missverständnisse zu vermeiden. - **Lizenzprüfung:** Das System könnte bei Upload z.B. prüfen, ob eingebundene Fremdmaterialien kompatibel sind. Z.B. wenn jemand ein Bild hochlädt, das All Rights Reserved ist, dürften sie es nicht unter CC stellen. Solche Prüfungen sind schwer zu automatisieren, aber Richtlinien werden Moderatoren sensibilisieren, darauf zu achten. Im Zweifelsfall wird zweifelhaft lizenziertes Material entfernt, um Abmahnungen zu vermeiden. - **Änderung der Lizenz:** Falls ein Creator nachträglich die Lizenz ändern will (z.B. von NC auf offener), sollte das möglich sein – aber *nie rückwirkend restriktiver*. CC ist irreversibel in dem Sinne, dass bereits unter CC veröffentlichtes Material von anderen weitergenutzt werden darf unter den alten Bedingungen. Darum kann man höchstens offener werden oder zusätzliche Rechte

einräumen, nicht aber plötzlich entziehen. Das System sollte hierzu eine Warnung ausgeben und es evtl. nur Admins erlauben (um Chaos zu vermeiden).

**Lizenzreporting:** Um dem Bildungsauftrag gerecht zu werden, kann Breakpilot auch externe Schnittstellen nutzen, z.B. Meldung der Inhalte an Kataloge wie **OER Commons** (via API), wenn CC-lizenziert, um die Sichtbarkeit zu erhöhen. Creative Commons veröffentlicht jährlich Berichte über die Anzahl offener Werke[34] – Breakpilot könnte Teil dieser Commons-Bewegung werden. Gleichzeitig behält man durch die Plattform-Community einen *Schutzraum*: Lehrer finden Inhalte leichter als in der weiten OER-Welt, weil Breakpilot kuratiert und fokussiert auf DACH ist.

**Fazit Lizenz:** Das duale Modell (CC + Plattformrecht) ermöglicht maximal offene Nutzung im Bildungssektor und schafft gleichzeitig Klarheit. Die meisten Inhalte werden als **OER frei nutzbar** sein, was dem Bildungsauftrag entspricht, während die Monetarisierung über Impact dafür sorgt, dass Creator dennoch profitieren, ohne klassische Lizenzgebühren zu erheben. Dieses Modell setzt voraus, dass alle Nutzer ein Grundverständnis von offenen Lizenzen haben – hier kann Breakpilot edukativ wirken, indem auf Info-Seiten erklärt wird, was CC bedeutet (ggf. Integration von Creative Commons Erklärmaterial)[35].

## Rollen- und Rechtearchitektur

Die **Rollen- und Rechtearchitektur** bildet das Rückgrat der Sicherheits- und Governance-Matrix von Breakpilot. Sie definiert, welche Aktionen jede Nutzergruppe durchführen darf und stellt sicher, dass Daten nur den vorgesehenen Personenkreisen zugänglich sind. Basierend auf den identifizierten Nutzergruppen ergeben sich folgende Hauptrollen und deren Berechtigungen:

- **Lehrkraft (Teacher):**
- *Zugriff:* Lehrer-Bereich (Arbeitsraum) der Plattform. Kann die Content-Bibliothek durchsuchen und Inhalte ansehen, herunterladen und für den Unterricht verwenden.
- *Interaktion mit Inhalten:* Darf Inhalte **bewerten** (z.B. Sterne vergeben) und **Kommentieren** (eine Rückmeldung an den Creator oder an Kollegen hinterlassen). Kann Inhalte zu Favoriten/Playlists hinzufügen.
- *Erstellen:* Optional kann ein Lehrer auch eigene Inhalte erstellen – hier gibt es eine Design-Entscheidung: Entweder jede Lehrkraft ist automatisch auch Creator (für eigene Materialien), oder es gibt ein separates Creator-Onboarding. Im Sinne der Offenheit könnte man es nicht strikt trennen – viele Lehrer werden Materialien teilen wollen. Also sollten Lehrkräfte zumindest grundlegende Upload-Rechte haben (für einfache Dinge oder schulinterne geteilte Materialien). Aber qualitativ hochwertige Beiträge für die breite Plattform könnten ein "Creator"-Status erfordern (siehe Creator).
- *Community:* Lehrkräfte können – je nach Wunsch – auch in der Community aktiv sein. Standardmäßig sind Lehrkräfte **nicht** als solche in der Eltern-Community sichtbar (denn dort treten sie eher als private Person oder Experte auf, nicht in offizieller Funktion). Sie könnten sich aber z.B. einem Experten-Forum anschließen.

Die Rollentrennung hier ist wichtig: Ein Lehrerprofil im Arbeitsraum enthält evtl. Klarnamen und Schuldaten, während im Elternforum der gleiche Nutzer vielleicht nur einen Vornamen/Alias anzeigt, um Privates zu schützen.

- *Dateneinsicht*: Ein Lehrer sieht keine persönlichen Daten von Eltern oder Creator (außer dem öffentlichen Profil der Creator). Unter Lehrer-Kollegen könnte man Kontaktinformationen teilen, aber das wäre freiwillig.
- *Rechte inhaltlich*: Lehrer dürfen Inhalte im Rahmen der Lizenz nutzen. Können sie Inhalte weiterverbreiten? Bei CC BY schon – aber Breakpilot könnte die einfache Weitergabe technisch einschränken (z.B. keine Massenexport-Funktion), um Traffic auf der Plattform zu halten. Jedoch dürfen Lehrer natürlich OER mit Kollegen teilen – das werden sie wohl eher durch Empfehlung tun ("geh auf Breakpilot und hol es dir").
- **Creator (Inhaltsersteller):**
  - *Zugriff*: Creator-Studio. Kann eigene Inhalte hochladen, bearbeiten, veröffentlichen oder offline nehmen.
  - *Eigene Inhalte verwalten*: Vollzugriff auf die von ihm erstellten Content-Objekte: aktualisieren, ergänzen, Versionen verwalten. Allerdings keine Löschung, falls bereits genutzt? Hier muss eine Policy festgelegt werden: Wenn ein Creator seinen Content löscht, was passiert bei Lehrern, die ihn in Favoriten hatten? Evtl. bleibt er für diese weiterhin verfügbar oder es gibt Ersatz. In den AGB könnte stehen, dass einmal eingestellte Inhalte nicht ohne weiteres gelöscht werden können (um das Commons zu schützen), außer bei berechtigten Gründen. Das ist heikel (Stichwort Urheber-persönlichkeitsrecht vs. Community-Interesse).
  - *Analytics*: Creator sehen **Nutzungsstatistiken** ihrer Inhalte (wie oft genutzt, Durchschnittsbewertung).
  - *Interaktion*: Können Kommentare zu ihren Inhalten beantworten (Dialog mit Lehrern), eventuell auch in Lehrer-Gruppen teilnehmen als Experten.
  - *Community*: Sie können im Eltern-Forum ggf. als "Creator" auftreten – vielleicht gibt es verifizierte Creator-Accounts, die auch Elternfragen beantworten. Das bringt Reputation.
  - *Monetarisierung*: Zugriff auf ihr Monetarisierungs-Dashboard (verdiente Punkte/Geld, Auszahlungsoptionen). Können dort Bankdaten hinterlegen, Auszahlungen nachverfolgen.
- **Eltern (Parent):**
  - *Zugriff*: Eltern-Community (Minds oder Forum). Sie haben **keinen Zugang** zum internen Lehrer-Workspace oder zu unveröffentlichten Inhalten. Eltern sehen Inhalte nur insofern, wie sie von Creator/Lehrern für die Öffentlichkeit freigegeben wurden – vermutlich bleiben detailreiche Materialien nur für registrierte Lehrer. Denkbar ist aber z.B., dass Eltern **Preview-Versionen** sehen oder Inhaltsbeschreibungen, um zu verstehen, was in der Schule genutzt wird. Evtl. bietet Breakpilot eine *Eltern-Info-Seite* je Inhalt: z.B. „Diese Übung wurde in Klasse 4a

gemacht, Ziel: Motorik, hier Download eines Elternbriefs“. Solche Schnittstellen müssten Creator/Lehrer aber bereitstellen.

- *Aktionen*: Eltern können **eigene Beiträge** in der Community verfassen – Fragen stellen, Tipps austauschen, Erfolgsgeschichten posten. Sie können **auf Beiträge antworten, liken, teilen** (je nach Social-Funktionen der Plattform). Sie haben allerdings **keine** Bearbeitungs- oder Uploadrechte im Sinne des Content-CMS – also sie können keine Lehrmaterialien in die Bibliothek laden (höchstens in der Community, aber das wäre dann UGC wie Fotos).
- *Rechte & Sichtbarkeit*: Elternprofile sind separat, enthalten vielleicht nur Pseudonym oder Vornamen, keine Schulbezüge (um Privatsphäre zu schützen). Eltern können Gruppen beitreten, z.B. nach Regionen oder Interessensgebieten.
- *Einschränkungen*: Eltern sollen keinen Zugriff auf Schülerdaten haben (Breakpilot verwaltet im Standard gar keine Schülerdaten, was gut ist). Sie können auch keine Lehrkraft direkt anschreiben (außer in öffentlichen Threads, falls Lehrer dort aktiv sind), um die Lehrer nicht zu überlasten – Moderation stellt hier Puffer.
- *Rollen innerhalb Community*: Es könnte "Eltern-Moderatoren" geben – besonders engagierte Eltern, die Community-Regeln durchsetzen. Diese wären dann eine **Subrolle** (Parent + Moderator).
- **Administrator (Platform Admin)**:
  - Vollzugriff auf alle Systeme oder jeweilige Admin-Module. Kann Benutzer sperren, Rollen zuweisen, Inhalte entfernen (z.B. bei Urheberrechtsverstößen).
  - Konfigurationsrechte: Ändern von Kategorien, globalen Einstellungen.
  - Notfallrechte: z.B. das System in Wartungsmodus setzen.
  - Typischerweise nur wenige (interne) Personen haben diese Rolle, evtl. getrennt nach Bereichen (Community-Admin, Content-Admin).
  - In großen Communities gibt es auch *Staff vs. Admins* Unterschiede; hier aber überschaubar.
- **Moderator / Kurator (Community Moderator bzw. Content Curator)**:
  - *Community Moderator*: Hat spezielle Rechte in der Community-App: kann Posts editieren/löschen, Threads schließen, Verwarnungen oder temporäre Sperren für Nutzer verhängen. Siehe Moderationskonzept unten. Diese Rolle wird ggf. an vertrauenswürdige Lehrer oder Eltern vergeben.
  - *Content Curator/Moderator*: Im Content-CMS könnte es eine Rolle geben, die neue Inhalte **reviewt und freischaltet**. Falls Breakpilot eine Qualitätsprüfung vor Veröffentlichung vorsieht (kann man initial machen, später bei großem Volumen evtl. Community-reguliert), dann haben solche Moderatoren Zugriff auf unveröffentlichte Uploads, können Feedback an den Creator geben („bitte Quellenangabe ergänzen“) oder bei klar ungeeigneten Inhalten die Veröffentlichung ablehnen.

- Solche Rollen könnten Unterarten von „Administrator“ sein mit begrenzterem Zugriff. Z.B. „Content-Moderator“ darf nur im CMS markieren, ob ein Inhalt öffentlich sichtbar wird, hat aber keine Server-Adminrechte.

Technisch wird RBAC wie folgt umgesetzt: - Das Identity-Management (Keycloak oder eigenes System) verwaltet **Benutzergruppen/Rollen**: e.g. teacher, creator, parent, admin, moderator. - Bei Authentifizierung erhält der Client (Frontend) einen Token, in dem die Rolle codiert ist (z.B. JWT mit Role-Claim). - Jeder Microservice prüft bei sensiblen Operationen die Rolle aus dem Token. Z.B. Content-Service erlaubt POST /content nur, wenn Role==creator (oder teacher, falls Upload erlaubt). Ebenso GET /content?fullData (mit evtl. vertraulichen Feldern) nur für berechnigte. - Das Frontend rendert Navigationspunkte abhängig von den Rollen des eingeloggten Users (z.B. nur Creator sehen den "Upload"-Button). - Für Community (Minds): Hier müssen Rollen eventuell synchronisiert werden. Minds kennt eigene Roles (Admin, Moderator, Member). Über SSO könnte man z.B. allen Breakpilot-Admins die Minds-Admin Rechte geben, und bestimmten Breakpilot-Usern die Moderator-Gruppe in Minds zuweisen. Evtl. kann man dies manuell pflegen oder via Minds API. - Role-Based Access ist ein bekanntes Konzept und war auch beim OpenSIS SIS ein Kernfeature („permissions tailored to each role — from admins to teachers to parents“[4]). Breakpilot folgt dem genauso strikt, um **Datensicherheit** zu gewährleisten: Eltern können z.B. *niemals* auf interne Lehrer-Daten zugreifen, Lehrer sehen nur die Inhalte, nicht etwa private Community-Daten, etc.

**Feingranulare Berechtigungen:** Innerhalb der Module kann es weitere Rechteebenen geben. Z.B. könnte man unter Lehrern "Gruppenleiter" oder sowas definieren, aber vermutlich overkill. Wichtiger ist z.B.: Ein Creator kann *nur seine eigenen* Inhalte editieren, nicht die von anderen. Das erreicht man durch **Owner-Prüfungen** bei entsprechenden API-Calls (der Content-Service kennt die Creator-ID eines Inhalts und vergleicht mit der Aufrufer-ID). Bei Community: Moderatoren können Sachen aller Leute moderieren, normale User nur ihre eigenen Beiträge editieren.

**Datenschutz & Rollen:** Rollen definieren auch, welche Datenfelder ein Nutzer sehen darf. Beispiel: Das Profil eines Creators könnte Kontaktinfo enthalten (wenn Creator das will). Ein Lehrer sieht vielleicht davon nur den Namen und Bio. Ein Admin sieht E-Mail etc. Solche Unterschiede muss das API bzw. Backend berücksichtigen und Daten ausblenden, sofern unbefugt. (Beispiel in Hygraph e-learning: unterschiedliche Sichten je nach Rolle auf ein Profil[36] – analog könnte Breakpilot bestimmte Infos je nach Betrachterrolle ausblenden).

Insgesamt stellt die sauber implementierte Rollen- und Rechtearchitektur sicher, dass jede Nutzergruppe **zielgerichtet arbeiten** kann, ohne überfordert zu sein oder Sicherheitslücken zu erzeugen. Es verhindert auch **Datenlecks** – z.B. dass Eltern aus Versehen Zugriff auf urheberrechtlich geschützte Lehrer-Materialien bekommen oder Ähnliches. Damit dies robust bleibt, sollte es gründlich getestet und dokumentiert werden.

## Datenschutz und DSGVO-Konformität

Die Einhaltung des **Datenschutzes** und der **DSGVO** (Datenschutz-Grundverordnung) ist für Breakpilot unabdingbar, da personenbezogene Daten von Lehrkräften und Eltern

verarbeitet werden und die Plattform im Bildungsumfeld operiert. Schon bei der Architekturplanung werden daher *Privacy-by-Design*-Prinzipien umgesetzt[10]. Die wichtigsten Maßnahmen und Überlegungen sind:

**Datenminimierung:** Breakpilot erhebt nur die Daten, die für den Zweck unbedingt nötig sind[11]. Bei Registrierung zum Beispiel: - Für Lehrkräfte reichen Name, E-Mail, ggf. Schule/Institution (letzteres optional, aber hilfreich zur Verifizierung). Man könnte sogar Vorname+Alias statt Klarnamen erlauben, sofern die Identität nicht für die Plattformfunktion kritisch ist. - Für Eltern: noch weniger – z.B. Nickname, E-Mail. Keine Adressen, keine Kinder-Namen (das könnte irgendwann relevant sein, wenn Eltern sagen wollen "Mein Kind Klasse 4 nutzt das", aber besser nicht einbauen – lieber allgemein halten). - Keine sensiblen Daten wie Geburtsdatum (außer ggf. zur Verifikation Volljährigkeit falls nötig). - Telemetriedaten werden soweit möglich anonymisiert. Z.B. Analytics zählt Events, aber protokolliert nicht "User X hat um 14:00 das gemacht" in personenbeziehbarer Form, sondern pseudonym oder aggregiert.

**Einwilligungen und Transparenz:** - Nutzungsbedingungen und Datenschutzerklärung werden von Anfang an erstellt und im Onboarding müssen Benutzer diesen ausdrücklich zustimmen (kein vorangekreuztes Kästchen – aktive Zustimmung gemäß DSGVO). - Bei Funktionen, die eventuell zusätzliche Daten nutzen, wird explizit gefragt. Beispielsweise, wenn ein Lehrer optional ein Profilfoto hochladen kann, wird das freiwillig sein. - Cookies: Einsatz eines Consent-Banners, wo nur technisch notwendige Cookies standardmäßig aktiv sind. Tracking-Cookies Dritter werden vermieden; Web Analytics (für die Plattformentwicklung) könnte man selbst gehostet (z.B. Matomo) betreiben und IP-Anonymisierung aktivieren, oder ganz darauf verzichten, stattdessen die internen Metrics nutzen.

**Datenaufbewahrung und -löschung:** - Jeder Nutzer hat das Recht, seine Daten einzusehen, zu exportieren und löschen zu lassen (Art. 15-20 DSGVO). Daher braucht Breakpilot eine **Datenexport-Funktion** (z.B. JSON/XML Download aller persönlichen Daten) und eine **Konto-Löschen-Funktion**. Letztere muss sorgfältig umgesetzt sein: Wenn ein Creator sein Konto löscht, was passiert mit seinen Inhalten? Evtl. bleiben sie als "anonym" erhalten (da sie ja offen lizenziert sind), oder man muss sie auf Wunsch auch entfernen (bei CC würde das zwar bereits erteilte Rechte nicht zurückholen, aber aus Plattform kann man es nehmen). Die Policy dazu sollte in den AGB klar sein. Generell muss eine **Löschroutine** vorhanden sein, die alle personenbezogenen Daten eines Users auf Anforderung entfernt, außer solchen, die aus Rechtsgründen archiviert werden müssen (z.B. Zahlungsbelege). - *Speicherfristen:* z.B. Logfiles mit IP-Adressen (falls vorhanden) werden nach X Tagen gelöscht oder anonymisiert. Unnötige Protokolldaten gar nicht erst speichern.

**Server-Standort und Auftragsverarbeitung:** - Alle Server und Datenbanken werden in der EU betrieben[37][12] – idealerweise sogar bei einem EU-Company Host, um Probleme wie den CLOUD Act zu minimieren[26]. Falls Cloud-Dienste der großen Anbieter genutzt werden, müssen Standardvertragsklauseln und Verschlüsselung streng eingehalten werden, aber aus PR-Sicht wäre es besser, z.B. auf einen deutschen Cloud-Anbieter (Ionos,

Hetzner, etc.) zu setzen, was DSGVO einfacher macht. - Mit allen Drittanbietern (z.B. E-Mail-Service, SMS, Stripe) müssen **AV-Verträge** (Auftragsverarbeitung) abgeschlossen werden, um DSGVO-Konformität zu sichern. - **Schrems II** wird beachtet: Keine Datenübermittlung in die USA ohne adäquates Schutzniveau. E-Mails könnte man z.B. über EU-basierte SMTP Relay senden, oder falls man US-Dienst nutzt, die Einwilligung der User einholen.

**Datensicherheit:** - Alle Verbindungen TLS 1.3 verschlüsselt. HSTS Header etc. - Passwörter nicht im Klartext, nur als salted Hash gespeichert. Besser noch: Delegation an Keycloak, das state-of-art Hashing und ggf. 2FA anbietet. - Regelmäßige Sicherheitsupdates der OSS-Komponenten. Container-Images scannen (z.B. Trivy). - Internes Prinzip der **Minimalprivilegien**: Jeder Service bekommt nur die DB-Zugänge/Rechte, die er braucht. In Kubernetes z.B. Network Policies, dass z.B. der Community-Pod keinen Zugriff aufs interne Content-DB-Pod hat etc. - Backups: Datenbanken werden verschlüsselt gesichert, Zugriff nur Befugte. Offsite-Backups auch verschlüsselt. Testweise Wiederherstellungen, um sicher zu sein.

**Privacy-by-Design Features:** - Voreinstellungen datenschutzfreundlich: z.B. Profile standardmäßig privat, Tracking default aus. - Wo möglich Pseudonymisierung: In Analytics nur User-IDs, keine Klarnamen. Wenn Statistiken öffentlich geteilt werden (z.B. „Lehrer X hat 5 Inhalte genutzt“), lieber aggregiert („5 Inhalte wurden genutzt in Schule Y“ ohne Namensnennung). - Falls tatsächlich mal Schülerdaten ins Spiel kämen (z.B. ein Lehrer will Klassen anlegen, um Fortschritt zu tracken – im Konzept bislang nicht vorgesehen), müsste das streng isoliert und nur lokal bleiben. Besser, Breakpilot verzichtet komplett auf Schüler-Personendaten, das vereinfacht DSGVO enorm (keine Kinderdaten direkt in der Plattform).

**Moderation & Datenschutz:** - Moderatoren dürfen nur das sehen, was für ihre Aufgabe nötig ist. Ein Community-Moderator sieht z.B. die Inhalte der Posts und das öffentliche Profil, aber nicht die E-Mail des Users (es sei denn, er ist auch Admin). - Bei Meldungen (Reports) von Verstößen sollte das System für den Moderator alle relevanten Infos anzeigen, aber keine unnötigen. Beispiel: Ein Eltern-Post wird gemeldet, Moderator sieht den Post-Inhalt und den Username, kann bei Bedarf den Account kontaktieren über ein System (ohne die E-Mail direkt zu kennen, z.B. via interner Message). - Logs: Zugriffe von Admins/Mods auf persönliche Daten sollten protokolliert werden (Audit Log), um Missbrauch zu verhindern.

**Schulung und Awareness:** - Auch Teil der Architektur im weiteren Sinne: Das Team und ggf. Moderatoren werden in DSGVO geschult, damit sie korrekt handeln. Z.B. kein Austausch personenbezogener Infos via unsicherer Kanäle, klare Verfahren bei Datenanfragen (an wen wenden bei Auskunft etc.).

Durch diese Maßnahmen wird Breakpilot **DSGVO-konform** betrieben. Die Planungsdokumente sollten vom Datenschutzbeauftragten geprüft werden. Zudem kann man erwägen, eine **Datenschutz-Folgenabschätzung (DPIA)** durchzuführen, weil Social Community + Bildungsdaten durchaus Risiko haben. In Summe soll Breakpilot ein Vorzeigeprojekt sein, das zeigt, dass digitale Bildungsplattformen mit hoher Funktionalität

und strengem Datenschutz realisierbar sind – ein Aspekt, der z.B. bei öffentlichen Schulen sehr wichtig ist, da hier skeptische Eltern und Vorgaben der Ministerien beachtet werden müssen.

## Community-Moderation und Qualitätskontrolle

Eine lebendige Plattform benötigt klare Mechanismen, um **Inhalte und Interaktionen zu moderieren** und **Qualität sicherzustellen**. Breakpilot setzt hier auf einen kombinierten Ansatz aus *Community-getriebener Moderation*, *automatisierten Prüfungen* und *professioneller Kontrolle durch das Plattform-Team*.

**Community-Moderation (soziale Kontrolle): - Melden-Funktion:** Jedes Community-Mitglied (sei es Lehrer oder Eltern) kann unangemessene Inhalte melden. In der Eltern-Community ist unter jedem Post eine „Melden“-Option verankert, ebenso bei Kommentaren zu Inhalten im Lehrerbereich. Diese Meldungen werden in ein zentrales Moderations-Queue gespeist, welches Moderatoren einsehen[14]. - **Moderatoren-Rollen:** Wie oben erwähnt, werden aus der Nutzerschaft Moderatoren ernannt. Anfangs übernimmt das Kernteam diese Rolle, später können vertrauenswürdige Nutzer hinzukommen. Moderatoren haben das Recht, Beiträge zu **editieren, auszublenden oder zu löschen** und User zu verwarnen. Ein Eskalationsstufen-System kann implementiert werden: z.B. 1. Verstoß -> private Verwarnung, 2. Verstoß -> temporäre Schreibsperre, 3. Verstoß -> Account-Sperre (letzteres nur durch Admin). - **Richtlinien & Onboarding:** Es gibt ausführliche **Community-Richtlinien**, denen Nutzer bei Registrierung zustimmen. Darin wird z.B. untersagt: Beleidigungen, diskriminierende Inhalte, politische/religiöse Propaganda, Werbung, Verletzung von Urheberrechten, Nennen von Schülern/Klarnamen etc. Moderatoren achten auf Einhaltung. Sie haben einen **Leitfaden**, was zu tun ist bei bestimmten Verstößen, um konsistente Entscheidungen zu gewährleisten. - **Reputation & Anreize:** Gutes Verhalten kann belohnt werden – z.B. „Top Community Mitglied“ Abzeichen für hilfsbereite Eltern. Dies fördert eine positive Kultur und entlastet Moderatoren, weil Mitglieder sich bemühen, die Guidelines einzuhalten, um ihre Reputation nicht zu verlieren. - **Partizipative Elemente:** Man könnte langfristig erwägen, Modelle wie *“weighted voting”* einzuführen, wo vertrauenswürdige Nutzer mehr Gewicht haben bei Inhaltsbewertungen, um Qualität zu bestimmen. In dezentralen Netzwerken gibt es Ideen wie Reputation Scores oder Stakeholder-Voting[38]. Für Breakpilot könnte etwa gelten: Lehrkräfte mit hoher Erfahrung (viele Beiträge, viel genutzte Inhalte) haben mehr Einfluss bei der Bewertung von neuen Inhalten. Das würde qualitativ gute Materialien schneller nach oben spülen. Allerdings sind solche Systeme komplex und müssen transparent sein, sonst wirken sie willkürlich.

**Automatisierte Moderationstools: - Wortfilter:** Eingebaute Filter erkennen bestimmte obszöne Wörter, Hassrede, persönliche Daten (z.B. wenn jemand Telefonnummer postet) etc. und *blockieren* diese entweder direkt oder markieren sie für Moderator-Sichtung. Tools dafür gibt es (z.B. Open Source Libraries für Profanity-Filter in verschiedenen Sprachen). - **Bild- und Video-Moderation:** Sollte in der Community multimedialer Content geteilt werden, könnten KI-Services zur Erkennung von ungeeigneten Bildern (Gewalt, Pornografie) eingesetzt werden. Open-Source-Modelle oder moderate SaaS (sofern DSGVO-konform)

können hier hilfreich sein. Mindestens aber manuelle Sichtung. - **Spam-Erkennung:** Captcha beim Registrieren und Posten, Ratenbegrenzungen (nicht 100 Posts pro Minute etc.) schützen vor Spam-Bots. Auch E-Mail-Verifikation erforderlich, um Troll-Accounts zu erschweren.

- **Plagiatsprüfung (für Inhalte):** Um Qualität und Urheberrechte zu sichern, könnte bei neu hochgeladenen Materialien eine Plagiatsprüfung laufen: Vergleich mit bekannten Quellen (z.B. Websearch oder einer DB mit Schulbüchern). Open-Source-Tools dafür sind rar, aber man könnte z.B. Integrationen zu PlagScan (falls öffentl. API) oder eine simple Websuche nach markanten Sätzen anstoßen. Das Moderationsteam würde auffällige Ergebnisse einsehen und entscheiden, ob der Inhalt zugelassen wird. - **Qualitäts-Checklisten:** Angelehnt an OER Commons kann Breakpilot beim Erstellen von Inhalten bereits Hinweise geben: z.B. „Haben Sie Quellen angegeben? Ist das Material aktuell? Ist es inklusiv formuliert?“<sup>[15]</sup>. Eine eingebaute **Checkliste** beim Upload führt den Creator durch Qualitätskriterien, um selbst schon Fehler zu vermeiden. - **Peer-Review-Funktion:** Man könnte (evtl. in Phase 3+) einen Workflow einführen, bei dem neue Inhalte erst nach dem Prinzip „4 Augen“ freigeschaltet werden. D.h. wenn ein Creator etwas hochlädt, wird es zunächst nur für Moderatoren oder eine kleine Gruppe von *Peer-Reviewern* sichtbar. Diese können Feedback geben („methodisch gut, aber Rechtschreibfehler auf Folie 3“) und der Creator verbessert es, bevor es offiziell erscheint. Das ist aufwendig, aber für Qualität enorm förderlich. Vielleicht beschränkt man es auf „neue Creator“ – erfahrene Creator können direkt publizieren, weil sie sich bewährt haben.

**Qualitätskontrolle der Inhalte: - Bewertungssystem:** Kern der Qualitätssicherung ist das Bewertungssystem durch die Nutzer (Lehrkräfte). Wenn viele Lehrer einen Inhalt gut bewerten, ist das ein starkes Signal für Qualität. Diese Bewertungen fließen ins Impact-Scoring ein, was Creator motiviert, qualitativ zu arbeiten. - **Kommentare und Reviews:** Ausführlichere Rückmeldungen ermöglichen es, Stärken/Schwächen von Inhalten zu verstehen. Moderatoren könnten hier steuernd eingreifen, falls mal unsachliche Kritik auftaucht, aber überwiegend sollen fachliche Diskussionen gefördert werden („Ich habe dieses Spiel ausprobiert, meine Klasse war begeistert. Tipp: eventuell Zeitlimit verlängern.“). So entsteht eine *gemeinschaftliche Verbesserung* der Materialien. - **Kuratierte Sammlungen:** Das Breakpilot-Team oder Partner (z.B. Bildungsinitiativen) können Inhalte sichten und **empfohlene Sammlungen** erstellen – etwa „Top 10 Bewegungsübungen für die Grundschule“. Solche Sammlungen werden hervorgehoben, was guten Inhalten Reichweite gibt. Gleichzeitig fungiert es als Qualitätssiegel, das neuere oder weniger bewertete Inhalte noch nicht haben. - **Kontinuierliche Aktualität:** Inhalte haben ein „Zuletzt geprüft am...“ Feld. Moderatoren oder Creator sollten längere Zeit unveränderte Inhalte gelegentlich checken (sind Links noch aktiv? Stimmt das didaktisch noch?). Die Plattform kann z.B. nach 2 Jahren eine automatische Erinnerung schicken, den Inhalt zu aktualisieren. - **Offensive/Inappropriate Content:** Sollte trotz aller Vorauswahl mal ein Inhalt hochgeladen werden, der unpassend ist (politisch extrem, werblich, falsch), greifen die **Content-Moderatoren** ein: Der Inhalt wird aus der Bibliothek entfernt oder gar nicht erst freigegeben. Materialien, die bspw. urheberrechtlich geschützt sind und fälschlich hochgeladen wurden, werden sofort gelöscht (Notice-and-Takedown). Das

Moderationsteam muss solche Fälle dokumentieren, um im Wiederholfall Nutzer zu sperren.

**Governance-Elemente:** - Breakpilot könnte einen **Beirat** aus Lehrpersonen, Datenschützern, vllt. Schülervertretern einrichten, der die inhaltliche Ausrichtung und Community-Regeln begleitet. Dieser Beirat könnte bei schwierigen Moderationsfragen konsultiert werden (z.B. politisch heikle Inhalte). - **Nutzereinbindung:** Tools wie Umfragen oder ein öffentliches Moderationslog (angepasst) steigern die Transparenz: Community sieht „User X wurde 1 Tag gesperrt wegen Y“ – das mag in Elternkontext zu hart sein (vielleicht lieber anonymisiert), aber intern wenigstens sollte dokumentiert sein, wie Regeln durchgesetzt werden. So wird Willkür vermieden. - Moderatoren selbst sollten evaluiert werden – gab es Beschwerden über Moderation? Dann greift der Admin ein und schult nach oder tauscht aus.

**Quality Metrics:** - Plattformseitig kann man Kennzahlen tracken wie „Durchschnittliche Inhaltsbewertung“ oder „Anteil der gemeldeten Posts pro Tag“. Diese geben Hinweis auf die Community-Gesundheit. Sinkt die Qualität (z.B. viele Streitgespräche im Forum), kann man gegensteuern (Themen moderieren, neue Regeln). - Eine erfolgreiche Qualitäts-Community sieht so aus, dass **die Nutzer selbst stolz sind** auf das inhaltliche Niveau. Dies wird erreicht, indem man gute Beiträge belohnt (via positive Rückmeldung, Abzeichen) und schlechte konsequent entfernt.

Bereits etablierte Plattformen zeigen, wie wichtig diese Schritte sind. In Minds z.B. gab es Kritik wegen zu laxer Moderation[24] – Breakpilot positioniert sich bewusst anders: *Freie Meinungsäußerung ja, aber in einem respektvollen, bildungsorientierten Rahmen.* Entsprechend stärker wird moderiert, jedoch immer nachvollziehbar.

Als Orientierung könnten die **OER Commons Qualitätsrichtlinien** dienen, die definieren, wann ein Material als „bildungsreif“ gilt[15], oder die **Quality Matters** Initiativen für Online-Lernen. Diese Standards können ins Prüfschema einfließen. Breakpilot strebt an, dass Materialien zumindest die grundlegenden Qualitätskriterien erfüllen: inhaltliche Richtigkeit, Altersangemessenheit, keine Bias/Diskriminierung, pädagogische Nützlichkeit, technische Zugänglichkeit (z.B. barrierearm, offene Formate).

**Zusammenfassung Moderation/QS:** Die Systemarchitektur unterstützt dies mit entsprechenden Modulen (Moderations-Dashboard, Rollen, Bewertungsfunktionen, Filtern). Darüber hinaus sind es aber die *Prozesse* und *Menschen*, die Qualität letztlich sichern. Deshalb werden klar geregelte Workflows und Verantwortlichkeiten definiert: - Tägliche Durchsicht der Meldungen durch Moderatoren. - Wöchentliche Team-Meetings zu auffälligen Fällen. - Monatliche Reports an das Management/Beirat über Communitystatus. - Ständige Verbesserung der Moderationstools (wenn z.B. toxische Phrasen auftauchen, Filter erweitern).

All das stellt sicher, dass Breakpilot ein **vertrauenswürdiges Ökosystem** bleibt, in dem sich die Nutzer wohlfühlen und die Inhalte hohen Ansprüchen genügen.

## Strategische und technische Zusammenfassung

**Breakpilot** soll als modulare, skalierbare Plattform eine neuartige Infrastruktur für Bildungspartner im DACH-Raum bieten. Strategisch wurden ambitionierte Ziele gesetzt – von der breiten Open-Source-Nutzung bis zum Impact-basierten Belohnungssystem – und technisch untermauert. Durch die **Trennung der Nutzerbereiche** (Lehrer-Arbeitsraum, Creator-Studio, Eltern-Community) mit jeweils optimierten Open-Source-Komponenten entsteht ein System, das auf die unterschiedlichen Bedürfnisse eingeht und dennoch integriert zusammenwirkt[3]. Das **Open-Source-Ökosystem** liefert dabei viele Bausteine, die bewährt und flexibel sind, wodurch Breakpilot Kosten spart und auf Community-Wissen aufbaut (etwa bei LMS-Funktionalitäten und Kommunikationstools[1][16]).

Besonders hervorzuheben ist das **Monetarisierungsmodul** mit Impact-Scoring: Es bietet eine innovative Lösung für eines der größten Probleme von offenen Bildungsressourcen – nämlich der fairen Entlohnung der Ersteller. Indem es Engagement und Nutzungsdaten in **transparente Punktwertungen** umsetzt und daran echte Auszahlungen koppelt, motiviert es Creator, exzellente Inhalte bereitzustellen, ohne auf restriktive Paywalls oder volatile Kryptotoken zurückzugreifen[5]. Dieses Modell ist im Kontext von EdTech relativ neu, aber vielversprechend, um nachhaltige Content-Erstellung zu gewährleisten.

Die vorgeschlagene Architektur legt starken Wert auf **Datenschutz und Sicherheit** – ein Muss im europäischen Bildungsbereich. Durch EU-zentrisches Hosting und Privacy-by-Design (Minimaldaten, SSO-Kontrolle, Rechteverwaltung) kann Breakpilot Schulen und Nutzern glaubhaft versichern, dass ihre Daten sicher behandelt werden[12][11].

**Rollen & Rechte** wurden detailliert durchdacht, sodass jede Nutzergruppe genau die Freiheiten und Begrenzungen hat, die ihrem Zweck entsprechen[4]. Dadurch wird nicht nur Sicherheit erhöht, sondern auch die **User Experience verbessert**: Nutzer sehen nur, was für sie relevant ist, was die Anwendung übersichtlich und effizient macht.

**Community-Moderation und Qualitätskontrolle** runden das Konzept ab, indem sie einen Rahmen schaffen, in dem wertschätzender Austausch und hochwertige Inhalte gedeihen können. Mit einer Kombination aus Community-Regulierung (Melden, Moderatoren) und professionellen Richtlinien (Review-Prozess, Qualitätskriterien[15]) wird Breakpilot Inhalte bieten, denen Lehrer vertrauen und die Eltern überzeugen.

Die Systemarchitektur ist bewusst **modular und skalierbar** gehalten: neue Module können ergänzt werden, Lastspitzen durch horizontale Skalierung aufgefangen, regionale Erweiterungen vorgenommen, ohne das Gesamtsystem zu gefährden. Es wurde auch Raum für zukünftige Features gelassen – seien es KI-Assistenzfunktionen für Lehrkräfte, erweiterte Schüler-Interaktionen oder internationale Content-Partnerschaften. Dank Microservices-Ansatz können solche Erweiterungen parallel entwickelt und integriert werden, sofern sie die definierten APIs nutzen.

**Fazit:** Breakpilot verbindet auf einzigartige Weise eine **strategische Vision für offene Bildung** mit einer robusten, modernen **Software-Architektur**. Mit diesem Fundament kann die Plattform sich als zentrales digitales Ökosystem im DACH-Bildungsraum etablieren –

**innovativ durch Open Source, sozial nachhaltig durch Impact-Monetarisierung, vertrauenswürdig durch Datenschutz und Qualitätsmanagement.** Die vorgeschlagene Architekturumsetzung schafft die Voraussetzung, diese Vision Schritt für Schritt Realität werden zu lassen, und kann mit den richtigen Ressourcen in die Tat umgesetzt werden. Alle relevanten Module, Schnittstellen, Datenflüsse und Governance-Elemente wurden berücksichtigt, sodass Breakpilot auf einem **soliden, skalierbaren und community-orientierten** Fundament aufbauen kann.

---

[1] [2] [16] [19] [28] [29] 25 most popular open source software in education | Rocket.Chat

<https://www.rocket.chat/blog/open-source-software-in-education>

[3] [36] Site architecture of an e-learning platform | Hygraph

<https://hygraph.com/blog/elearning-platform-architecture>

[4] openSIS | Student Information & School Management Software

<https://www.opensis.com/>

[5] How to Build an NFT Launchpad in 2026? A Step-by-Step Guide for Startups | by Blockchain App Factory | Predict | Dec, 2025 | Medium

<https://medium.com/predict/how-to-build-an-nft-launchpad-in-2026-a-step-by-step-guide-for-startups-4bfe34401cbd>

[6] [24] [27] [30] Minds (social network) - Wikipedia

[https://en.wikipedia.org/wiki/Minds\\_\(social\\_network\)](https://en.wikipedia.org/wiki/Minds_(social_network))

[7] [32] [33] [35] What is Creative Commons? | Program for Open Scholarship and Education

<https://pose.open.ubc.ca/open-education/creative-commons/what-is-creative-commons-and-what-are-creative-commons-licenses/>

[8] [9] [34] Technology Platforms - Creative Commons

<https://creativecommons.org/share-your-work/platform/>

[10] [11] [12] [13] [26] [37] GDPR-Compliant Hosting: Best Practices for Developers in 2025 - DEV Community

[https://dev.to/dev\\_tips/gdpr-compliant-hosting-best-practices-for-developers-in-2025-jl5](https://dev.to/dev_tips/gdpr-compliant-hosting-best-practices-for-developers-in-2025-jl5)

[14] 7 Essential Online Community Moderation Steps | Higher Logic

<https://www.higherlogic.com/blog/community-moderation-best-practices/>

[15] Resource Quality Guidelines : OER Commons Help Center

<https://help.oercommons.org/support/solutions/articles/42000020983-resource-quality-guidelines>

[17] [18] [20] [21] [22] [23] [25] [31] Minds Networks | Monetize and retain your community

<https://networks.minds.com/>

[38] Decentralized Social Media | Business & Information Systems Engineering

<https://link.springer.com/article/10.1007/s12599-025-00952-4>